

DELOS Software Inventory

Editors:
Vittore Casarosa (ISTI-CNR)
Alessandro Launaro (ISTI-CNR)
First Edition - February 2007

DELOS: a Network of Excellence on Digital Libraries
<http://www.delos.info/>



Table of contents

3D CBR - Three dimensional Content Based Retrieval	3
DELOS DLMS – Digital Library Management System	5
IEMSR - Metadata Schema Registry	8
ISIS – Interactive Similarity Search.....	10
ITR - Item Recommender	12
MESSIF - Metric Similarity Search Implementation Framework	14
MILOS - Multimedia Content Management System.....	16
MINERVA - Peer-to-peer Web Search Engine BINGO! - Focused crawler	18
MM4U – A Component Framework for Personalized Multimedia Applications	21
O’GRAPE - OSIRIS Graphical Process Editor	23
OSIRIS - Open Services Infrastructure for Reliable und Integrated process Support.....	25
OSIRIS-SE – OSIRIS Stream-Enabled.....	27
P2P DL - Peer to peer Digital Library	30
RPextract - Music Feature Extractor.....	32
Sightseeing4U - Development of personalized mobile multimedia applications	34
SOMToolbox - Implementation of Self-organizing map models	35
Transformation4U - Transformation of Multimedia documents for mobile applications	37
xSMART - Semi-automatic context-aware authoring tool	40

3D CBR - Three dimensional Content Based Retrieval

Contact point:

Alberto Del Bimbo (delbimbo@dsi.unifi.it)
University of Florence, Italy

Technical Contact Points:

Stefano Berretti (berretti@dsi.unifi.it)
Pietro Pala (pala@dsi.unifi.it)

Type of Software:

PROTOTYPE

Descriptive Keywords:

3D content based retrieval

Potential Use and Applications:

3D models are being created and employed in a wide range of application domains, including medicine, computer aided design and engineering, and cultural heritage. The development of techniques to enable retrieval by content of 3D models assumes an ever increasing relevance in all those area where it is necessary to find 3D objects according to their shape similarity.

General Description:

At the Media Integration and Communication Center of the University of Florence, a prototype system for content-based retrieval of 3D objects has been developed. The basic idea of 3D content-based retrieval, is to allow users to find objects in a large repository by providing to the system an object as example of the main characteristics the user is looking for. The query object is described according to a set of representative features, which capture salient information of the object surface. The goal for the retrieval system is to find 3D objects similar to the query model in terms of the extracted features. In the proposed prototype, the spatial distribution of curvature information extracted from the object surface is used to represented 3D objects by *curvature correlograms*. This 3D description and retrieval module is under integration into the DELOS DLMS.

Technical Description:

The 3D retrieval system, supports feature extraction and content-based retrieval of three-dimensional (3D) objects. Functionalities that are currently supported by the system are:

- *3D feature extraction* – Given a 3D model as a mesh of triangles, this component extracts a feature vector representation of the 3D object. An object is described by a set of feature vectors capturing its main shape characteristics.
- *Description storage* – Given a 3D object descriptor, this module stores it in a local archive of 3D object descriptors. In the current implementation, feature vectors are stored in a relational database (MySQL).
- *3D object retrieval* – Given an input 3D object, this module matches the query model descriptor against the set of reference descriptors stored in the local archive. The match is performed by determining the best correspondence between features vectors representing the query and each of the database models. The module is devised to perform matching between 3D object representations in the format generated by the “3D Feature Extraction” module. The module returns a list of (name of) 3D models and their distances, ranked in order of increasing distance from the query.
- *Query interface* – A user interface is available to query the retrieval system. The interface is developed as a web application using the JSP/servlet technology.

Required User Skills:

No particular skills are required in using the prototype. Just a normal level of practice in interacting with software applications is needed.

Pre-Requisites for Installation:

Two 3D retrieval applications are available. The first one is based on a web interface that allows remote querying of an archive of 3D objects. This application is completely written in Java so, in principle, it is platform independent. The requirement to run the server side part of the system is the installation of the Java platform. Running the client side only requires a web browser. The second prototype is a standalone application written in C++ developed for the Windows operating system.

Conditions of Use:

Contact the authors.

DELOS DLMS – Digital Library Management System

Contact point:

Heiko Schuldt (heiko.schuldt@unibas.ch)
University of Basel, Switzerland
Hans-Jörg Schek (schek@inf.ethz.ch)
University of Konstanz, Germany

Technical Contact Points:

Gert Brettlecker (gert.brettlecker@unibas.ch)
Paola Ranaldi (paola.ranaldi@unibas.ch)

Type of Software:

STABLE PROTOTYPE

Descriptive Keywords:

Windows, C++, Process Management, Scalability, Reliability, Replication, Service Oriented Architecture, SOAP, Web Services, Java, Axis, Tomcat

Potential Use and Applications:

DelosDLMS is a prototype of a next-generation Digital Library (DL) management system. Currently, it combines text and audio-visual searching, offers new information visualization and relevance feedback tools, provides novel interfaces, allows retrieved information to be annotated and processed, integrates and processes sensor data streams, and finally, from a systems engineering point of view, is easily configured and adapted while being reliable and scalable.

General Description:

The overall goal of the DelosDLMS is the implementation of a prototype of a next-generation digital library management system by integrating the powerful and highly sophisticated prototype systems and services which have originally been developed independently in the context of the DELOS network of excellence. The prototype is based on the OSIRIS/ISIS platform, a middleware environment developed by ETH Zürich and now being extended at the University of Basel.

OSIRIS is a general-purpose middleware for reliable and fault tolerant distributed processes. ISIS is a prototype application for information retrieval in multimedia collections. Both of them are fully described as separate entries in this Inventory. An online demonstration is available at <http://isisdemo.cs.unibas.ch> (username and password are provided upon request).

The DELOS DLMS is continuously extended by integrating further DL services. The services available as of February 2007 are the following, which are fully described as separate entries in this Inventory.

- Daffodil (University of Duisburg-Essen)
- MedioVis (University of Konstanz)
- Fast Annotation Service (University of Padua)
- SOM Service (University of Konstanz)
- Audio Features Vienna (TU Vienna)
- 3D Features Florence (University of Florence)
- Video Services (University of Florence)
- OntoNL Services (TU Crete)
- Interactive Paper (ETH Zürich)

Technical Description:

OSIRIS/ISIS is implemented completely in C++ based on the Windows API. At a basic level it offers a generic message oriented middleware, which allows exchange of messages and efficient execution of processes by implementing a distributed workflow execution engine. ISIS is a process-based digital library application, which is implemented as a set of services that act on top of OSIRIS. The SOAP gateway allows integrating external loosely coupled services. Using O'GRAPE (OSIRIS GRaphical Process Editor), new workflows combining existing services can easily be created.

Detailed information about ISIS and OSIRIS is presented in "The OSIRIS Process Support Middleware and the ISIS Process-Based Digital Library Application" (http://dbis.cs.unibas.ch/delos_website/jpa2/osiris.pdf).

Required User Skills:

Fair knowledge of C++ and Windows programming, conceptual knowledge of message oriented middleware, SOA, and process management is recommended in order to be able to fully understand the source code and implement new tightly coupled (OSIRIS) services. For the integration of WSDL/SOAP based services (loosely coupled) into custom processes no knowledge of C++ and the OSIRIS source code is needed. For SOAP service implementation with Java and Axis, fair knowledge of Java is needed.

