EuDML
The EUROPEAN DIGITAL MATHEMATICS LIBRARY

ICT PSP
ICT POLICY SUPPORT PROGRAMME
part of the Competitiveness and Innovation Framework Programme CIP

DELIVERABLE

Project Acronym:	EuDML
Grant Agreement number:	250503
Project Title:	The European Digital Mathematics Library

D11.2: EuDML assessment and evaluation plan

Revision: 1.0 as of 25th May 2011

Authors:

Michael Jost	FIZ Karlsruhe, FIZ
Radoslav Pavlov	Institute of Mathematics and Informatics, BAS, IMI-BAS

Contributors:

Jiří Rákosník	Institute of Mathematics AS CR, IMAS
Lígia Rodrigues	Instituto Superior Técnico, IST

Project co-funded by the European Comission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	✓
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Revision	Date	Author	Organisation	Description
0.1	01 April 2011	Michael Jost	FIZ	First version with structure of the deliverable.
0.2	14 April 2011	Michael Jost	FIZ	Content added
0.3	28 April 2011	Michael Jost	FIZ	Introduction and Evaluation Framework consolidated
0.4	2 May 2011	Michael Jost	FIZ	Digital Library Conformance Check List added with generic digital library evaluation criteria
0.5	5 May 2011	Michael Jost	FIZ	Evaluation Roadmap consolidated
0.6	6 May 2011	Michael Jost	FIZ	Summary added
0.7	10 May 2011	Michael Jost	FIZ	Version for internal review
1.0	25 May 2011	Michael Jost	FIZ	Review comments integrated

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Contents

Executive summary

This report develops the global plan for self-evaluation of EuDML. The EuDML project aims to design and build a collaborative digital library service that will collate the mathematical content brought by 11 of its partners and make it accessible from a single platform, tightly integrated with relevant infrastructures.

An evaluation framework is defined according to findings and recommendations from the DELOS Digital Library Classification and Evaluation Scheme. Assessment and evaluation criteria are structured following the concepts of digital library components (content, system, users) and their relationships (usability, usefulness, and performance), as well as policies governing the system. Further general evaluation criteria are derived from the Digital Library Reference Model Conformance Checklist from project DL.org.

Special evaluation topics and evaluation criteria suitable and applicable to EuDML are derived from the project description, goals, and assertions of the EuDML "Description of Work", which will be checked for conformance, and from relevant findings of the "Report on evaluation of existing services and content providers" (Deliverable 11.1 of the EuDML project), which provides benchmarks from previous evaluations of related services. Also, recommendations from the "Usability study" (Deliverable 6.1 of the EuDML project) are considered.

An evaluation roadmap is given which describes how the evaluation framework and criteria will be implemented during two rounds of evaluation — the first with the aim of providing guidance and recommendations for further development during the project, and the second with the aim of guiding further actions beyond the lifetime of the project (issues of sustainability).

“Everything that can be counted does not necessarily count;
everything that counts cannot necessarily be counted.”
Albert Einstein

1 Introduction

Assessment is the process of measuring, quantifying, and/or describing the attributes of a system covered by the evaluation.

Evaluation is the systematic process of determining the merit, value, and worth of something. Evaluation is broader than assessment and involves making a judgement as to the effectiveness of an assessment. Evaluation has a certain goal, methodology and devices or techniques. The goal suggests some evaluation criteria, which sometimes break down to evaluation parameters. At times these parameters are qualitative, at times quantitative. If quantitative, a metric can be defined. In the case of metrics the values are comparable over different evaluations. A successful evaluation should be transferable (apply to other contexts) and confirmable (verifiable).

An *evaluation framework* defines a structure of methods and supporting devices designed to support the evaluation needs of a community.

The *evaluation goal* is the desired knowledge about a system, its performance or usability[6].

1.1 EuDML context

This report develops the global plan for self-evaluation of EuDML. The EuDML project aims to design and build a collaborative digital library service that will collate the mathematical content brought by 11 of its partners and make it accessible from a single platform, tightly integrated with relevant infrastructures such as the Zentralblatt MATH. As such, it is the first attempt toward a large-scale implementation of a Digital Mathematics Library (DML), and is expected to pave the way towards a truly inclusive and global DML.

Findings and recommendations from the EuDML assessment and evaluation activity should support decisions about EuDML future development and sustainability. For this purpose, the following facets are to be considered:

- Organization: The organizational context and sustainability (risks, threats, weaknesses, potentials, strengths and opportunities for survival and growth)
- Services: The technical and functional characteristics of the services (interfaces, reliability, robustness, etc., for both humans and other machines usage)
- Contents: The quantity and quality of the contents

Annex I to the EuDML Grant Agreement (The “Description of Work”, DoW) defines general and specific project goals and describes intended EuDML service components. Actual developments during the project are to be assessed and compared to these stated goals.

Three Key Success Indicators (KSIs) are explicitely mentioned in the DoW. The first two mainly deal with timeline and budget compliance, conformance to the Work

Plan, and other project management issues. The third KSI focuses on project outcomes and perspectives, and is of special relevance to this evaluation exercise, since it contains explicit indicators that will serve as evaluation criteria:

- "Long-term Goals: The proposed project has the aim that the common EuDML services will be proved to satisfy needs of the scientists. Furthermore, the project is intended to promote a successful further process to extend and propagate the whole EuDML network initiative. First steps in this direction can be measured by the number of additional partners and projects joining the network. Progress in EuDML adoption by content providers and users will be monitored by indicators outlined in Table 1."

Table 1: Performance monitoring

No.	Objective/expected result	Indicator name	Year 1	Year 2	Year 3
1	Diversity of partnership (European dimension)	Number of participating content repositories (partners or associated partners)	5	8	12
2	Diversity of content (Discipline coverage)	Number of integrated collections (e.g. journal runs, book series, conference proceedings, Ph.D. theses)	50	100	200
3	Critical mass	Number of integrated resources (digital items reused from the network)	30,000	80,000	160,000
4	Enhancement progress	Number of supported resources (items with enhanced MathML full text and metadata)	10,000	30,000	90,000
5	Internal networking	Links between database objects	100,000	250,000	500,000
6	External networking	External links (generated for third parties using dedicated component, or detected through referrer website)	50,000	150,000	500.000
7	Registered users contributing annotations	User's involvement	0	500	1,000
8	User's contribution	User generated content (annotations and external links)	0	500	10,000

These figures will be used as evaluation criteria, embedded in the evaluation framework that is developed below.

1.2 Setting up an evaluation framework

It is clear that setting up a framework for the evaluation of digital library systems is not only a difficult task but also quite controversial with many models and formalisms already available and many contributions from related research areas.

A digital library is a particular kind of information system and consists of a set of components, typically a collection (or collections), a computer system offering diverse services on the collection (a technical infrastructure), people, and the environment (or usage), for which the system is built. When designing a digital library the starting points are its intended usage and the corresponding user needs. The model is based upon the assumption that the user and the user needs specify the main requirements with respect to the range and content of the collections. The nature of the collections will thus predetermine the range of technologies that are needed. The attractiveness of the collections to the users and the ease of use of the technologies by the user group will determine the extent of the usage of the digital library.

The design and development of a digital library is expensive. Evaluation results from previous systems can give guidelines as well as assist in determining methods for the construction of cost-effective and sustainable new systems. "Doing it right" in the first phases of system development is critical for the final result as the quality of an end-product depends to a great extent on the quality of early requirements and conceptual models.

This evaluation will be carried out according to findings and recommendations from the DELOS Classification and Evaluation Scheme [6].

DELOS was a Network of Excellence on Digital Libraries partially funded by the European Commission in the frame of the Information Society Technologies Programme (IST) [3]. DELOS conducted a joint program of activities aimed at integrating and coordinating the ongoing research efforts of the major European teams working in Digital Library-related areas. Its main objective and goal was to develop the next generation of Digital Library technologies, based on sound comprehensive theories and frameworks for the life-cycle of Digital Library information. The currently active project DL.org continues several of the activities of DELOS. DL.org is a Coordination Action project, funded by the Commission of the European Union (EC) under the 7th Framework Programme ICT Thematic Area "Digital Libraries & Technology-Enhanced Learning" [4].

The DELOS Working Group dealing with the evaluation of digital libraries problem proposed a model [5, 6] that is broader in scope than the one usually adopted in the evaluation context. The aim was to be able to satisfy the needs of all digital library researchers, either from the research community or from the library community. This group started from a general-purpose definition of Digital Library and identified three non-orthogonal components within this digital library domain: the users, the data/collection and the chosen system/technology. These entities are related and constrained by means of a series of relationships, namely: (1) the definition of the set of users predefines the range and content of the collection relevant and appropriate for them; (2) the nature of the collection predefines the range of technologies that can be used; and (3) the attractiveness of the collection content with respect to the user needs and the ease of use of the technologies by these users determine the extent of usage of the digital library.

The general model proposed in [6] will be further supplemented by the Digital Library Reference Model Conformance Checklist which is a new part of the recently published Digital Library Reference Model [2] from project DL.org. Earlier versions of this Reference Model have been published (without the Checklist) by the DELOS project [1].

1.3 EuDML evaluation methodology

The four dimensions of evaluation 1) the context — why the evaluation is taking place: this has been described in the previous sections; 2) the construct — what is being evaluated: EuDML systems and services as outcomes of the EuDML project; 3) the criteria and 4) the methodology — how the evaluation will be carried out: this will be described in the following sections.

The DELOS group pays special attention to the relations between the components of a digital library, i.e. the relations User Content, Content-System and User-System. This is shown in Figure 1.