Pre-Requisites for Installation:

At least one PC with Microsoft Windows 2000 or newer and an available SQL Server database server preferable at the same machine is needed to run the OSIRIS platform. Additional, memory and disc requirements are mostly demanded by the services one wants to install. The OSIRIS core services and the middleware layer itself are not resource demanding. Additionally a current Java runtime is needed for O'Grape and services hosted by Tomcat/Axis.

Conditions of Use:

Contact the authors.

IEMSR - Metadata Schema Registry

Contact point: Emma Tonkin (e.tonkin@ukoln.ac.uk) UKOLN, University of Bath, UK
Technical Contact Points: Emma Tonkin (e.tonkin@ukoln.ac.uk)
Type of Software: PROTOTYPE
Descriptive Keywords: Metadata schema registry. Application profile and schema development
Potential Use and Applications: Metadata schema registries enable the publication, navigation and sharing of information about metadata. The IEMSR will act as the primary source for authoritative information about metadata schemas recommended by the JISC IE Standards framework, providing the JISC IE with a single point of referral for recommended schemas.
General Description: <p>The IEMSR (Information Environment Metadata Schema Registry) project aims to progress development of a metadata schema registry as a shared service within the JISC Information Environment. Metadata schema registries enable the publication, navigation and sharing of information about metadata. The IEMSR project is building on the work of previous projects, which have explored provision of information about metadata at the level of data elements, element sets or application profiles. The Registry will act as the primary source for authoritative information about metadata schemas recommended by the JISC IE Standards framework, providing the JISC IE with a single point of referral for recommended schemas.</p> <p>The Registry project is scoped to target the UK education community where both Dublin Core (DC) and IEEE Learning Object Metadata (LOM) standards are used to build schemas. It will allow various initiatives within the JISC IE to publish "application profiles" of these standards in a registry, making them available to others. This provides a concrete way of encouraging sensible uniformity alongside necessary divergence.</p>
Technical Description: <p>The IEMSR itself is a web-based service that offers a SPARQL interface for access to the data, stored internally in RDF format. IEMSR development includes both this backend service, a web-based system to permit searching and browsing of data already placed in the registry, and a cross-platform Java-based client designed to support and speed up the process of application profile and metadata profile design. As the API is built around W3C standards, the data within the registry can be extracted, transformed and reused using standard technologies such as XML style sheets.</p> <p>The data within the system is designed to support both machine-to-machine (m2m) access without compromising reusability and the ability to search and browse the data.</p>

Required User Skills:

Potential users of the client software and Web-based system are welcome, although the system remains in development at the present time, meaning that users should liaise with project management in the first instance. Reuse of data via the machine-to-machine backend service will require an understanding of SPARQL and XML transforms as of the present time.

Pre-Requisites for Installation:

The IEMSR backend software requires a Unix or compatible system with a recent Perl installation. The web-based system requires Java and Apache Tomcat for installation, and can be browsed using a standard web browser such as Firefox. The client requires a Java runtime environment. No special requirements exist in terms of hardware; the client is cross-platform.

Conditions of Use:

Contact the authors.

ISIS – Interactive Similarity Search

Contact Point: Heiko Schuldt (heiko.schuldt@unibas.ch) University of Basel, Switzerland
Technical Contact Points: Paola Ranaldi (paola.ranaldi@unibas.ch) Michael Springmann (michael.springmann@unibas.ch) Gert Brettlecker (gert.brettlecker@unibas.ch)
Type of software: STABLE PROTOTYPE
Descriptive Keywords: Content-Based Multimedia Retrieval, Image Retrieval, Audio Retrieval, Video Retrieval, P2P, High-dimensional index, VA-File, Relevance Feedback
Potential Use and Applications: ISIS consists of a set of services, which can be used for content-based retrieval. The entire system or individual services can be used to build digital library applications. E.g. it provides the user interface and many of the core services for the DelosDLMS.
General Description: ISIS stands for Interactive Similarity Search and is a prototype application for information retrieval in multimedia collections originally built at ETH Zürich, Switzerland. The development has continued at UMIT, Hall i.T., Austria and now at the University of Basel, Switzerland. It supports content-based retrieval of images, audio and video content, and the combination of any of these media types with sophisticated text retrieval. An online demonstration is available at http://isisdemo.cs.unibas.ch (username / password upon request). One of the main considerations in designing ISIS was to ensure high scalability and flexibility. Therefore, instead of implementing one monolithic application, ISIS consists of a set of specialized application services for similarity search that are combined by the OSIRIS middleware. The ISIS services can be easily distributed among several nodes in a network. Detailed information about ISIS and OSIRIS is presented in “The OSIRIS Process Support Middleware and the ISIS Process-Based Digital Library Application” (http://dbis.cs.unibas.ch/delos_website/jpa2/osiris.pdf).
Technical Description: ISIS is a process-based digital library application, which is implemented as a set of services that act on top of OSIRIS, a distributed workflow execution engine (see OSIRIS in the Software Inventory for more details). Its main parts have been implemented in C++ and will run on Microsoft Windows Systems providing the OSIRIS middleware. A web service interface exists to ease communication with external services. Using O’GRAPE (OSIRIS GRAPHICAL Process Editor), new workflows to combine services can easily be created. Documentation for each operation provided by ISIS exists; each can even be tested via online forms.

Required User Skills:

For using the online demonstration, no special skills are required. A web interface optimized for Microsoft Internet Explorer is provided at above mentioned location. For using some services in other digital library applications, understanding the concept of SOA (Service-Oriented Architectures) is essential. For deploying ISIS in a new environment, the ability to setup and manage OSIRIS is required.

Pre-Requisites for Installation:

At least one PC with Microsoft Windows 2000 or newer, the OSIRIS platform, enough free disc space to hold all desired content of the digital library, indexes, and log files (rule of thumb: 1.5 times the size of the content should be sufficient).

Conditions of Use:

Please contact the authors.

ITR - Item Recommender

Contact Point: Giovanni Semeraro (semeraro@di.uniba.it) University of Bari, Italy
Technical Contact Points: Pierpaolo Basile (basile@di.uniba.it) Marco Degemmis (degemmis@di.uniba.it)
Type of software: PROTOTYPE
Descriptive Keywords: Content-based Recommender System, User Profiling System
Potential Use and Applications: The main target of personalization in a virtual information and knowledge environment is the reduction of information overload. User modeling and tracking activities provide the basis for a wide range of services that reuse the semantic information describing properties of the user. Such personalization services contribute to a more targeted information access. Typical applications of user profiles in an information environment are information filtering, personalized information recommendations, and targeted notification about changes in the information space. ITR system could be used for developing intelligent recommendation services in which items to be recommended are described by using text.
General Description: ITem Recommender (ITR) is a <i>content-based</i> profiling system able to infer user profiles from textual documents rated by users according to their interests. User profiles are exploited in a recommendation process to filter and to suggest new documents to the users. The system implements a naïve Bayes text categorization algorithm and it is able to classify documents as interesting or uninteresting for a specific user by exploiting a probabilistic model learned from training examples, the documents rated by that user in the past. ITR represents documents by using <i>senses</i> corresponding to concepts identified from words in the original text through an automated Word Sense Disambiguation procedure. The final outcome of the learning process is a probabilistic model of the user interests. The model is used as a personal profile including those concepts that turn out to be most indicative of the user's preferences, according to the value of the parameters of the model.
Technical Description: The system is implemented in Java. Database: MySQL ver. 4.1 or later (© 1995-2007 MySQL AB. All rights reserved) Web Server: Apache Tomcat ver. 5.0 or later (Copyright © 1999-2006, The Apache Software Foundation) Other resources: WordNet 2.0 (WordNet 2.0 Copyright 2003 by Princeton University. All rights reserved)

Required User Skills:

Java programming.