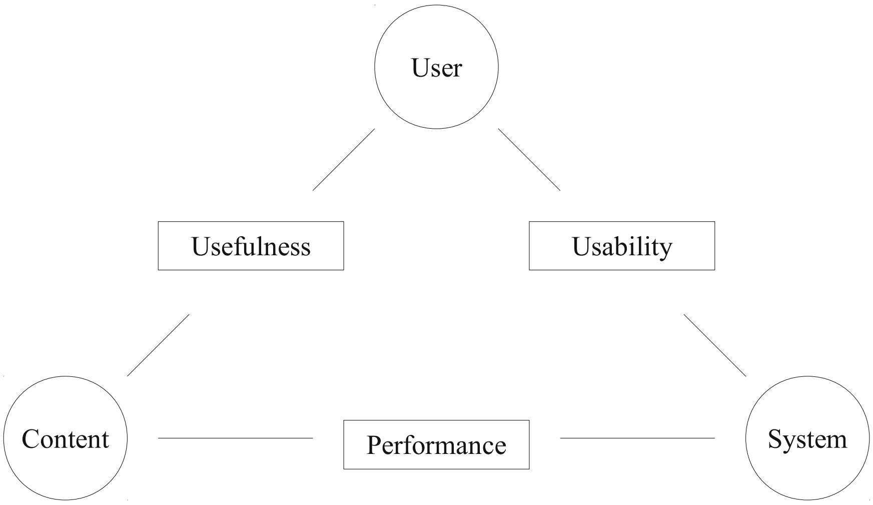

Figure 1: Digital library components and relationships

The Content-System pair is related to the performance attributes (precision, recall, response time, etc.), the User-System pair is related to usability aspects (effectiveness, satisfaction, etc.), and the User-Content pair is related to usefulness aspects. These pairs represent the relations/interactions among the digital library components and define a three-axes framework for the digital library evaluation. The following section on the evaluation framework for EuDML is structured according to this logical model when discussing various evaluation aspects. Of course, there are sometimes different places where to put a criterium in this framework, as these concepts are interrelated and sometimes overlapping.

The supplementary Digital Library Reference Model Conformance Checklist from project DL.org is given in appendix A of this report. It lists five domains of digital library services, according to [2]. These domains will be mapped to the relevant components and relations as outlined above.

2 Evaluation framework

The following subsections give general descriptions of the digital library components (content, system, users) and their relationships (usability, usefulness, and performance), according to the DELOS evaluation framework.

Additionally, references to the relevant sections of the Digital Library Reference Model Conformance Checklist are given where appropriate.

Special evaluation topics and evaluation criteria suitable and applicable to EuDML are listed. These are derived from the promises of the EuDML DoW, which have to be checked for conformance, and from relevant findings of the "Report on evaluation of existing services and content providers" (Deliverable 11.1 of the EuDML project), which provides benchmarks from previous evaluations of related services. Also, recommendations from the "Usability study" (Deliverable 6.1 of the EuDML project) will be considered.

2.1 Digital library components

2.1.1 Content

Content is the prime reason for interacting with a digital library. This component addresses the user's information needs. The relation between the user and the content strongly depends on the informational need of the user. The perceived usefulness of content is the first selection criterion for the user. During a search session, the users' perceptions of the usefulness of the content and their information needs can change and they may be forced to reformulate them and re-direct the information strategy (if any).

Content characterizes the digital library and affects the consequent processes dramatically. It is indispensably bound with the purposes of the digital library and the desired outcomes of its operation and serves the information needs of a community. The evaluation of content must take place under strict rules and standards that guarantee user access to high quality, appropriate-for-their-needs information. However, underneath this user-centered layer of evaluation lie numerous other issues; issues related to the nature, the structure and the administration of the content. Also, situational and contextual factors of digital library are important, such as organisational and group issues.

General issues

Content-related studies include:

- Content nature
- Form (text, images, video, sounds etc.)
- Language (language(s) used and whether the content is monolingual, bilingual, multilingual)
- Method of creation (digitized, born digital)
- Type (white, grey literature)
- Content structure
- Level (primary object, metadata)
- Size

- Cohesion (homogeneous or heterogeneous attributes usually apply at collection level and are connected with interoperability standards and technologies)
- Delivery mode (hypertext protocols, streaming technologies)
- Administration
- Collection growth (acquisitions, drop-outs)
- Rights (related to the license mode and content type, e.g. evaluating policies on acquisition and provision)
- License mode (open access, one-off purchasing, subscription-based with multiple variations such as e.g. moving walls)

Conformance Check List criteria

The relevant criteria regarding the content from the Digital Library Conformance Check List are those listed in appendix A.1 on page 23 (Content-oriented criteria).

EuDML performance indicators

Table 2 lists relevant performance indicators as given in the EuDML DoW.

Table 2: Performance monitoring — Content

No.	Objective/expected result	Indicator name	Year 1	Year 2	Year 3
1	Diversity of partnership (European dimension)	Number of participating content repositories (partners or associated partners)	5	8	12
2	Diversity of content (Discipline coverage)	Number of integrated collections (e.g. journal runs, book series, conference proceedings, Ph.D. theses)	50	100	200
3	Critical mass	Number of integrated resources (digital items reused from the network)	30,000	80,000	160,000

EuDML specific criteria

Specific criteria from the EuDML DoW are:

- Content consists of published texts holding mathematical knowledge that has been validated through a scientific editorial process (criterium to be understood in a broad sense).
- Content typically open access within 5 years of its publication date.
- Less than 5% of the whole content being subject to any form of restricted access.
- The network of documents is constructed by merging and augmenting the information available about each document from each collection, and matching documents and references across the entire combined library.
- Additional reporitories (in Europe) have been stimulated to join.
- European publishers have been persuaded to cooperate with the library, licensing it for open access with a reasonable moving wall licensing policy.

- Collections contain texts that would be qualified as belonging to physics, economics and social sciences, etc.
- Existing thesauri of mathematical keywords are used (Jahrbuch project, etc).
- Translation lists (Mathematical Subject Classification, UNESCO thesaurus, full text analysis) are used so that a query in one language can return documents in any other.

2.1.2 System

The system is governed by the rationale of the developers and service providers. It consists of various subsystems that perform different basic and auxiliary operations. Digital library systems generally include collections of multimedia digitized data and services that help storage, access, retrieval and analysis of the data collections. The final aim of a digital library system should be that of enabling people to access human knowledge any time and anywhere, in a friendly multi-modal way, by overcoming barriers of distance, language and culture, and by using multiple Internet-connected devices. In order to fulfill all the requirements on it, a digital library system must be the result of the integration of a number of different components. Discussions as how to implement each single component depends on the architectural choices, on the type of data access and retrieval, on the visualization and personalization of information, etc. The individual evaluation of such components can be carried out following evaluation standards already available in the literature. However, an important issue is the evaluation of how the individual components interoperate inside a digital library. The evaluation of a digital library system makes sense mainly if it is performed in relation with users and contents.

General issues

General technology-related issues are:

- user technology: document creation, disclosure, interface, browsing, search
- printing, group/individual information access: retrieval, navigation, filltering, extraction, text mining
- efficiency, effectiveness
- systems structure technology: repository, transport model (protocols)
- document technology: document model, format

Conformance Check List criteria

The relevant criteria regarding the system from the Digital Library Conformance Check List are those listed in appendix A.5 on page 30 (Architecture-oriented criteria).

EuDML specific criteria

Specific criteria from the DoW are:

- The services are accessible to humans through a web portal with many innovative discovery features, and to other machine services through a set of common digital libraries' interoperability protocols (namely Z39.50, SRU and OAI-PMH with an optional dedicated schema).

- EuDML tools are able to search, browse and exploit a distributed network of resources as if it were a single well-managed library.
- Metadata schemas from the content providers have been identified.
- A common EuDML metadata schema has been defined, and present a low barrier of entry for potential EuDML partners in the future.
- A framework to federate the metadata in a central metadata repository is in place.
- A metadata registry makes it possible to easily integrate new data providers with new metadata schemas, reusing the aggregated metadata in any new required schema.
- Discovered metadata are validated and merged with those already registered for the target items.
- Tools are in place to identify, from the Metadata Repository, items that may benefit from metadata enhancement and automatically apply such enhancement processes to them.
- As much of the full texts as possible has been converted to structured XML with MathML representation of formulas and English metadata.
- A set of dedicated tools has been packaged in order to generate metadata (structured textual OCR, mathematical OCR, keyword extraction, subject classification, bibliographic linking and citation, etc.).
- Common authority and interlinking structures have been developed.
- Tools and workflows to extract textual (coded in a proper XML schema) and mathematical (MathML) metadata (i.e. titles, keywords, authors, references etc.) from items in the content repositories, namely various types of mathematical documents, including scanned images, TeX/LaTeX sources, PDF documents, etc., are in place.
- Important requirements for interoperability were considered.

2.1.3 Users

Whatever the particular methodology chosen, when performing a user-oriented evaluation, the objectives are to acquire a deep understanding of the user requirements for technology design and planning, and to receive systematic user feedback throughout the design process.

- Types of users and their characteristics, such as different levels of knowledge and experience.
- The information needs of the users.
- Different satisfaction factors (functionalities and task accomplishment, etc.).
- Types of information handling strategies.
- Tasks and task procedures.
- Representation of work domains and environment.
- Collaboration between users and groups of users.

General issues

General user/use characteristics are:

- user (who)

- internal
- general
- education
- professional
- research
- domain (what) = subject area
- info seeking (how)
- object seeking
- browsing (rummagers, surfers)
- purpose (why)
- consume
- analyse
- synthesize

Conformance Check List criteria

The relevant criteria regarding the users from the Digital Library Conformance Check List are those listed in appendix A.2 on page 25 (User-oriented criteria).

EuDML performance indicators

Table 3 lists relevant performance indicators as given in the EuDML DoW.

Table 3: Performance monitoring — Users

No.	Objective/expected result	Indicator name	Year 1	Year 2	Year 3
7	Registered users contributing annotations	User's involvement	0	500	1,000
8	User's contribution	User generated content (annotations and external links)	0	500	10,000

EuDML specific criteria

Specific criteria from the DoW are:

- User surveys will be carried out in order to receive user feedback and better understand user needs.
- Replicated, synchronised management of user annotations to items in the content repositories are in place.
- Annotations can be comments, discussion threads, tutorials, reviews, reading lists, or other user contributed elements that can be attached to individual items in the collection.
- The user interface supports viewing of such content and the search engine supports searching within it.
- Annotations can themselves contain references to items and are analysed for the association analyser toolset.

Two rounds of user surveys will be carried out that will follow the structure and criteria of this report.

2.2 Component relationships

2.2.1 Usability and related studies (user-system)

Usability defines the quality of interaction between the “User” and the “System”. This helps the user to manipulate a system effectively, in an efficient and enjoyable way and to exploit all the available functionalities. A usable system is easy to learn, flexible and adapts to user preferences and skills.

General issues

General user interface related studies include:

- task-oriented interface for a digital library
- information sources
- information objects
- users and groups of users
- how well the different components and functionalities are integrated.

EuDML specific criteria

Specific criteria from the DoW are:

- EuDML is accessed via a web interface for human users, and a web service interface for tools and systems.
- Human interfaces all share the same graphical style and look and feel, promoting an “EuDML brand” which is multilingual and provides accessibility options to visually impaired and dyslexic users.
- A set of tools improves, to the extent that can be reached within this project’s duration and resources, the accessibility to the corpus for visually impaired users. Born digital content gas been converted to MathML and Daisy and made usable to Braille readers and text-to-speech engines.
- Explicit support is provided, using the latest technologies, for visually impaired or dyslexic users as well as automatic language translation support.
- Web 2.0 features are used.
- Insufficient metadata are augmented to a minimal level of quality among all integrated collections.
- Mathematical knowledge management techniques are applied to overcome language barriers and connect various items related by their subject.
- Machine interfaces are based in common standards, or in effective web-services, following the Representational State Transfer (REST) paradigm when relevant (or SOAP when recommended), and outputting common standards-based representations including XML (agreed schema), RSS and JSON.
- Functional interfaces and widgets have been developed making it possible to include a “EuDML Search Box” in other local systems and portals.
- A widget configuration facility has been developed, making it easy for users to create tailored search interfaces for their own websites. Other functional interfaces have been designed and implemented for services related to interoperability.

Recommendations from D11.1

Keep it simple! A clear and intuitive structure (less is more!) and consistent layouts for periodicals, browsing (e.g., like in NUMDAM) should be. The main focus should be on the results/contents.
Formulae should be display in MathML or TeX.

- Search
  - Advanced search should be comprised of boxes with assignable fields with combinable by selectable conjunctions (not too many categories, e.g., Euclid, author, title, anywhere, full text, subjects).
  - Phrase search and wildcards should be supported.
  - Short help and/or hints should be provided on the same page as the search form.
  - Sorting of results (year, title, author) would be preferable.
- Browse
  - Do not use too many categories. Browsing by author, or in periodicals by subject, is sufficient.
  - Within categories, as a refinement or alternative, browsing by title, author, subject should be given. Sorting for result lists should be provided by title, year, author.
  - Browse by title or an over-extensive list of subjects on a global level is clearly overkill.
- Provided information
  - For documents, again, keep it short and clear!
  - The following information should be provided: abstracts (preferably at least in English) with (if no references are provided) readable references therein, titles (original and in English), links to predecessor/successor, errata, other related documents and cited in.
  - MSC2010 classifications should be provided (also a possibility to clarify the meaning of them).
  - References should be included (authors, title, source in short, inside/outside link to document if available (Arxiv, DOI), ZBMATH, MSC2010).
  - For authors, a short information, number of papers in EuDML, etc., for periodicals, ISSN, ISBN, publisher, existing volumes in EuDML, a link to gain access to publications of the periodicals not in EuDML (external home page, publisher's page), related publications (continuation of another periodical, etc.) should be provided.
  - Preferably, when browsing, a short description on the periodical should be given at the top of each page. Further information should be provided as a link to a separate page or an outside page or the publisher.
  - Export of citations (BibTeX, etc.) should be offered.
- Connectedness/relations
  - Browse combined with search as search within certain categories (like within a periodical or within authors) would be preferable.

- Browse lists/ search results should not have too many entries per page by default; a possibility to change the number of entries should be provided.
- The amount of information should be restricted and be dependent on the type of list (search, browse and different types therein).
- Links should be intuitive and clear (e.g., not two leading to the same result). The title should lead to the detailed view (abstracts, etc.), the author name to a list of papers by the author. A full text link (e.g., Euclid, ZBMATH) should also be provided.

Recommendations from D6.1

Further recommendations from the EuDML Deliverable D6.1, Usability Study, are listed in appendix B and will have to be checked.

2.2.2 Usefulness (user-content)

Usefulness concerns the “User” and “Content” components. The usefulness of the content and its relevance to the user tasks and needs are the reasons behind the selection and usage of a digital library. This relevance is translated into actual relevance, type and level of resource relevance and task relevance.

Digital libraries can be used for many reasons, but the most central set of use cases focuses around information access. Finding certain types of content, retrieving specific information, locating known items, accessing material the client does not know enough about, there are many content-based, more or less goal-directed motivations that will lead a user to the access terminal of a digital collection of information.

Usefulness is the abstraction of every information need that stimulates user interaction with digital libraries. It lies between the content and the user needs and it reflects how users perceive the relevance of a digital library with their needs, the width, the breadth, the quality, as well as the validity of its collection, and the ability to serve their goals. The assessment of usefulness depends on the features of both the user and the content components.

General issues

Within a task-oriented evaluation perspective, aspects such as interaction and interactivity should be considered. Aspects that can be evaluated include:

- types of tasks and stages of a single task
- task performance and task procedures
- task complexity and task variations
- differences between domains
- impact on organizational levels
- types of information access systems and processing activities
- support for different access and usage strategies (e.g. analytical search, browsing, navigation, bibliographic search, collaboration, annotations.

Conformance Check List criteria

The relevant criteria regarding the system from the Digital Library Conformance Check List are those listed in appendix A.3 on page 26 (Functionality-oriented criteria).

EuDML performance indicators

Table 4 lists relevant performance indicators as given in the EuDML DoW.

Table 4: Performance monitoring — Usefulness

No.	Objective/expected result	Indicator name	Year 1	Year 2	Year 3
4	Enhancement progress	Number of supported resources (items with enhanced MathML full text and metadata)	10,000	30,000	90,000
5	Internal networking	Links between database objects	100,000	250,000	500,000
6	External networking	External links (generated for third parties using dedicated component, or detected through referrer website)	50,000	150,000	500.000

EuDML specific criteria

Specific criteria from the DoW are:

- Users can navigate by browsing the collections, following links to related items (same author, text citing or cited by the given one, similar subject, similar mathematical content, etc.).
- Users are guided by tips or additional keywords left by other users, and leave their own annotations as well.
- User profiles are derived in order to rank the query results depending on the user's mathematical background.
- Linked resources, such as Zentralblatt by partner FIZ can be used to explore further with other methods, bringing the user back when a reference to an interesting EuDML item is finally found.
- All item pages provide a link to the associated full text, 95% of which point to a partner's repository serving the file under open access. The remaining 5% will be hosted at their publisher's platform, possibly charging for access.
- A number of features will help locate related items, refine queries, and support quickly retrieving the most relevant documents.
- Content providers enjoy new services such as metadata capture, augmentation and merging, document and reference matching and cross-repository document linking.
- EuDML will make historical and comparative analysis, or even serendipitous discovery, of the development and achievements of European mathematics much easier.

- It is an essential feature of EuDML to address the difficult question of providing access to articles based on their subject, or scientific meaning, rather than on their language.
- Interlinking is used as a powerful access tool to the mathematical resources regardless of their language, but rather according to their subject and/or their scientific importance.
- A link to a documents review in one of the reviewing databases (Jahrbuch, Zentralblatt MATH, Math. Reviews) is given.
- The interlinking infrastructure deployed in the project allows the exploitation of links to and from other related resources, such as citations from reviews and from subsequent works.
- Webs of citations provide a language-neutral way to find resources across multiple languages.
- A ‘social networkâĂŹ designed for each work in EuDML makes it possible to offer a powerful scenario for resource discovery by serendipity.
- Mathematical content is encoded in a semi-structured format.
- Mathematical knowledge management techniques will be solicited to assess its novel technologies such as mathematical OCR, XML/MathML full-text generation from (La)TeX source files or PDF, formula representation and searching, and mathematical similarity metrics.
- Tools are in place to identify various types of referential and semantic connections between different items in the EuDML Metadata Repository, and also between such items and external resources.
- Simple navigation and browsing through networks of related and interconnected documents and linking of elements (e.g. names of people, theorems or concepts) in these items to and from external resources such as encyclopaedia entries (including Wikipedia), historical information and cultural references is possible.

2.2.3 Performance (system-content)

Performance is placed between the “Content” and the “System”. Although it is of crucial importance, this is an aspect of the system that the user cannot see or evaluate directly. The performance of the system depends strongly on the formats, structures and representations of the content.

The performance also depends on the technical parameters closely related to the system’s work and to the Internet connection. Several system performance indicators measuring the interaction between the EuDML platform and the EuDML content providers systems could also be tracked.

Several measurable system performance parameters are:

- the response time upon invocation of a function
- the presence of delays (with performance indicators more tailored to the multimedia and streaming contexts)
- precision and recall (with performance indicators in the information retrieval)
- performance time of parallel queries processing
- statistics on system usage over time

- statistics on content growth over time
- data provided by monitoring/self-monitoring and reporting mechanism (may be implemented internally in the digital library software).

Several measurable Internet performance parameters are:

- internet traffic/speed (bandwidth, throughput) — Network bandwidth represents bit rate measure of available or consumed data communication resources. Throughput or network throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node
- number of parallel queries and Internet connections (system load).
- threshold of performance causing denial-of-services (denial-of-service and availability service are critical to performance specifications (requirements)
- other network performance parameters related to services quality and availability.

Performance testing could serve different purposes: It could demonstrate if the system meets performance criteria. It could compare two or more parameters in certain conditions in order to find in which conditions the system performs better. It also could measure what parts of the system or workload causes the system performs badly.