Pre-Requisites for Installation:

Java Virtual Machine 1.5.0 or later

Apache Tomcat 5.0 or later

MySQL 4.1 or later

WordNet 2.0

ITR has been compiled and tested on JDK 1.5.0 from SUN, and should work on any compliant Virtual Machine. Currently running under Linux/Windows.

Conditions of Use:

Contact the authors.

MESSIF - Metric Similarity Search Implementation Framework

Contact Point:

Michal Batko (xbatko@fi.muni.cz)
Masaryk University Brno, Czech Republic

Technical Contact Points:

Michal Batko (xbatko@fi.muni.cz)
David Novak (xnovak8@fi.muni.cz)

Type of Software:

STABLE PROTOTYPE

Descriptive Keywords:

Metric Similarity Search Implementation Framework (MESSIF), similarity search, metric space, data structures, index structures, distributed data structures, structured peer-to-peer networks, implementation platform.

Potential Use and Applications:

Number of researchers has recently focused on indexing and searching using the metric space model of data. The implementation framework called MESSIF eases the task of building metric-based similarity-searching prototypes. It provides a number of modules from storage management to automatic collecting of performance statistics.

General description:

The similarity search has become a fundamental computational task in many applications. One of the mathematical models of the similarity – the metric space – has drawn attention of many researchers resulting in several sophisticated metric-indexing techniques. An important part of a research in this area is typically a prototype implementation and subsequent experimental evaluation of the proposed data structure. Individual structures are often based on very similar underlying principles or even exploit some existing structures on lower levels. Therefore, the implementation calls for a uniform development platform that would support a straightforward reusability of code. Such a framework would also simplify the experimental evaluation; make the comparison fairer and thus the results would be of greater value. The Metric Similarity Search Implementation Framework (MESSIF) pursue the following objectives:

- to provide basic support for the indexing based on metric space – let developers focus on the higher-level design
- to create a unified and semi-automated mechanism for measuring and collecting statistics
- to define and use uniform interfaces to support modularity and thus allow reusing of the code
- to provide infrastructure for distributed processing with focus on peer-to-peer paradigm – communication support, deployment, monitoring, testing, etc.
- to support complex similarity search in multi-metric spaces

For more details see the following paper:

Batko M., Novak D. and Zezula P. forthcoming. MESSIF: Metric Similarity Search Implementation Framework. To be published at the DELOS Conference (12-16 February 2007).

Technical description:

The MESSIF is implemented as a collection of cooperating Java 6 packages. The MESSIF networking operates over the standard internet or intranet using the family of IP protocols. Individual peers are identified by the IP address plus a port number. The entire communication is based on messages using the TCP and UDP protocols.

For more details see the following paper:

Batko M., Novak D. and Zezula P. forthcoming. MESSIF: Metric Similarity Search Implementation Framework. To be published at the DELOS Conference (12-16 February 2007).

Required User Skills:

Java programming (J2SE v5.0 and higher) skills needed to build new indexing structures. No additional skills are required to use already created structures with included clients (after reading documentation).

Pre-Requisites for Installation:

Hardware: Any computer that can run applications in Java Runtime Environment; distributed index structures need several computers (peers) connected with TCP/IP network and at least one UDP and one TCP port available (not blocked by firewall).

Software: Java Runtime Environment version 5.0 or higher; MESSIF; Any index structure built for MESSIF

Conditions of Use:

Contact the authors.

MILOS - Multimedia Content Management System

Contact Point: Giuseppe Amato (giuseppe.amato@isti.cnr.it) ISTI – CNR, Italy
Technical Contact Point: Giuseppe Amato (giuseppe.amato@isti.cnr.it) Paolo Bolettieri (paolo.bolettieri@isti.cnr.it)
Type of software: STABLE PROTOTYPE
Descriptive Keywords: Multimedia Content Management, Native XML database
Potential Use and Applications: Multimedia document intensive applications
General description: MILOS is a multimedia content management system with special functionalities for multimedia document intensive applications. It offers functionalities for multimedia document management, arbitrary metadata management and retrieval, metadata mapping. Documents and metadata can be retrieved by combining standard search paradigms with similarity search paradigms. Applications can be easily built on top of MILSO by using a well-defined API based on Web Services technology. More info at: http://milos.isti.cnr.it and http://milos.isti.cnr.it/milos/album/
Technical Description: The MILOS MCMS has been developed by using the Web Services technology. MILOS is composed of three main components: <ol style="list-style-type: none">1) the XML Search Engine (XMLSE) component;2) the Multi Media Server (MMS) component;3) the MultiMedia Digital Library service (MMDLS) component. All these components are implemented as Web Services and interact by using SOAP. The XMLSE manages the metadata of the DL. It relies on our technology for native XML databases, and offers the functionality illustrated at point 2) above. The MMS manages the multimedia documents used by the DL applications. MMS offers the functionality of point 1) above. The MMDLS implements the service logic of the repository providing developers of DL applications with a uniform and integrated way of accessing MMS and XMLSE. In addition, it supports the mapping of different metadata schemas as described at point 3) above. All these components were built choosing solutions able to guarantee the requirements of flexibility, scalability, and efficiency, as discussed in the next sections. More info on: http://milos.isti.cnr.it and http://milos.isti.cnr.it/milos/album/

Required User Skills:

System Administration, Web Service programming, Java Programming

Pre-Requisites for Installation:

Intel based computers with Windows 2000 or XP

Conditions of Use:

BSD License + separate licenses for third party components.

MINERVA - Peer-to-peer Web Search Engine

BINGO! - Focused crawler

Contact Point:

Tom Crecelius (tcrecel@mpi-sb.mpg.de)
Matthias Bender (mbender@mpi-sb.mpg.de)
Sebastian Michel (smichel@mpi-sb.mpg.de)
Max-Planck Institute for Informatics, Saarbruecken, Germany

Technical Contact Point:

Tom Crecelius (tcrecel@mpi-sb.mpg.de)
Matthias Bender (mbender@mpi-sb.mpg.de)
Sebastian Michel (smichel@mpi-sb.mpg.de)

Type of software:

STABLE PROTOTYPE

Descriptive Keywords:

Minerva, Bingo, P2P, Web Search Engine, Focused Crawler

Potential Use and Applications:

A user can automatically crawl the web according to her thematic interests in order to create a customized local collection. This collection can be shared with other peers in the network, so that users can execute queries in a distributed way involving a tunable number of peers.

General description:

MINERVA is a P2P Web search engine prototype that envisions a collaboration of autonomous peers. Peers are expected to run a local search functionality on a local index, e.g., created by the integrated focused crawler Bingo!. MINERVA creates synopses that compactly describe the local index and publishes these synopses to a directory, which is physically distributed using a distributed hash table (DHT).

When a user is not satisfied with search results from her local engine, she can turn to MINERVA, which identifies promising information sources for the query based on these synopses and automatically forwards the query accordingly. Selected remote peers execute the query on their local indexes and return their results to the query initiator, where MINERVA merges the results into a single result list that is displayed to the user.