Content providers systems are web applications executing on particular web. So it could be used tools estimating a web server performance and running on web applications including overall performance, response time and delays. Some of the tools and software for load/performance testing are:

- ApacheBench (or ab), a command line program bundled with Apache HTTP Server JMeter a Java Open Source software to make scenarios for several protocols: HTTP, AJP, JDBC and different load types.
- Curl-loader a powerful loading, testing and benchmarking open-source tool.
- OpenSTA, distributed software testing architecture which can perform scripted HTTP and HTTPS heavy load tests with performance measurements from Win32 platforms.
- HTTP Test Tool a script driven command line tool.
- Httperf (HTTP performance measurement tool) — while running, it keeps track of a number of performance metrics that are summarized in the form of statistics that are printed at the end of a test run.
- Online tools for measuring website performance.
- mtr — a network diagnostic tool (Unix, Linux), combining the functionality of the traceroute and ping programs in a single network diagnostic tool.

Appendix C contains extended list of performance test tools with detailed descriptions and basic requirements.

Key parameters estimating a web server/web application performance

The performance is usually measured in terms of:

- number of requests that can be served per second (depending on the type of request, etc.)
- latency response time in milliseconds for each new connection or request
- throughput in bytes per second (depending on file size, cached or not cached content, available network bandwidth, etc.).

The measurements must be performed under a varying load of clients and requests per client. It also has to be determined the performance goals — the criteria that the team wants to meet.
The key parameters in performance measuring for system-content in the context of EuDML are:

- Internet speed (bandwidth, throughput) between EuDML aggregator and content providers. If there are available any statistical reporting or system services providing information on usage and workloads. It can be very useful for time scheduling for EuDML aggregator when workload is less and than retrieve content and harvest metadata and etc.
- HTTP performance — It is important because EuDML platform uses OAI-PMH protocol which transfers xml structured data over http protocol.

2.3 Policies

On top of digital library components and relations as described above lie several policies that govern the deployment, functioning, and use of the EuDML digital library. Although it is not the central aim of the EuDML project to develop those policies, there are several policies already implicit to project goals (e.g. eventual open access to full-texts), and others will be imposed onto users by the services that are developed (use policies). A set of criteria to evaluate these policies is given in the Digital Library Conformance Check List, see appendix A.4 on page 28 (Policy-oriented criteria).

3 Evaluation roadmap

In this section, an explanation is given on how the evaluation framework that has been defined above will actually be used during the evaluation task. According to the EuDML DoW, the structure of the evaluation exercise (in terms of project deliverables) is:

- D11.2 EuDML assessment and evaluation plan (this report; month 15)
- D11.3 EuDML assessment and evaluation — First report (month 24)
- D11.4 EuDML final assessment and evaluation — Final report (month 36)

3.1 Two rounds of evaluation

As defined in the DoW, two evaluation rounds will have to take place. The first round can be seen as evaluating the intermediary outcomes and prototypes of the project with the goal of providing actual guidance and recommendations for further development during the project. In particular, weak points and open issues need to be identified and weighed up.

The second round, which will be completed near the end of the project, can be seen as an assessment of final project results, with the aim of guiding further actions beyond the lifetime of the project. In particular, issues of sustainability of EuDML services are of importance in this final evaluation.

3.2 Evaluation teams

Both evaluation rounds will be structured according to the structure of the evaluation framework. Evaluation teams will be formed which will examine the areas

1. Content
2. System
3. Users
4. Usability
5. Usefulness
6. Performance
7. Policies

according to the criteria stated in the previous section.

The teams will be formed from project partners that participate in the Assessment and Evaluation Work Package of the project, in a way that the required expertise and diverse topical background is adequately represented. Teams will have to be formed during a project meeting in summer 2011.

3.3 Production of evaluation reports

An evaluation coordinator (nominated by the evaluation work package leader) will compile lists of evaluation criteria for each area from the criteria listed in the previous section and the relevant appendices. The teams will assess and evaluate EuDML achievements according to these criteria and produce evaluation reports for the particular areas. These will be integrated into a clearly structured report deliverable by the evaluation coordinator.

3.3.1 Performance evaluation

According to Section 2.2.3 there are two cases that should be taken into account. In the first one the system is EuDML system itself and the content is stored and processed inside EuDML system. The second case is related to (EuDML system) — (content provider) where the difference is that content is remotely stored and retrieved from external software systems via Internet. Therefore performance parameters and measurement tools are closely related to the Internet and network performance.

System-content case

The case of performance for system-content is concerning processes inside the EuDML system. Therefore for precise performance measurement it should be implemented internal system functions reporting current performance and usage. For example the user can do search by some keywords/phrase. With the search results additional information is returned from the system's reporting functions. The additional information gives to the user the current total amount of searched records, what time in milliseconds is spent for generated results, the number of results, etc.

EuDML system-content provider case

The key parameters in performance measuring for system-content in the context of EuDML are: Internet throughput and http performance. The starting point of performing tests should be the server where EuDML platform is running. These tests should be done from EuDML server to each content providers system (as it is depicted on the following figure). Public Internet cloud contains many different nodes, routes and ISPs (Internet Service Provider) infrastructure. It have to be performed multiple tests in order to be collected enough data for statistical analysis.

Necessary functions for performance evaluation tools

To evaluate network system performance from the point of view of usability, the system administrators must know how their services are working and must improve them to satisfy user requests. Network system performance with regard to usability is determined by how the client provides performance to the user and that is it. The system administrator should be aware of client system performance factors such as:

- How long the client takes to access the server.
- How long the client takes to process the transaction.
- How much data is throughput in the client.

We recommend a common framework to evaluate the performance of the end-point application. The evaluation tools should have the following characteristics:

1. *The tool is able to measure the throughput and response speed at the end-point applications, which has an impact on the user.* The performance evaluation tool aims to improve the performance of the end-point application. The total system performance doesn't always interrelate with the performance of the network path. Therefore, the network system performance should be evaluated in the client applications.

2. *The tool should handle various kinds of datalinks.* The Internet has been used on various datalinks such as Ethernet, Integrated Services Digital Network (ISDN), and so on. The transmission Control Protocol/Internet Protocol (TCP/IP) technology is a set of protocols of the upper layer of these datalinks. Therefore, the measurement method should be independent of the datalinks.
3. *The measurement tool should be independent of applications.* There are various applications and application protocols utilized in the Internet. The performance measurement should be a standard framework, and it should not be dependent on a single application and a single application protocol.
4. *The measurement tool should be able to be applied to existing applications without any modification.* It is costly to modify application software to operate the measurement tool. Also, there is a number of applications that would be difficult to modify for use with the measurement tool.
5. *The measurement tool should be able to be applied to running systems.* Using computer simulation, it is difficult to calculate all performance factors, and the benchmark is only valid under specified conditions. It is more effective to evaluate running systems by measuring the performance in the actual situation.
6. *The measurement tool should be able to be applied not only to the network links, but also to the server and client systems.* The total performance of the network systems is affected by the servers, the clients, and the network links.

Existing performance measurement tools

Appendix C contains complete list of tools with short descriptions and basic requirements. The following several tools are the most general and well known, they have been developed to evaluate network performance.

1. *Statistical analysis of servers logs.* Statistical analysis from server access logs allows us to determine the operation status of the servers, such as the number of accesses, number of data transfers, and processing time. However, the results of the analysis of server logs determine only the performance of the servers themselves. The performance of the clients and the network links is not included in the results.
2. *Measurement of network usage and Round Trip Time (RTT).* Simple Network Management Protocol (SNMP) is widely used to measure network usage. System administrators can use simple tools such as ping and traceroute to measure system usage. However, TCP performance degrades as network usage increases, and it is also affected by the characteristics of the network links. The performance of end-point application is affected by not only the network usage but also by the characteristics, end-to-end throughput, and capacity of the servers and clients.

Accordingly, we need to choose particular performance evaluation tool/s for network systems that can be applied to actual systems operation under various configurations.

Second evaluation round

During the second round the performance evaluation tests are executed again. The results will be compared with the data from the first round evaluation (results and recommendations for performance optimizations). It could also give information about further planning for system scalability regarding EuDML platform and content providers.

Appendices

A Evaluation criteria from the DELOS Digital Library Reference Model

The DL.org Digital Library Reference Model [2] aims at contributing to the creation of foundations. It exploits the collective understanding on Digital Libraries that has been acquired by European research groups active in the Digital Library field for many years, both within the DELOS Network of Excellence and beyond, as well as by other groups around the world. It identifies the set of concepts and relationships that characterise the essence of the Digital Library universe. This model can be considered as a roadmap allowing the various players involved in the Digital Library domain to follow the same route and share a common understanding in dealing with the entities of such a universe. In the case of the present evaluation, it is used to guide the selection of the evaluation criteria, as specified below.

The Digital Library Conformance Check List, which is Part IV of the DL.org Digital Library Reference Model provides assessors with a set of criteria which can be used to determine whether a digital library complies with the DL.org Reference Model. These criteria are clustered into mandatory, recommended and optional criteria.

The following set of criteria results from an analysis of the Digital Library Reference Model concepts and relationships. These criteria have been selected because of their discriminating power with respect to defining whether a ‘digital library’ conforms to the characterisation of such systems as envisaged by the Digital Library Reference Model. The presentation of the criteria is structured according to the six Reference Model domains characterising the digital library service (Content, User, Functionality, Policy, Quality and Architecture).

A.1 Content-oriented criteria

The following criteria have been selected to verify whether or not the ‘digital library’ conforms to the Digital Library Reference Model from the Content domain point of view.

Mandatory

Regardless of the type of Content a ‘digital library’ was conceived to hold, it must meet at least the following criteria:

- The Digital Library must manage a set of Information Objects and the set cannot be empty. By definition the purpose of a digital library is to collect, manage and preserve in perpetuity digital content.
- Every Information Object must have a unique identifier (Resource Identifier). This guarantees that each Information Object managed by the ‘digital library’ is distinguishable from the remaining ones in the context of the same ‘digital library’
- Every Information Object must have at least one element of Metadata associated with it. This ensures that each Information Object is equipped with data supporting its management and use.