More details on MINERVA can be found at:

<http://www.mpi-sb.mpg.de/departments/d5/software/minerva/index.html>

and in the following publication:

Matthias Bender, Sebastian Michel, Josiane Xavier Parreira, Tom Crecelius:

P2P Web Search: Make It Light, Make It Fly. Third Biennial Conference on Innovative Data System Research (CIDR 07), Asilomar, USA, January 2007, pp. 164-168.

Electronic proceedings available at <http://www.cidrdb.org/2007Proceedings.zip>

Technical Description:

Bingo! and MINERVA are implemented in Java 5 and compile platform-independently. By default, both use Apache Derby (Cloudscape) as a common database system to store the local index and the precomputed synopses, but also support external JDBC-compliant database systems. The MINERVA synopses contain per-key statistical information that describes the local index and support different techniques for query routing and result merging. Developers can extend MINERVA by implementing their own routing and merging strategies. MINERVA uses Past/Pastry to implement the distributed directory, using its network routing and storage functionality to become resilient to network dynamics and node failures.

Required User Skills:

Only low technical skills are required.

Pre-Requisites for Installation:

Hardware:

PC with 512 Mbyte main memory (1 Gbyte recommended)

Software:

Operating system: Linux / Windows

Conditions of Use:

Copyright by Max-Planck Institute for Informatics, Department for Databases and Information Systems, headed by Prof. Dr. Gerhard Weikum, Stuhlsatzenhausweg 85, D-66123 Saarbruecken, Germany

For further information please contact Prof. Dr. Gerhard Weikum: weikum@mpi-inf.mpg.de

THESE INFORMATIONS ARE PROVIDED WITHOUT ANY EXPRESSED OR IMPLIED WARRANTIES. IN NO EVENT SHALL THE DATABASE AND INFORMATION SYSTEMS RESEARCH GROUP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THESE RECOMMENDATIONS, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The name "BINGO!" must not be used to endorse or promote products derived from this software without prior permission from Prof. Weikum. This does not apply to add-on libraries or tools that work in conjunction with BINGO! framework.
- Prof. Weikum may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published. No one other than Prof. Weikum has the right to modify the terms applicable to covered code created under this License.

- Redistributions of any form whatsoever must retain the following acknowledgment:
"This software includes BINGO!, freely available from:
<http://www.mpi-sb.mpg.de/units/ag5/software/>".

Additional information:

The MINERVA software can be downloaded from the URL
<http://www.mpi-sb.mpg.de/departments/d5/software/minerva/index.html> .

To cite MINERVA, please use the following reference:

Matthias Bender, Sebastian Michel, Josiane Xavier Parreira, Tom Crecelius:

P2P Web Search: Make It Light, Make It Fly. Third Biennial Conference on Innovative Data System Research (CIDR 07), Asilomar, USA, January 2007, pp. 164-168.

Electronic proceedings available at <http://www.cidrdb.org/2007Proceedings.zip>

The development of MINERVA has been supported by the EU Integrated Project DELIS and the EU Network of Excellence DELOS.

MM4U – A Component Framework for Personalized Multimedia Applications

Contact Point: Jochen Meyer (jochen.meyer@offis.de) OFFIS, Germany
Technical Contact Point: Ansgar Scherp (ansgar.scherp@offis.de)
Type of Software: STABLE PROTOTYPE
Descriptive Keywords: Multimedia Authoring, Multimedia Content, Dynamic Generation of Personalized Multimedia Content and Adapted Multimedia Content, Hypermedia, Software Framework, Component Framework, Component Technology, Context, User Modeling; Personalization, Adaptation
Potential Use and Applications: The MM4U component framework can be used for the development of personalized multimedia applications. Personalized multimedia applications means that these applications need to provide multimedia content according to the different needs, requirements, preferences etc. of their users.
General description: The aim of the MM4U component framework is to provide application developers with an extensive and domain independent support for the authoring of personalized multimedia content. The framework provides generic functionality for typical tasks of the general process chain for creating personalized multimedia content. This includes selecting media elements such as audio, video, image, and text according to the user's individual profile and context information. These media elements are then assembled and composed in time and space using an abstract multimedia content representation model. This model captures the different aspects of the multimedia presentation such as the temporal course, the spatial layout, and the interaction possibilities without instantiating these in a concrete syntax and format. Only in the subsequent transformation phase -- called the <i>last mile</i> -- the multimedia content is transformed into a concrete multimedia presentation format, like SMIL, SVG, and Flash, and delivered to the targeted (mobile) end device.
Required User Skills: Programming in Java.
Pre-Requisites for Installation: Java Runtime Environment and a couple of freely available software libraries.
Conditions of Use: Contact the authors.

Additional Information:

- Scherp A. and Boll S. 2005. MM4U: a framework for creating personalized multimedia content. In U. Srinivasan and S. Nepal (eds.), *Managing Multimedia Semantics*. IRM Press. 246-287.
- Scherp A. and Boll S. 2005. Context-driven smart authoring of multimedia content with xSMART. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*. ACM Press: New York. 802-803.
- Scherp A. and Boll S. 2005. A lightweight process model and development methodology for component frameworks. In W. Weck, J. Bosch, R. Reussner and C. Szyperski (eds.), *10th International Workshop on Component-Oriented Programming (ECOOP 2005)*. Glasgow (UK), July 2005.
- Scherp A. and Boll S. 2005. Paving the last mile for multi-channel multimedia presentation generation. In: Y.-P. Phoebe Chen (ed.), *Proceedings of the 11th International Conference on Multimedia Modeling (MMM 2005)*. Melbourne (Australia). Washington: IEEE Computer Society. 190-197.
- Scherp A. and Boll S. 2004. Generic support for personalized mobile multimedia tourist applications. In *Proceedings of the 12th ACM Int. Conf. on Multimedia*, S. 178-179, New York, NY, USA, ACM, October, 2004. ISBN 1-58113-893-8
- Boll S., Krösche J. and Scherp A. 2004. Personalized mobile multimedia meets location-based services. In P. Dadam and M. Reichert (eds.), *Multimedia-Informationssysteme Workshop of the 34th Annual Meeting of the Germany Society for Computing Science. Ulm (Germany), September 2004*. Lecture Notes in Informatics 51. Bonn: GI. Vol. 2: 64-69.
- Scherp A. and Boll S. 2004. MobileMM4 – framework support for dynamic personalized multimedia content on mobile systems. In C. Branki, J. F. Hampe, R. Helfrich, K. Kurbel, G. Schwabe, F. Teuteberg, S. Uellner, R. Unland und G. Wanner (eds.), *Multikonferenz Wirtschaftsinformatik. Duisburg (Germany), March 2004*. Vol. 3: 204-215.