- Every Information Object must belong to at least one Collection. This guarantees that the overall set of Information Objects managed by the ‘digital library’ pertains to an organized body.
- Every Collection must have a unique identifier (Resource Identifier). This establishes that each Collection managed by the Digital Library is distinguishable from any others in the context of the same Digital Library.
- Every Collection must have at least one element of Metadata associated with it. This asserts that each Collection is equipped with data supporting its management and use.

Recommended

Additional desired properties of a ‘digital library’ are:

- Every Information Object should conform to an explicit and known format (Resource Format).
  This guarantees that the system is aware of the “structure” each Information Object conforms to and that this structure is publicly declared thus making the Information Object usable by third party actors whether human or machine. The notion of Resource Format is wide and might range from an abstract one (e.g. “enhanced publication” to a concrete one (e.g. PDF).
- Every Metadata should conform to an explicit and known format (Resource Format).
  This criterion — a specialization of the previous one — ensures that the system is aware of the “structure” the metadata object conforms to and that this structure is publicly declared so that it can be used by third party actors whether human or machine. In this case the notion of Resource Format corresponds to the notion of metadata schema.
- Every Annotation should conform to an explicit and known format (Resource Format).
  This criterion guarantees that the system is aware of the particular “structure” to which the annotation object conforms. Being publicly declared, this structure can be used by third party actors whether human or machine.
- Every Collection should have a well-defined intension, i.e., the set of criteria characterising Collection membership, and should have a well-defined extension, i.e., the set of Information Objects belonging to the collection.
  The collection concept is fundamental to keep the set of Information Objects organised. Because of this, it is recommended that both the set of Information Objects belonging to a collection and the criteria driving the membership of an information object into a collection are clearly defined.
- Every Information Object should be regulated by Policies.
  Policies are essential to establish conditions, rules, terms or regulations governing the management of information objects.

Optional

Finally, a ‘digital library’ may also meet the following set of criteria:

- An Information Object may have multiple Editions each represented by a different related Information Object.
- A ‘digital library’ might be employed to manage multiple editions of the same work. In these cases it is important to deal effectively with the edition concept.
- An Information Object may have multiple Views each represented by a different related Information Object.
- A ‘digital library’ might be called to manage multiple “views”expressions” of the same conceptual work. In these cases it is important to properly deal with the view concept.
- An Information Object may have multiple Manifestations each represented by a different related Information Object.
- A ‘digital library’ might be called to manage multiple “items” of the same conceptual work or view. In these cases it is important to properly deal with the manifestation concept.
- An Information Object may be compound, i.e., it may consist of multiple Information Objects.
- Modern ‘digital libraries’ are usually expected to deal with emerging forms of “documents” Very often such a “documents” consists of aggregates of other objects (of different media).
- An Information Object may be associated with other Information Objects for a certain Purpose.
- Managing compound objects may require linking other objects. The motivations leading to linking are diverse and context specific, e.g., citation and lineage.
- An Information Object may have multiple elements of Metadata associated with it.
- Metadata are a type of Information Object intended to support the management and use of the Information Objects to which they are attached. Different metadata can be conceived to support diverse needs. The majority of ‘digital libraries’ tend to deal with a single metadata format.
- An Information Object may be associated with multiple Annotations.
- Annotations are kinds of Information Objects that are attached to existing Information Objects for various purposes including objects enrichment and cooperative working.
- A Collection may be associated with multiple Metadata.
- According to the Reference Model, Collections are a type of Information Object. Because of this, they inherit all the features of Information Objects and benefit of multiple metadata. A Collections may be associated with multiple Annotations.
- According to the Reference Model, Collections are a type of Information Object. Because of this, they inherit all the features of Information Objects and benefits of multiple annotations.

A.2 User-oriented criteria

The following criteria have been selected to verify whether or not the ‘digital library’ conforms to the Digital Library Reference Model from the User domain point of view.

Mandatory

Regardless of the type of Users a 'digital library' is conceived for, it must meet at least the following criteria:

- The Digital Library must serve a clearly identified set of Actors and this can not be an empty set. Actors represent the entities that interact with any digital library 'system' i.e., humans and inanimate entities such as software programs or physical instruments. This guarantees that they exist, i.e., there is no 'digital library' without users interacting with it.
- Every Actor must have a unique Resource Identifier. This guarantees that every Actor is distinguishable from other Actors in the context of the same 'digital library'
- Every Actor must be described by at least one Actor Profile. This guarantees that every Actor uses the 'digital library' and interacts with it as well as with other Actors in a personalised and codified way.
- Every Actor must act with at least one Role. This guarantees that an Actor cannot interact with a Digital Library if its role is not specified.
- The set of managed Roles must include the DL Manager Role. DL Managers are Actors that exploit DLMS facilities to define, customise, and maintain the DL service.
- The set of managed Roles must include the DL Software Developer Role. DL Software Developers exploit Digital Library Management System facilities to create and customise the constituents of the Digital Library Systems.
- The set of managed Roles must include the End-user Role. This guarantees that the digital library supports at least typical end-user roles, like content consumers, content creators or digital librarians.

Recommended

Additionally, a Digital Library should meet the following criteria:

- Every Actor should perform Actions that apply Functions and concern Resources.
- Every Actor that interacts with a digital library should be able to perform certain Actions that involve the application of Functions and specific Resources.

Optional

Finally, a Digital Library may meet the following criteria:

- Actors may belong to more than one Group.
- During the interaction of an Actor with a Digital Library, the Actor may communicate or collaborate with other Actors that belong to various Groups; thus, the specific Actor may participate in different Groups. The concept of Group in the User domain has commonalities with the concept of Collection in the Content domain, it is a mechanism to organise Actors.

A.3 Functionality-oriented criteria

The following criteria have been selected to verify whether or not the 'digital library' conforms to the Digital Library Reference Model from the Functionality domain point of view.

Mandatory

Regardless of the type of Functionality a ‘digital library’ is conceived for, it must meet at least the following criteria: The Digital Library must offer a clearly identified set of Functions and this can not be an empty set.

- The purpose of the DL is to offer functions, i.e., a particular processing task that can be realised on a Resource or a Resource Set as the result of an Action of a particular Actor.
- Every Function must have a unique identifier (Resource Identifier). A Function is a Resource, thus it must be identified by a persistent identifier if it is to be distinguished from other Functions managed by the DL.
- Every Function must be performed by Actors. DL Functions are the implementations of functions and services enabling Actors to interact with the DL.
- Every Actor must be provided with Functions to Access Resources. The DL must implement functions to enable actors to access, e.g., discover, acquire and visualize, all types of Resources (Information Objects, Actors Profiles).
- Every Actor must be provided with Functions to Discover Resources. Actors must be able to find the desired Information Objects, search and access not only the DL Content, but also other Actors or Functions.
- Every DL System Administrator must be provided with Functions to Manage & Configure DLS. DL must implement functions for handling the DLS and configuring its settings.

Recommended

Additionally, a Digital Library should meet the following criteria:

- Every Function should be able to interact with other Functions.
  DL functions should exchange information with other functions regulating their behaviour and performance.
- Functions to Acquire Resources should be provided.
  DL functionality should enable Actors to retain Resources e.g., Information Objects and Actor Profiles, in existence past their interaction with the Digital Library System.
- Functions to Browse the Resources should be provided.
  DL should implement services enabling Actors (virtual or real) to browse the available DL content, user profiles, policies, etc.
- Functions to Search the Resources should be provided.
  Actors should be able to look for specific objects held within the DL by expressing queries and by entering specific keywords and constraints.
- Functions to Visualize the Actor’s requested Resources should be provided.
  A DL should deliver to Actor the requested information using the appropriate visualizations to produce comprehensive and well-presented objects, lists and query result sets.
- Functions to Manage Information Object(s) should be provided.
  A DL should implement functions to handle, i.e., disseminate, publish, process, analyze and transform, the Content of the DL, i.e., Information Objects.

- Functions to Manage Actor(s) should be provided.
  A DL should implement Functions to establish registered actors, personalize their preference and apply user profiles.
- Functions to Manage DL specific domains in a large scale should be provided.
  The DL should implement services and mechanisms to handle DL domains as a whole, e.g., Manage (import, export) all the Content of DL rather than handling each Information Object individually.

Optional

Finally, a Digital Library may meet the following criteria:

- Functions may depend on the Actor's Profile who invokes them.
- DL Functions that are offered to the Actor(s) may be customized according to his/her profile, DLS role and rights and /or personal preferences.
- Functions may consist of other parts, i.e., sub-functions.
- Functions may be organized in arbitrarily complex workflows, based on composition and linking facilities.
- Functions may be enriched with Metadata and Annotation.
- DL Functions may have a description, which tells what the function does and how a system or human can interact with it.
- Functions may enable Actors (virtual or real) to Collaborate with each other.
- Actors (virtual or real) may act as peers who are able to communicate, share and exchange information collaboratively.

A.4 Policy-oriented criteria

The following criteria have been selected to verify whether or not the 'digital library' conforms to the Digital Library Reference Model from the Policy domain point of view.

Mandatory

Regardless of the type of Policy a 'digital library' is conceived for, it must meet at least the following criteria:

- The Digital Library must be regulated by a clearly defined set of Policies and this can not be an empty set. Policies are a set of conditions, rules, terms or regulations governing the operation of the DL;
- Access Policies must regulate the use of the Digital Library by Actors. Access Policies are essential to establish conditions, rules, terms or regulations governing the interactions between the Digital Library and Actors.
- Every Policy must be addressed at least to an Actor. Defining the recipients of a Policy ensures the interaction between the Digital Library and its Actor(s).
- Every Policy must have clearly defined scope(s) and characteristics (Policy Quality Parameter). A Policy must have defined objectives and consequences affecting the DL system as a whole, a certain domain, a specific task or entity.