O'GRAPE - OSIRIS Graphical Process Editor

Contact Point: Heiko Schuldt (heiko.schuldt@unibas.ch) University of Basel, Switzerland
Technical Contact Points: Paola Ranaldi (paola.ranaldi@unibas.ch) Michael Springmann (michael.springmann@unibas.ch) Gert Brettlecker (gert.brettlecker@unibas.ch)
Type of software: STABLE PROTOTYPE
Descriptive Keywords: Java, XML, Process Management, Process Modeling
Potential Use and Applications: O'Grape is the graphical process editor for OSIRIS, ISIS, OSIRIS-SE, and the Delos DLMS.
General Description: <p>O'Grape allows defining processes in a boxes and arrows approach. The arrows in the process describe the control flow and boxes represent services invocations. The data flow of a process is modeled by using the concept of whiteboard, which contains the global variables of a process instance.</p> <p>During process execution, the whiteboard of a process instance is first filled with the process arguments. Whiteboard parameters might be modified during process execution by service invocations. Finally, the return parameters of a service invocation are fed back to the whiteboard and the result of the process invocation is assembled from its contents. Moreover, O'Grape allows to model transactional processes, which may have additional failure-, compensation-, and rollback-edges.</p>
Technical Description: O'Grape is implemented in Java and runs as a standalone application as well as an applet. O'Grape is able to directly contact the process description repository of OSIRIS/OSIRIS-SE, to download and upload process definitions. Internally O'Grape uses the Java Swing framework for the graphical user interface. Process definitions are represented in XML format.
Required User Skills: In order to use O'Grape no special skills are required. In order to further develop O'Grape Java and XML knowledge is recommended.
Pre-Requisites for Installation: O'Grape is based on Java 1.3 and is not resource demanding. In order to have O'Grape and the OSIRIS web-interface working properly together the use of Microsoft Internet Explorer as browser is recommended.

Conditions of Use:

Contact the authors.

OSIRIS - Open Services Infrastructure for Reliable und Integrated process Support

Contact Point:

Heiko Schuldt (heiko.schuldt@unibas.ch)
University of Basel, Switzerland

Technical Contact Points:

Paola Ranaldi (paola.ranaldi@unibas.ch)
Michael Springmann (michael.springmann@unibas.ch)
Gert Brettlecker (gert.brettlecker@unibas.ch)

Type of Software:

STABLE PROTOTYPE

Descriptive Keywords:

Windows, C++, Process Management, Scalability, Reliability, Replication, Service Oriented Architecture

Potential Use and Applications:

Standards like SOAP, WSDL, and UDDI facilitate the proliferation of services. Based on these technologies, processes are a means to combine services to applications and to provide new value-added services ("programming in the large"). For large information systems, a centralized process engine is no longer appropriate due to limited scalability. Instead, OSIRIS offers a distributed and decentralized process engine that routes process instances directly from one node to the next ones. Such a Peer-to-Peer Process Execution (P3E) promises good scalability characteristics since it is able to dynamically balance the load of processes and services among all available service providers. OSIRIS is the process execution engine of the DelosDLMS.

General description:

OSIRIS, short for Open Service Infrastructure for Reliable and Integrated process Support has been developed at the ETH Zurich, and continued at UMIT in Hall in Tyrol and now at the University of Basel. OSIRIS combines the benefits of several existing infrastructures, namely: i.) discovery and invocation of web services from frameworks, ii.) process support and execution guarantees from workflow management systems, iii.) late binding of service calls and load balancing from grid infrastructures, and, finally, iv.) Peer-to-Peer Processes Execution (P3E) in analogy to peer-to-peer file systems. An interesting characteristic of P3E in OSIRIS is that process navigation costs only accumulate on nodes of the community that are directly involved in the execution of a process. Especially, there is no central component in charge with process execution. Therefore, and in contrast to a centralized process engine, P3E bears the potential to scale well with the number of concurrent processes and the number of service providers. However, this requires sophisticated strategies for the replication of meta information for P3E. Especially, replication mechanisms should avoid frequent accesses to global information repositories. OSIRIS deploys a clever publish/subscribe based replication scheme together with freshness predicates to significantly reduce replication costs.

An online demonstration is available at <http://isisdemo.cs.unibas.ch> (username and password will be provided upon request).

Technical Description:

OSIRIS is implemented completely in C++ based on the Windows API. At a basic level it offers a generic message-oriented middleware, which allows exchange of messages in an efficient proprietary way based on TCP socket connections. Based on the messaging system, OSIRIS offers a component (service) framework that allows for configuration, lifecycle management, and service execution. Additionally, a replication framework allows for sophisticated replication of meta-data between services (components). Message processing at component level is done in a service-based concept, where service interaction is in request-response style.

Moreover, OSIRIS offers a dedicated agent component, which is responsible for communication between OSIRIS providers and offers pipeline based message processing. The agent has an incoming and outgoing message processing pipeline, which, allows also the arbitrary plugging of dedicated pipeline handlers. An example of a pipeline handler is the process execution engine running on each provider. Additional pipeline handlers are e.g., responsible for message routing and load balancing. This pipeline based message processing allows an easy extension of OSIRIS.

A dedicated gateway component offers the ability to integrate non-OSIRIS services, like standard WSDL/SOAP services in a loosely coupled way. The OSIRIS infrastructure offers a generic web-based user interface, which allows taking control of participating providers and their services (components). Newly integrated services are easily to adapt to this generic user interface by applying a web templates by using XSLT. Moreover, process execution is monitored and the web interface allows investigating the process logs. Finally, O'GRAPE (OSIRIS GRaphical Process Editor) is accessible via the web interface in order to create and modify workflows to combine services into processes.

Detailed information about ISIS and OSIRIS is presented in "The OSIRIS Process Support Middleware and the ISIS Process-Based Digital Library Application" (http://dbis.cs.unibas.ch/delos_website/jpa2/osiris.pdf).

Required User Skills:

Fair knowledge of C++ and Windows programming, conceptual knowledge of message-oriented middleware, SOA, and process management is recommended in order to be able to fully understand the source code and implement new tightly coupled (OSIRIS) services. For the integration of WSDL/SOAP based services (loosely coupled) into custom processes no knowledge of C++ and the OSIRIS source code is needed.

Pre-Requisites for Installation:

At least one PC with Microsoft Windows 2000 or newer and an available SQL Server database server preferable at the same machine is needed to run the OSIRIS platform. Additional, memory and disc requirements are mostly demanded by the services one wants to install. The OSIRIS core services and the middleware layer itself are not resource demanding.

Conditions of use:

Contact the authors.

OSIRIS-SE – OSIRIS Stream-Enabled

Contact points:

Heiko Schuldt (heiko.schuldt@unibas.ch)
Gert Brettlecker (gert.brettlecker@unibas.ch)
University of Basel, Switzerland

Technical Contact Point:

Gert Brettlecker (gert.brettlecker@unibas.ch)

Type of Software:

STABLE PROTOTYPE

Descriptive Keywords:

Java, Data Stream Management, Process Management, Mobile Devices, Sensor Data, Telemonitoring, Reliability, Checkpointing

Potential Use and Applications:

Data Stream Management (DSM) addresses the continuous processing of sensor data. DSM requires the combination of stream operators, which may run on different distributed devices, into stream processes. Due to the recent advantages in sensor technologies and wireless communication, DSM is increasingly gaining importance in various application domains.

Especially in healthcare, the continuous monitoring of patients at home (telemonitoring) can significantly benefit from DSM and produces data which needs to be stored in an eHealth Digital Library. A vital requirement in telemonitoring is however that the infrastructure for DSM provides a high degree of reliability. This may be achieved by a novel approach to efficient and coordinated stream operator checkpointing supporting reliable DSM while maintaining the high result quality needed for healthcare applications.

General Description:

OSIRIS-SE (SE for Stream-Enabled) is an extended version of the OSIRIS system (Open Service Infrastructure for Reliable and Integrated process Support). OSIRIS offers an infrastructure for reliable distributed (P2P) process execution. The development of OSIRIS-SE has been started at UMIT in Hall in Tyrol and is continued at the University of Basel. OSIRIS-SE, in addition to OSIRIS, allows for the P2P execution of stream-processes in a distributed environment. Stream processes are combinations of continuously running data stream operators, that are interconnected to perform a certain data stream processing task on incoming sensor data.

OSIRIS-SE is programmed completely in Java and runs on various platforms like Windows and Linux, including also PDAs with MS Windows Mobile 2003. The nodes that are participating in an OSIRIS-SE network are also called providers, because according to the service oriented approach, each participating node provides OSIRIS-SE services to the network. Providers are equipped with a local middleware layer (the OSIRIS-SE software layer). The OSIRIS-SE layer running at each node also contains a process execution engine. Thus these providers are able to run stream processes as well as traditional processes based on discrete service calls. The process execution is based on locally replicated meta-data from core services, like service repository or process description repository. This replication allows for decoupling meta-data distribution and process execution. Therefore, processes –both streaming and traditional– can be executed without contacting any central service at runtime. Moreover, OSIRIS-SE implements a reliable and efficient operator checkpointing for data stream operators within a stream process. Since data stream operators are continuously running and aggregating a state during runtime (e.g., the average of the last 10 blood pressure measurements) one has to access and migrate this state when a failure affects a running operator instance.

Technical Description:

OSIRIS-SE is implemented completely in Java. At a basic level it offers a generic message oriented middleware which can deal with various pluggable transport layers, as a proprietary TCP socket transport, Web-services based transport, or a peer-to-peer based JXTA transport. Based on the messaging system, OSIRIS-SE offers a component framework which allows for configuration, lifecycle management, and service execution. Message processing at component level is done in a service based concept, where service interaction is in request-response style. Moreover, OSIRIS-SE offers a dedicated agent component, which is responsible for communication between OSIRIS-SE providers and offers pipeline-based message processing. The agent has an incoming and outgoing message processing pipeline which also supports arbitrary plugging of dedicated pipeline handlers. An example of a pipeline handler is the process execution engine running on each provider. Additional pipeline handlers are e.g., responsible for message routing and load balancing. This pipeline-based message processing allows to easily extend OSIRIS-SE. A dedicated gateway component offers the ability to integrate non-OSIRIS-SE services, like standard WSDL/SOAP services in a loosely coupled way. The data stream management functionality of OSIRIS-SE is implemented by offering an additional stream component framework, which offers the design of data stream operators. Extended functionality of the message transport system allows dealing with data streams and offering a reliable buffered first-in/first out transport of data stream elements between operators. At the level of the transport layer data stream elements are dynamically grouped into messages. Moreover, the process execution engine is able to create, maintain, and stop data stream processes. Finally, running operator instances are monitored and their state is saved at a suitable backup provider in an efficient way.

Required User Skills:

Fair knowledge of Java, conceptual knowledge of message oriented middleware, SOA, process management, and data stream management is recommended.

Pre-Requisites for Installation:

The software is based on Java. The core parts of the infrastructure are implemented to work even on Java 1.3 in order to be compatible with mobile devices. If a current Java 1.5 JVM is available full functionality is offered. The hardware requirements are very low since the software was designed to run also on mobile devices. For mobile devices 64 MB available free memory is sufficient.

Conditions of Use:

Contact the authors.

P2P DL - Peer to peer Digital Library

Contact Point: Aggeliki Dimitriou (angela@dblab.ece.ntua.gr) National Technical University, Athens, Greece
Technical Contact Point: Aggeliki Dimitriou (angela@dblab.ece.ntua.gr)
Type of Software: PROTOTYPE
Descriptive Keywords: peer-to-peer, digital library, query reformulation, mappings
Potential Use and Applications: Dynamic networks of DL systems
General Description: Our prototype supports data management in a dynamic network of autonomously managed digital library nodes. Autonomy in this context refers to the following objectives: <i>Joining and leaving the library.</i> Each DL node chooses itself when to join and/or leave the library network. The system can gracefully adapt to these joins and departures without any global structural knowledge. <i>Data management.</i> A DL node is responsible for managing its own local data, as well as maintaining information (expressed as schema mappings) about data residing at other peers. Every DL node in our system should store data organized in the RDFS model. However, there are no restrictions on content representation nor any global schema information, i.e., a DL node may use its own RDFS representation. <i>Query processing.</i> Queries can be issued at any DL node and propagated to other DL nodes in its neighborhood (and so on). A query, prior to its propagation to other DL nodes, is reformulated using schema mappings in order to match schema information at the remote DL nodes. For more information: http://milos.dblab.ece.ntua.gr/p2pdl/

Technical Description:

Our prototype has been built on top of the JXTA PP software (<http://www.jxta.org>). The DL network is organized in JXTA groups. At least one DL node in each group provides the necessary scope to message propagation. Every such DL node maintains JXTA advertisements for schema information and mapping information.

Our system implements the following facilities:

Setup. After its creation, every new DL node makes its schema advertisement (with its RDFS schema primitives) public to any other DL node.

Joining the network. To join the network, a DL node obtains the list of available DL nodes that participate in the network. Then, the DL node's administrator (using a mapping wizard) selects some of those DL nodes (i.e., its neighborhood) and determine the mappings between those nodes and the new DL node. Mappings are stored in the mapping advertisements of the new DL node.

Querying. Queries can be issued at any DL node and propagated to other DL nodes in its neighborhood (and so on). Query reformulation is assisted using a query wizard. In each DL node, we provide a Sesame RDF query engine (<http://www.openrdf.org/>). Every query formulated using the wizard is transformed to a Sesame RDF query to be executed on the RDF data. A query, prior to its propagation to other DL nodes, is reformulated by renaming its RDFS properties according to the mappings defined in the mapping advertisement, in order to match schema information at the remote DL nodes.

For more information: <http://milos.dblab.ece.ntua.gr/p2pdl/>

Required User Skills:

No special skills needed.

Pre-Requisites for Installation:

JDK 1.4.2, JXTA 2.3.1, Sesame 1.2.6

Conditions of use:

Free downloading. For instructions please refer to <http://milos.dblab.ece.ntua.gr/p2pdl/>

RPextract - Music Feature Extractor

Contact points:

Thomas Lidy (lidy@ifs.tuwien.ac.at)
Andreas Rauber (rauber@ifs.tuwien.ac.at)
Technical University of Vienna, Austria

Technical Contact Points:

Thomas Lidy (lidy@ifs.tuwien.ac.at)

Type of software:

PROTOTYPE

Descriptive Keywords:

Audio feature extraction, music descriptor, music genre classification, music similarity retrieval, Rhythm Patterns, Statistical Spectrum Descriptor, Rhythm Histograms

Potential Use and Applications:

Content-based access to audio files, particularly music, requires the development of feature extraction techniques that capture the acoustic characteristics of the signal. The RPextract music feature extractor extracts features of music files capturing aspects related to rhythm and timbre. This enables the computation of similarity between pieces of music, useful for a large range of potential Music Information Retrieval tasks, such as retrieval of similar music from an archive, identification of pieces of music, music genre recognition, artist detection, organization and classification of music libraries.