Recommended

Additional desired properties of 'digital library' policies are:

- Every Policy should be expressed by an Information Object.
  The digital representation of a Policy ensures its controlled description, management and use within the Digital Library. This representation enables automatic enforcing. Moreover, it is a prerequisite for a series of other automatic actions including policy comparison, policy reconciliation and policy interoperability.
- Every Policy should have a unique identifier (Resource Identifier).
  The use of a persistent identifier ensures that each DL Policy is distinguishable from the others in the context of the same Digital Library.
- Every Policy should have a known format (Resource Format).
  The implementation of a Policy in a known format guarantees that the system is aware of which "structure" each Policy conforms to and that this structure is publicly declared as to be used by third party actors whether human or machine.

Optional

Finally, a Policy governing a 'digital library' may also meet the following criteria:

- A Policy may regulate the service of the system as a whole (System Policy).
  Generic processes within the 'digital library' may be regulated by policies.
- A Policy may regulate functionalities related to Content (Content Policy).
- A Policy may regulate processes related to the Content domain.
- A Policy may regulate DL Functions (Functionality Policy).
  DL Functions' lifetime and behaviour may be governed by specific policies.
- A Policy may regulate User profiles and behaviour (User Policy).
- A Policy may regulate processes related to the User domain.
- A Policy may be extrinsic (Extrinsic Policy).
- A Policy may be imposed from outside the organisation domain of the 'digital library' e.g., by wider organisations regulating the Digital Library itself, by national and international laws, or by customs.
- A Policy may be intrinsic (Intrinsic Policy).
  The Policy governing the Digital Library may be defined and determined by the Digital Library organisation itself. Intrinsic Policy manifests the Policy principles implemented in the DL. A Policy that is defined by the DL or its organisational context that reflects the organisation's mission and objectives, the intended expectations as to how Actors will interact with the DL, and the expectations of Content Creators as to how their content will be used.
- A Policy may be implicit (Implicit Policy).
  The Policy governing the Digital Library may be inherent by accident or design. Implicit Policies usually arise as a result of ad-hoc decisions taken at system development level or as a consequence of the inadequate testing of a DLS that results in an interaction of Policies leading to unintended policy deployment.
- A Policy may be explicit (Explicit Policy).
  Explicit Policy is a Policy defined by the DL managing organisation and reflecting the objectives of the DL and how it wishes its users to interact with the DL. The

implementation of an Explicit Policy at the Digital Library Management System level corresponds to the definition and Actor expectations.

- A Policy may be prescriptive (Prescriptive Policy).
  The Policy governing the Digital Library may constrain the interactions between DL Actors (virtual or real) and the DL. Prescriptive Policies can cover a broad range of Policies from the kind of Function to which specific types of Actors can have access, to those that govern Collection development.
- A Policy may be descriptive (Descriptive Policy).
  Descriptive Policies are used to present the aspects of a particular Policy in the form of explanation. A Descriptive Policy is a Policy that describes modes of behaviour, expectations of Actor interaction, collecting and use guidelines, but which do not manifest themselves through the automated application of rules, as a Prescriptive Policy does.
- Policy may be enforced (Enforced Policy).
  The Policy governing the Digital Library may be deployed and strictly applied within the DL. An Enforced Policy is a Policy applied consistently and strictly in the DL. Monitoring and reporting tools are necessary to follow up how the Policy is being applied.
  A Policy may be voluntary (Voluntary Policy).
- The Policy governing the Digital Library may be monitored by an actor (human or machine). Voluntary Policy basically means a Policy that is followed according to the decision of the Actor. This is valid for all Policies for which its application is a matter of choice. In some cases, users may comply with Policies that are not officially communicated.
- A Policy may be compound.
  A Policy may be organised in arbitrarily complex and structured forms. A compound policy may be obtained by properly combining constituent Policies.

A.5 Architecture-oriented criteria

The following criteria have been selected to verify whether or not the ‘digital library’ conforms to the Digital Library Reference Model from the Architecture domain point of view.

Mandatory

Regardless of the content, user, functionality, policy and quality characteristics of the ‘digital library’ the Digital Library System supporting its operation must meet at least the following criteria:

- The Digital Library System underlying the ‘digital library’ must have a well-defined Software Architecture. The Software Architecture describes the digital library system enabling software by clearly defining how it is structured in components, i.e. programmes, how they communicate and are interrelated to offer the digital library service.
- The Digital Library System underlying the ‘digital library’ must have a well-defined System Architecture. The System Architecture is the conceptual model that de-

fines the organisation and relations between the Hosting Nodes, i.e. the (virtual) hardware environments hosting and running the Software Components, and the Running Component, i.e. the running instances of a Software Component active on a Hosting Node.

- Every Architectural Component must have a unique identifier (Resource Identifier, identifiedBy). The use of a persistent identifier ensures that each DL Architecture Component is distinguishable from the remaining ones in the context of the same Digital Library System.
- The Software Architecture must consist of at least one well identified Software Architecture Component. The Software Architecture must include at least one Component, i.e. a software package, a web service, or a module, with well-defined interfaces, that encapsulates a set of related functions (or data).
- The System Architecture must consist of at least one Hosting Node and one Running Component. The System Architecture of a DLS is implemented by a set of components (System Component) running on servers which act as Hosting Nodes. The resulting system organisation (i.e., Software Components used and resulting Running Components and Hosting Nodes) can evolve over the time. A single Running Component hosted by a single Hosting Node corresponds to the minimal System Architecture structure.

Recommended

Additional desired properties of a ‘digital library’ (its Digital Library System) are:

- The ‘digital library’ service is deployed and operated by means of a Digital Library Management System.
  The Digital Library Management System facilitates the set up and maintenance of DL Systems by offering facilities for their production and administration. These facilities also assure a well-defined Quality of Service for the managed DL Systems.
- Every Software Component should be regulated by a License.
  The License is a particular policy which specifies the permission on use, re-use and modification of the Software Component.
- The Software Architecture should be composed of more than one identifiable Software Architecture Components.
  A component-oriented approach for digital library systems offers many advantages with respect to system building, openness, and evolution, and it is thus preferable to other solutions especially for large systems.
- The System Architecture should be composed of more than one identifiable System Architecture Components.
  A System Architecture based on a number of running components distributed on different hosting nodes offers many advantages with respect to system building, openness, and evolution, and it is thus preferable to other monolithic solutions especially when dealing with large systems.
- Every Architectural Component should conform to a Framework Specification.
  Architectural Components should interact through a Framework Specification. The Framework Specification prescribes the set of Interfaces to be implemented by the

components and the protocols governing how components interact with each other. In so doing, it facilitates components composition and interoperability.

Optional

Finally, a ‘digital library’ (its Digital Library System) may also meet the following set of criteria:

- Every Architectural Component, be it a Software Architecture Component or a System Architecture Component, may exploit (use) one or more other not conflicting Architectural Components.
  The exploitation of functionality offered by other components is a very common practice in software engineering. It reduces the complexity of the problem to be deal with and favours reusability.

B EuDML D6.1 Usability Study recommendations

This section lists the recommendations as they have been made in Deliverable D6.1 “Usability Study” of the EuDML project. In that report, several digital library sites were evaluated from a usability point of view. Several specific recommendations for EuDML systems and services have resulted from this evaluation, and these recommendations will now be used as evaluation criteria for EuDML itself.

The section headlines below correspond directly to the headings from the “Results” section of the Usability Study report.

Open Task

- We must recognise that entering the EuDML site via the home page may well be the exception rather than the norm. For this reason the site should have clearly tailored ‘landing pages’ for each journal, volume, paper and author, designed to catch searches on search engines and other indexing sites.
- We should pay special attention to ensuring that our metadata and link structures are attractive and interoperable with Google and Google Scholar.
- We should recognise that we are only able to own a small amount of this space due to the limited number of papers and publications that EuDML will contain. Certain features we might like to implement to make EuDML best-of-breed may in fact be impaired by the amount of content within the Library. In these circumstances we should integrate with or link to other platforms.

Field of Study Task

- We should consider citations to be a critical navigation feature. We should look to replicate and improve upon Google Scholar’s approach to citation indexing. Where possible we should include citations, even if the related papers are not within EuDML itself, this will increase the perceived usefulness of the DML.
- We should research Mendeley as a possible platform for delivering some of the more social aspects of EuDML that we are considering under ‘Annotations’.
- We should consider adding obvious links to Authors’ home pages.
- The ability to quickly and easily get to a page of results is perhaps more important than the ability to refine a search in precise detail.

Finding and Browsing Journals

- Provide an easy-to-use listing of all journals on the site, and a dedicated page for each journal, with a list of volumes and an easy search to find any paper from that journal. Link back to this page from any paper for which this journal is the source.
- Journal ‘landing pages’ should also appear in general search results when they provide a close match for a search. Users expect search to work like Google and don’t have a mental map of the underlying database schema that separates papers and journals.
- Consider alpha-listing journal titles beginning with the phrase ‘journal of’ by the third word.

- Call the journal list link ‘Journals’ rather than ‘Browse’ or similar.
- Have a separate page for each volume (listing papers) and a top-level listing page ordering from most recent to oldest.

Search

- Consider adopting the boolean search style, but hiding chained boolean search fields until the user indicates their wish to add an additional filter (see figure 17). This offers the following benefits: The relationship between the field, the boolean operator and the next field are made explicit by the interface. The transition between ‘simple’ search and ‘advanced’ search is simple and subtle, rather than jarring: You just add extra filters.
- Include a NOT operator.
- Support the following operators within the search, this could be achieved with synonyms and compilation to SQL:
  - ”exact phrase”
  - AND & +
  - OR |
  - Go* matches Gordon
  - published>1975 & (Author:*Frey | Author:*)
- Alternatively: Consider adopting Google’s approach. Certain operators are supported, but the search is mostly over a full-text index, and the ‘advanced search’ field provides a clear indication of how to perform boolean text searches.
- Include a ‘sort by’ control on the search, and on the results page.
- Enabling searching with LaTeX is more important to mathematicians than enabling searching with MathML.

Results Pages

- Link the title of the paper, either directly to the paper, but more likely to an intermediary page which features the abstract, the download, references and citations, and various available formats. This will keep the results page cleaner and more Google-like and provide a natural, indexable ‘home’ for each paper.
- Each result from the list page should feature a 30 word excerpt from the abstract, with keyword-search terms highlighted in bold. This provides reassurance that results are relevant and allows the user to scan the results quickly to identify relevant papers. If keywords are identified in the abstract, use the 30 words surrounding those keywords, if not use the first 30 words.
- PDFs with BiBTex for citations should be clearly promoted as the key option. Other formats should be promoted less.
- Clicking on authors within search results should lead to a dedicated page on that author where available, with a list of papers by them, or otherwise a search for that author name. Where possible that page should include further information about the author and a link to their home page.

Research & Lists

- Consider making journals a faceted navigation filter on sets of results.
- Consider integrating with CiteSeer for citations.
- Where using standard classifications include a link to the standards body. Link the written description of the classification as well as the classification code.
- Careful consideration needs to be given to keywords. In order to establish trust in keywords a relatively complex user trust system may be required. We may need to question whether we will get a quantity of responses based upon expected user volumes that will make this effective. Otherwise we may consider automatic keyword generation algorithms.
- If we can find effective ways to associate keywords with papers we should consider using them as a faceted navigation filter within results pages. i.e. display high-frequency keywords within the overall result set which can be switched on as optional search filters (see delicious.com).
- A basket or folder feature may be helpful, but should be viewed in the context of the library. Regardless of whether we implement a save for later feature or not, result pages should always include a full GET query string in the URI so that pages of results can be bookmarked and emailed.

Authors

- Ideally authors should be normalised, modeled and have synonyms (including foreign accents etc.), making Author Searching less hit and miss.
- This would also enable us to include a mini-profile page. This can list all of the author's papers on the website and link to author's home page where known.

C Performance test tools

Allmon

Description: The main goal of the project is to create a distributed generic system collecting and storing various runtime metrics collections used for continuous system performance, health, quality and availability monitoring purposes. Allmon agents are designed to harvest a range of metrics values coming from many areas of monitored infrastructure (application instrumentation, JMX, HTTP health checks, SNMP). Collected data are base for quantitative and qualitative performance and availability analysis. Allmon collaborates with other analytical tools for OLAP analysis and Data Mining processing.
Requirements: Platform independent

Apache JMeter

Description: Apache JMeter is a 100% pure Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions. Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.
Requirements: Solaris, Linux, Windows (98, NT, 2000). JDK1.4 (or higher).

benerator

Description: benerator is a framework for creating realistic and valid high-volume test data, used for (unit/integration/load) testing and showcase setup. Metadata constraints are imported from systems and/or configuration files. Data can be imported from and exported to files and systems, anonymized or generated from scratch. Domain packages provide reusable generators for creating domain-specific data as names and addresses internationalizable in language and region. It is strongly customizable with plugins and configuration options.
Requirements: Platform Independent

CLIF is a Load Injection Framework

Description: CLIF is a modular and flexible distributed load testing platform. It may address any target system that is reachable from a Java program (HTTP, DNS, TCP/IP...) CLIF provides 3 user interfaces (Swing or Eclipse GUI, command line) to deploy, control and monitor a set of distributed load injectors and resource consumption probes (CPU, memory...). An Eclipse wizard helps programming support for new protocols. Load scenarios are defined through XML-editing, using a GUI, or using a capture tool. The scenario execution engine allows the execution of up to millions of virtual users per load injector.
Requirements: Java 1.5 or greater, with enhanced support for Linux, Windows XP, MacOSX/PPC

ContiPerf

Description: ContiPerf is a lightweight testing utility that enables the user to easily leverage JUnit 4 test cases as performance tests e.g. for continuous performance testing. It is inspired by JUnit 4's easy test configuration with annotations and by JUnitPerf's idea of wrapping Unit tests for performance testing, but more powerful and easier to use.
Requirements: Windows, Mac OSX, Linux, Solaris and all other platforms that support Java 5

curl-loader

Description: A C-written web application testing and load generating tool. The goal of the project is to provide a powerful open-source alternative to Spirent Avalanche and IXIA IxLoad. The loader uses real HTTP, FTP and TLS/SSL protocol stacks, simulating tens of thousand and hundred users/clients each with own IP-address. The tool supports user authentication, login and a range of statistics.
Requirements: Linux

Database Opensource Test Suite

Description: The Database Opensource Test Suite (DOTS) is a set of test cases designed for the purpose of stress-testing database server systems in order to measure database server performance and reliability.
Requirements: Linux, POSIX

DBMonster

Description: DBMonster is an application to generate random data for testing SQL database driven applications under heavy load.
Requirements: OS Independent

Deluge

Description: An open-source web site stress test tool. Simulates multiple user types and counts. Includes proxy server for recording playback scripts, and log evaluator for generating result statistics. Note: this tool is no longer under active development although it is still available on Sourceforge.
Requirements: OS independent

Dieseltest

Description: Contains the high-end features common to packages costing $50,000 or more. Dieseltest is a Windows application that simulates hundreds or thousands of users hitting a website. To run a load test, you first create a test script using our script editor. The script contains all of the requests that a real-world user would make of a website. You then load the script and run the test. The system will show you real-time results while the script is running, and produce a report analyzing the results at the conclusion.
Requirements: Windows

Faban

Description: Faban is a facility for developing and running benchmarks, developed by Sun. It has two major components, the Faban harness and the Faban driver framework. The Faban harness is a harness to automate running of server benchmarks as well as a container to host benchmarks allowing new benchmarks to be deployed in a rapid manner. Faban provides a web interface to launch & queue runs, and extensive functionality to view, compare and graph run outputs.
Requirements: OS independent; JVM 1.5 or later.

FunkLoad

Description: FunkLoad is a functional and load web tester, written in Python, whose main use cases are functional and regression testing of web projects, performance testing by loading the web application and monitoring your servers, load testing to expose bugs that do not surface in cursory testing, and stress testing to overwhelm the web application resources and test the application recoverability, and writing web agents by scripting any web repetitive task, like checking if a site is alive.
Requirements: OS independent — except for the monitoring which is Linux specific.

FWPTT load testing web applications

Description: fwptt is an open source Web application testing program for load testing web applications. It can record normal and AJAX requests. It has been tested on ASP.Net applications, but it should work with JSP, PHP or other.
Requirements: Windows

Grinder

Description: The Grinder is a Java load-testing framework making it easy to orchestrate the activities of a test script in many processes across many machines, using a graphical console application.
Requirements: OS Independent

GrinderStone

Description: GrinderStone is an Eclipse plug-in for Grinder load testing scripts development including debugging, modularity and pretty logging.
Requirements: All

Hammerhead 2 — Web Testing Tool

Description: Hammerhead 2 is a stress testing tool designed to test out your web server and web site. It can initiate multiple connections from IP aliases and simulated numerous (256+) users at any given time. The rate at which Hammerhead 2 attempts to pound your site is fully configurable, there are numerous other options for trying to create problems with a web site (so you can fix them).
Requirements: Hammerhead has been used with Linux, Solaris and FreeBSD.

Hammerora

Description: Hammerora is a load generation tool for the Oracle Database and Web Applications. Hammerora includes pre-built schema creation and load tests based on the industry standard TPC-C and TPC-H benchmarks to deploy against the Oracle database with multiple users. Hammerora also converts and replays Oracle trace files and enables Web-tier testing to build bespoke load tests for your entire Oracle application environment.
Requirements: Platform independent (binaries for Linux and Windows)

httperf

Description: Httperf is a tool for measuring web server performance. It provides a flexible facility for generating various HTTP workloads and for measuring server performance. The focus is not on implementing one particular benchmark but on providing a robust, high-performance tool that facilitates the construction of both micro and macro level benchmarks. The three distinguishing characteristics of httperf are its robustness, which includes the ability to generate and sustain server overload, support for the HTTP/1.1 and SSL protocols, and its extensibility.
Requirements: Linux (Debian package available), HP-UX, perhaps other Unix

http_load

Description: http_load runs multiple HTTP fetches in parallel, to test the throughput of a Web server. However, unlike most such test clients, it runs in a single process, to avoid bogging the client machine down. It can also be configured to do HTTPS fetches.
Requirements: tbc

Iperf

Description: Iperf was developed by NLANR/DAST as a modern alternative for measuring maximum TCP and UDP bandwidth performance. Iperf allows the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, datagram loss.
Requirements: Platform independent

IxoraRMS

Description: Monitoring tool with great visualization and customization capabilities. It's quick to install and suitable for use in performance labs.
Requirements: Windows, Unix

JChav

Description: JChav is a way to see the change in performance of your web application over time, by running a benchmark test for each build you produce. JChav reads all the JMeter logs from each of your runs (one per build), and produces a set of charts for each test in each run.
Requirements: JMeter

JCrawler

Description: Stress-Testing Tool for web-applications. It comes with the crawling/exploratory feature. You can give JCrawler a set of starting URLs and it will begin crawling from that point onwards, going through any URLs it can find on its way and generating load on the web application. The load parameters (hits/sec) are configurable.
Requirements: OS independent

loadUI

Description: loadUI is a tool for Load Testing numerous protocols, such as Web Services, REST, AMF, JMS, JDBC as well as Web Sites. Tests can be distributed to any number of runners and be modified in real time. LoadUI is tightly integrated with soapUI. LoadUI uses a highly graphic interface making Load Testing Fun and Fast.
Requirements: Any

Lobo, Continuous Tuning

Description: Lobo is a tool for performance testing and monitoring that allows you to monitor the evolution of performance along the time-line of the project. It was specially designed to be used in agile-iterative and evolutionary approaches.
Requirements: Java