General Description:

Content-based access to audio files, particularly music, requires the development of feature extraction techniques that capture the acoustic characteristics of the signal, and that allow the computation of similarity between pieces of music. At TU Vienna - IFS three different sets of descriptors were developed:

- *Statistical Spectrum Descriptors*: describe fluctuations by statistical measures on critical frequency bands of a psycho-acoustically transformed Sonogram
- *Rhythm Patterns*: reflect the rhythmical structure in musical pieces by a matrix describing the amplitude of modulation on critical frequency bands for several modulation frequencies
- *Rhythm Histograms*: aggregate the energy of modulation for 60 different modulation frequencies and thus indicate general rhythmic in music

The algorithm considers psycho-acoustics in order to resemble the human auditory system. The feature extractor processes au, wav, mp3 and ogg files. Feature vectors are output in SOMLib format, an ASCII format containing descriptive headers.

Details about the algorithm are available at:

<http://www.ifs.tuwien.ac.at/mir/audiofeatureextraction.html>

A usage guide is available at

http://www.ifs.tuwien.ac.at/mir/howto_matlab_fe.html

Details about the output format are available at

http://www.ifs.tuwien.ac.at/~andi/somejb/download/ghsom_guide.html#input.

Technical description:

The feature extraction algorithm is as follows: in a pre-processing step the audio signal is converted to a mono signal and segmented into chunks of approximately 6 seconds. Typically, the first and last one or two segments are skipped and from the remaining segments every third one is processed. For each segment the spectrogram of the audio is computed using the short time Fast Fourier Transform (STFT). The Bark scale, a perceptual scale which groups frequencies to critical bands according to perceptive pitch regions, is applied to the spectrogram, aggregating it to 24 frequency bands. The Bark scale spectrogram is then transformed into the decibel scale. Further psycho-acoustic transformations are applied: Computation of the Phon scale incorporates equal loudness curves, which account for the different perception of loudness at different frequencies. Subsequently, the values are transformed into the unit Sone. The Sone scale relates to the Phon scale in the way that a doubling on the Sone scale sounds to the human ear like a doubling of the loudness. This results in a Bark-scale Sonogram - a representation that reflects the specific loudness sensation of the human auditory system. From this representation of perceived loudness statistical measures (mean, median, variance, skewness, kurtosis, min and max) are computed per critical band, in order to describe fluctuations within the bands extensively. The result is a Statistical Spectrum Descriptor. In a further step, the varying energy on the critical bands of the Bark scale Sonogram is regarded as a modulation of the amplitude over time. Using a Fourier Transform, the spectrum of this modulation signal is retrieved. The result is a time-invariant signal that contains magnitudes of modulation per modulation frequency per critical band. This matrix represents a Rhythm Pattern, indicating occurrence of rhythm as vertical bars, but also describing smaller fluctuations on all frequency bands of the human auditory range. Subsequent to the Fourier Transform, modulation amplitudes are weighted according to a function of human sensation of modulation frequency, accentuating values around 4 Hz. The application of a gradient filter and Gaussian smoothing potentially improves similarity of Rhythm Patterns, which is useful in classification and retrieval tasks. A Rhythm Histogram is constructed by aggregating the critical bands of the Rhythm Pattern (before weighting and smoothing), resulting in a histogram of rhythmic energy for 60 modulation frequencies. The feature vectors are computed for a piece of audio by taking the median of the descriptors of its segments.

Required User Skills:

- Edit options file with a Text Editor
- Minimal knowledge of Matlab (only to start algorithm)
- Alternatively, knowledge of how to start algorithm from a shell script
- Knowledge of how to process a csv data file

Pre-Requisites for Installation:

Windows or Linux

Matlab 6.1 or above, with Signal Processing Toolbox installed

For extracting MP3, OGG or FLAC files, additional binaries are required for decoding.

Conditions of use:

Available for research purposes, please contact the authors.

Sightseeing4U - Development of personalized mobile multimedia applications

Contact Point: Jochen Meyer (jochen.meyer@offis.de) OFFIS, Germany
Technical Contact Point: Ansgar Scherp (ansgar.scherp@offis.de)
Type of Software: STABLE PROTOTYPE
Descriptive Keywords: Travel and tourism, multimedia presentation, generation, personalized multimedia, mobile multimedia
Potential Use and Applications: Development of arbitrary personalized mobile multimedia tourist guides.
General Description: Mobile applications such as mobile tourist guides that provide tourists with location-based information today mostly aim to adapt the multimedia content to the different end user devices. More and more, these applications also exploit positioning information like GPS to guide the user on the trip. What is still lacking, however, is a personalization of the content to the interests and preferences of the individual tourist and the characteristics of the used end device. However, such a personalization increases the application's complexity since every individual alternative have to be considered and implemented. To provide substantial support for the development of personalized (mobile) multimedia applications, we developed a domain independent software framework for an efficient and cost-effective development of personalized mobile multimedia applications. We illustrate the framework in the specific domain of personalized mobile tourist information.
Required User Skills: Being familiar with using a PDA.
Pre-Requisites for Installation (hardware and software) Hardware: Pocket PC Software: Jeode Java Runtime Environment
Conditions of Use: Contact the authors.
Additional Information: Scherp A. and Boll S. 2004. Generic support for personalized mobile multimedia tourist applications. In Proceedings of the 12th ACM International Conference on Multimedia. New York (USA), 10-16 October 2004. New York: ACM Press. 178-179.

SOMToolbox - Implementation of Self-organizing map models

Contact Points:

Andreas Rauber (rauber@ifs.tuwien.ac.at)
Rudolf Mayer (mayer@ifs.tuwien.ac.at)
Technical University of Vienna, Austria

Technical Contact Points:

Rudolf Mayer (mayer@ifs.tuwien.ac.at)
Thomas Lidy (lidy@ifs.tuwien.ac.at)

Type of Software:

PROTOTYPE

Descriptive Keywords:

Self-organising map (SOM), Clustering, Visualisation.

Potential Use and Applications:

The Java SOMtoolbox can be used to cluster e.g. text, audio, image or other data that can be described in vectorial form. It provides a wealth of tools to inspect the clustering results in an innovative, interactive approach .

General Description:

The Java SOMtoolbox provides implementations of Self-organising map models, among them for the classical SOM, the Growing Grid, and the Growing Hierarchical SOM (see <http://www.ifs.tuwien.ac.at/~andi/ghsom/>). The SOM provides a clustering of the high-dimensional input space to an easier understandable two-dimensional output space. Labeling algorithms for the maps are also provided (only applicable if the data has some meaning to extract from the features, e.g. in Text mining). The software can handle any kind of media objects, as long as they are describable in a vectorial format.

In addition, the toolbox offers a rich desktop client for exploring the trained maps in a novel and interactive way. The interface allows for zooming and panning, and selecting regions, paths or single data objects, to create and export e.g. play lists in case of audio data. A wealth of visualisation techniques is available, e.g. the U-Matrix, D-matrix, Component planes, Smoothed Data Histograms (SDH), Class visualisations.

Screenshots and additional information can be found at:

<http://www.ifs.tuwien.ac.at/mir/#PlaySOM>

<http://www.ifs.tuwien.ac.at/dm/#GHSOM>.