MessAdmin

Description: MessAdmin is a light-weight and non-intrusive notification system and HttpSession administration for J2EE Web Applications, giving detailed statistics and informations on the application. It installs as a plug-in to any Java EE WebApp, and requires zero-code modification.
Requirements: OS Independant

mstone

Description: Mstone started as a mail performance measurement system but now can test svn, etc. It can simultaneously test SMTP, POP, IMAP, and some HTML based systems. It measures transaction latency in multiple stages, and graphs the combined results from multiple clients.
Requirements: multiple (Perl based)

Multi-Mechanize

Description: Multi-Mechanize is an open source framework for web performance and load testing. It allows you to run simultaneous python scripts to generate load (synthetic transactions) against a web site or web service.
Requirements: Any

NTime

Description: The NTime tool is very similar to NUnit tool to perform repeatable tasks that help managers, architects, developers and testers to test an application against its

performance.
Requirements: Windows 98 or above, .Net framework 1.1 or 2.0

OpenSTA

Description: A distributed software testing architecture based on CORBA. Using OpenSTA (Open System Testing Architecture) a user can generate realistic heavy loads simulating the activity of hundreds to thousands of virtual users. OpenSTA graphs both virtual user response times and resource utilization information from all Web Servers, Application Servers, Database Servers and Operating Platforms under test, so that precise performance measurements can be gathered during load tests and analysis on these measurements can be performed.
Requirements: Windows 2000, NT4 and XP

OpenWebLoad

Description: OpenWebLoad is a tool for load testing web applications. It aims to be easy to use and providing near real-time performance measurements of the application under test.
Requirements: Linux, Windows

Ostinato

Description: Ostinato is an open-source, cross-platform packet/traffic generator and analyzer with a friendly GUI. It aims to be "Wireshark in Reverse" and thus become complementary to Wireshark.
Requirements: Cross-platform

p-unit

Description: An open source framework for unit test and performance benchmark, which was initiated by Andrew Zhang, under GPL license. p-unit supports to run the same tests with single thread or multi-threads, tracks memory and time consumption, and generates the result in the form of plain text, image or PDF file.
Requirements: OS independent

PandoraFMS

Description: Pandora FMS is a monitoring Open Source software. It watches your systems and applications, and allows you to know the status of any element of those systems. Pandora FMS could detect a network interface down, a defacement in your website, a memory leak in one of your server application, or the movement of any value of the NASDAQ new technology market. If you want, Pandora FMS could send out SMS message when your systems fails… or when Google’s value drop below US$ 500.
Requirements: 32-bit MS Windows (NT/2000/XP), All POSIX (Linux/BSD/UNIX-like OSes), Solaris, HP-UX, IBM AIX

postal

Description: SMTP benchmarking tool. It is threaded, uses very little disk I/O (e-mail body content randomly generate text). It has an SMTP source, SMTP sink and POP server load tester (to pull sent mail).
Requirements: Linux/UNIX; requires C compiler

Pylot

Description: Pylot is a free open source tool for testing performance and scalability of web services. It runs HTTP load tests, which are useful for capacity planning, benchmarking, analysis, and system tuning. Pylot generates concurrent load (HTTP Requests), verifies server responses, and produces reports with metrics. Tests suites are executed and monitored from a GUI.
Requirements: Python 2.5+. required. Tested on Windows XP, Vista, Cygwin, Ubuntu, MacOS

Raw Load Tester

Description: This application calls the URL you select as many times as you choose and tells you how long it took the server to respond. It writes some additional runtime details to the PHP log file so you can optionally do more granular analysis afterwards. Although the server processes most of the statistics, all URL requests come from the browser. You can run as many browsers and workstations simultaneously as you want.
Requirements: PHP/JavaScript

Seagull

Description: Seagull is a multi-protocol traffic generator test tool. Primary aimed at IMS protocols, Seagull is a powerful traffic generator for functional, load, endurance, stress and performance tests for almost any kind of protocol. Currently supports Diameter, XCAP over HTTP, TCAP (GSM Camel, MAP, Win) protocols.
Requirements: Linux/Unix/Win32-Cygwin

Siege

Description: SIEGE is an http regression testing and benchmarking utility. It was designed to let web developers measure the performance of their code under duress, to see how it will stand up to load on the internet. It lets the user hit a webserver with a configurable number of concurrent simulated users. Those users place the webserver "under siege." SCOUT surveys a webserver and prepares the urls.txt file for a siege. In order to perform regression testing, siege loads URLs from a file and runs through them sequentially or randomly. Scout makes the process of populating that file easier. You should send out the scout, before you lay siege.
Requirements: GNU/Linux, AIX, BSD, HP-UX and Solaris.

Sipp

Description: SIPp is a performance testing tool for the SIP protocol. Its main features are basic SIPStone scenarios, TCP/UDP transport, customizable (xml based) scenarios,

dynamic adjustement of call-rate and a comprehensive set of real-time statistics. It can also generate media (RTP) traffic for audio and video calls.
Requirements: Linux/Unix/Win32-Cygwin

SLAMD

Description: SLAMD Distributed Load Generation Engine is a Java-based application designed for stress testing and performance analysis of network-based applications.
Requirements: Any system with Java 1.4 or higher

Soap-Stone

Description: Network benchmark application which can put your network under load and conduct automatic benchmark and recording activities.
Requirements: OS Independent

stress_driver

Description: General-purpose stress test tool.
Requirement: Windows NT/2000, Linux TestMaker *Description:* TestMaker from PushToTest.com delivers a rich environment for building and running intelligent test agents that test Web-enabled applications for scalability, functionality, and performance. It comes with a friendly graphical user environment, an object-oriented scripting language (Jython) to build intelligent test agents, an extensible library of protocol handlers (HTTP, HTTPS, SOAP, XML-RPC, SMTP, POP3, IMAP), a new agent wizard featuring an Agent Recorder to write scripts for you, a library of fully-functional sample test agents, and shell scripts to run test agents from the command line and from unit test utilities.
Requirements: Java 1.4 or higher virtual machine on Windows, Linux, Solaris, and Macintosh.

TPTEST

Description: The purpose with TPTEST is to allow users to measure the speed of their Internet connection in a simple way. TPTEST measures the throughput speed to and from various reference servers on the Internet. The use of TPTEST may help increase the consumer/end user knowledge of how Internet services work.
Requirements: MacOS/Carbon and Win32

Tsung

Description: Tsung is a distributed load testing tool. It is protocol-independent and can currently be used to stress HTTP, SOAP and Jabber servers (SSL is supported). It simulates complex user's behaviour using an XML description file, reports many measurements in real time (including response times, CPU and memory usage from servers, customized transactions, etc.). HTML reports (with graphics) can be generated during the load. For HTTP, it supports 1.0 and 1.1, has a proxy mode to record sessions, supports GET and POST methods, Cookies, and Basic WWW-authentication. It has already been used to simulate thousands of virtual users.
Requirements: Tested on Linux, but should work on MacOSX and Windows.

Valgrind

Description: Valgrind is an award-winning suite of tools for debugging and profiling Linux programs. With the tools that come with Valgrind, you can automatically detect many memory management and threading bugs, avoiding hours of frustrating bug-hunting, making your programs more stable. You can also perform detailed profiling, to speed up and reduce memory use of your programs.
Requirements: Linux

Web Application Load Simulator

Description: LoadSim is a web application load simulator. It allows you to create simulations and have those simulations run against your webserver.
Requirements: JDK 1.3 or above

Web Polygraph

Description: Benchmarking tool for caching proxies, origin server accelerators, L4/7 switches, content filters, and other Web intermediaries.
Requirements: C++ compiler

WebLOAD

Description: WebLOAD Open Source is a fully functional, commercial-grade performance testing product based on WebLOAD, Radview's flagship product that is already deployed at 1,600 sites. Available for free download and use, WebLOAD is a commercial-grade open source project with more than 250 engineering years of product development. Companies that require commercial support, additional productivity features and compatibility with third-party protocols have the option of purchasing WebLOAD Professional directly from RadView.
Requirements: Windows NT/2000/XP

References

[1] L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobreva, V. Katifori, and H. Schuldt. The Digital Library Reference Model, 2007. Version 0.98, December 2007, project report, DELOS, A Network of Excellence on Digital Libraries, Thematic Priority: IST-2002-2.3.1.12, Technology-enhanced Learning and Access to Cultural Heritage, http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf .

[2] L. Candela, A. Nardi, G. Athanasopoulos, D. Castelli, K. El Raheb, P. Innocenti, Y. Ioannidis, A. Katifori, A. Nika, G. Vullo, and S. Ross. The Digital Library Reference Model, 2011. Deliverable D3.2b of Project DL.org (Coordination Action on Digital Library Interoperability, Best Practices & Modelling Foundations), ICT Programme "Cultural Heritage and Technology Enhanced Learning", Project Number: 231551 http://bscw.research-infrastructures.eu/pub/bscw.cgi/d222816/D3.2b%20Digital%20Library%20Reference%20Model.pdf .

[3] DELOS Network of Excellence. DELOS Network of Excellence — Homepage, 2011. http://www.delos.info/.

[4] DL.org Consortium. DL.org: Coordination Action on Digital Library Interoperability, Best Practices and Modelling Foundations — Homepage, 2011. http://www.dlorg.eu/.

[5] N. Fuhr, P. Hansen, M. Mabe, A. Micsik, and I. Solvberg. Digital Libraries: A Generic Classification and Evaluation Scheme. In *ECDL'01: Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries, Lect. Notes Comput. Sci 2163*, pages 187–199. 2001.

[6] Norbert Fuhr, Giannis Tsakonas, Trond Aalberg, Maristella Agosti, Preben Hansen, Sarantos Kapidakis, Claus-Peter Klas, Lazlo Kovas, Monica Landoni, Andras Micsik, Christos Papatheodorou, Carol Peters, and Ingeborg Solvberg. Evaluation of Digital Libraries. *International Journal on Digital Libraries (IJDL)*, 8(1):27–38, 2007.