Technical Description:

The Self-Organising Map (SOM) is a unsupervised machine learning algorithm that provides a mapping from a high-dimensional input space (e.g. extracted music features) to a low-dimensional, usually two-dimensional, output space. In this mapping, the SOM is topology preserving, meaning that items that are close in the input space will also be closely located in the output space, while objects that are distant in the input space, and therefore dissimilar, will be mapped onto different regions of the SOM. The low-dimensional output space allows for an easier understanding of the underlying cluster structure in the data. Due to this, the SOM is a very interesting method to analyse data, and used in many different applications. With visualisation techniques as the U-Matrix or the SDH, the cluster boundaries can be visualised. The software provides those visualisations, plus the D-Matrix, component planes, visualisations of training quality, class distribution visualisations, and others.

More details on the SOM can be found at <http://www.ifs.tuwien.ac.at/~andi/ghsom/>.

Required User Skills:

Handling the command line scripts; if new maps are trained, possible data pre-processing and conversion to the (plain-text) file format needed by the training algorithms.

Pre-Requisites for Installation:

- 25 MB disc space
- 256 MB RAM for the SOM training
- 512 MB RAM for the SOM Viewer applications

Conditions of Use:

The software may be used for research purposed. Please contact the authors for more information.

Transformation4U - Transformation of Multimedia documents for mobile applications

Contact Point: Jochen Meyer (jochen.meyer@offis.de) OFFIS, Germany
Technical Contact Point: Ansgar Scherp (ansgar.scherp@offis.de)
Type of Software: STABLE PROTOTYPE
Descriptive Keywords: Multimedia content transformation, multi-channel creation of multimedia content, web service.
Potential Use and Applications: Applications that need to transform (personalized) multimedia content into different multimedia presentation formats.
General Description: With Transformation4U, we propose a web service for the DelosDLMS that provides for a multi-channel authoring of (personalized) multimedia presentations in different presentation formats targeted at different (mobile) end devices. The service bases on the Presentation Format Generators component of the MM4U framework [SB05a+b], which provides application-independent transformation algorithms and transformation rules to bring multimedia content in the internal multimedia content representation model of the framework into the syntax and features of the different concrete presentation formats we find today such as SMIL, SVG, and Flash. Employing the Transformation4U web service and its underlying Presentation Format Generators component allows remote multimedia applications to author (personalized) multimedia content in the internal multimedia content representation model of the MM4U framework and to bring this content over the last mile to the different (mobile) end users' device settings.

Technical Description:

The Transformation4U web service is hosted on a Unix server, provided by the OFFIS Institute for Information Technology in Oldenburg, Germany. A description of the web service is available at: <http://power1.offis.uni-oldenburg.de:60062/mm4uapp/t4uservice.html>

The Transformation4U web service itself is available at:
<http://power1.offis.uni-oldenburg.de:60062/axis/Transformation4U.jws>.

The WSDL description of the web service is available at:
<http://power1.offis.uni-oldenburg.de:60062/axis/Transformation4U.jws?wsdl>

To apply the Transformation4U web service, an application must provide the multimedia content in a XML-based document. This XML document will be validated against the MM4U framework's composition model. This composition model is defined in a DTD that is available at:
<http://www-is.informatik.uni-oldenburg.de/~mm4u/Specifications/mm4u.dtd>

Note: Currently, only a selected set of the multimedia composition functionality of the MM4U framework is provided by the Transformation4U web service. The example documents listed at the end of this document provide a good overview of that functionality.

To transform a multimedia document you can either send the document's XML description to the web service or specify an URL referring to that document. To specify the targeted output format, you can determine the output format directly when calling the web service. Once, the user profile server of DS-MIRF is provided and integrated into the Transformation4U server, one can also provide a user profile ID instead of directly specifying the output format. In that case, the Transformation4U web service will determine based on the information stated in the given user profile, which output format shall be selected. As a consequence of the description of the web service, Transformation4U will provide four different transformation methods:

- Transformation by providing a XML document together with the output format.
- Transformation by providing a URL reference to the document together with a target output format.
- Transformation by providing a XML document together with a user profile ID.
- Transformation by providing a URL reference to the document together with a user profile ID.

Sample Client: A sample client has been implemented in Java to demonstrate and to test the Transformation4U web service. The sample client can be downloaded from:
<http://power1.offis.uni-oldenburg.de:60062/mm4uapp/t4uservice.html>

The client bases on the libraries of the Apache Axis project, which can be found at:
<http://ws.apache.org/axis/>

The sample client consists of the following Java classes:

- Transformation4U.java
- Transformation4UService.java
- Transformation4UServiceLocator.java
- Transformation4USoapBindingStub.java
- TestClient.java.

Most of the classes have been automatically generated by the WSDL-to-Java tool that is provided by Apache's Axis project. The interesting part of the client from an application developer's point of view is the file `TestClient.java`. Here, the four variants to call the Transformation4U web service are demonstrated.

<p>Example Documents: For testing purpose and to getting started using the Transformation4U web service, we are currently providing a set of example multimedia documents. These example documents cover different aspects of multimedia content authoring:</p> <ul style="list-style-type: none"> • Testing font capabilities: http://www-is.informatik.uni-oldenburg.de/~mm4u/Examples/font.mm4u.xml • A simple slideshow: http://www-is.informatik.uni-oldenburg.de/~mm4u/Examples/slideshow.mm4u.xml • A video with subtitles: http://www-is.informatik.uni-oldenburg.de/~mm4u/Examples/video.mm4u.xml
<p>Required User Skills: Programming web services.</p>
<p>Pre-Requisites for Installation: None</p>
<p>Conditions of use: Contact the authors.</p>
<p>Additional Information: Scherp A. and Boll S. 2005. MM4U: a framework for creating personalized multimedia content. In U. Srinivasan and S. Nepal (eds.), <i>Managing Multimedia Semantics</i>. IRM Press. 246-287. Scherp A. and Boll S. 2005. Paving the last mile for multi-channel multimedia presentation generation. In: Y.-P. Phoebe Chen (ed.), <i>Proceedings of the 11th International Conference on Multimedia Modeling (MMM 2005). Melbourne (Australia)</i>. Washington: IEEE Computer Society. 190-197.</p>

xSMART - Semi-automatic context-aware authoring tool

Contact point: Jochen Meyer (jochen.meyer@offis.de) OFFIS, Germany
Technical Contact Point: Ansgar Scherp (ansgar.scherp@offis.de)
Type of software: STABLE PROTOTYPE
Descriptive Keywords: Multimedia authoring, context-driven authoring, semi-automatic multimedia content creation.
Potential Use and Applications: Context-driven creation of multimedia content.
General Description: In recent years, many highly sophisticated multimedia authoring tools have been developed. Up to today, these system's integration of the targeted user context, however, is limited. With our Context-aware Smart Multimedia Authoring Tool (xSMART) we developed a semi-automatic authoring tool that integrates the targeted user context into the different authoring steps and exploits this context to guide the author through the content authoring process. The design of xSMART allows that it can be extended and customized to the requirements of a specific domain by domain-specific wizards. These wizards realize the user interface that meets best the domain-specific requirements and effectively supports the domain experts in creating their content targeted at a specific user context.
Required User Skills: Basics in using tools like Microsoft PowerPoint.
Pre-Requisites for Installation: Software: Java Runtime Environment
Conditions of use: Contact the authors.
Additional Information: Scherp A. and Boll S. 2005. Context-driven smart authoring of multimedia content with xSMART. In <i>Proceedings of the 13th Annual ACM International Conference on Multimedia</i> . New York: ACM Press. 802-803.