

DISCOVER THE POWER OF VITAL

The Solution for Your Digital Collection Management

Authored by Heather Myers and Keara Axelrod

Vital is a digital object repository and management system designed for universities, libraries, museums, archives and information centers. This software is designed to simplify the development of digital collections and to provide seamless online search and retrieval of information for administrative staff, contributing faculty and end-users. Vital provides a way to broaden access to your institution's valuable resources that were once only available at a single location and to a finite number of patrons. By eliminating the limitations traditionally encountered by information seekers, this technology grants access to materials for all authorized end users, from professional researchers to recreational learners.

Vital builds on the Fedora™ repository architecture by providing extensive workflow extensions, management utilities, indexing enhancements, advanced searching capabilities and specialized content displays. Using Fedora™-defined web services, Vital provides a mechanism for your organization to create tools, enhance the functionality provided by Innovative, or leverage the open-source community for future applications. Together, the package is known as Vital. Vital delivers a flexible and highly customizable solution for managing the digital resources of any type of information center.

Vital provides every feature required to handle large text and rich content collections—ingesting, storing, indexing, cataloging, searching and retrieving. Vital takes advantage of technology standards such as XML, TEI, EAD, MODS, MADS and Dublin Core to easily describe and index an assortment of electronic resources. Vital conforms to common Internet data communications standards such as TCP/IP, HTTP, SRU, SOAP and FTP. Additional standards supported with Vital and the Fedora™ Object Repository include OAI-PMH, MARCXML, and MIX (Metadata for Images in XML Schema).

■ FUNCTIONAL INFORMATION

VITAL ACCESS PORTAL FUNCTIONALITY

The Vital Access Portal provides a simple, public web interface to your repository content. The Access Portal allows for complex searching of objects against cataloged descriptive metadata fields and optional searches against an object's full text. With functionality such as relevance ranking, user-defined search results display screens, and the ability to display specialized content types, the Vital Access Portal provides a powerful content discovery tool for your users.

Notable functionality includes:

Robust searching functionality – The Vital Access Portal provides users with powerful search options for content discovery. Users can discover content using simple keyword searches or they can perform more precise searches using included functionality for Boolean and phrase searching. Searches can be performed against the repository's full-text objects and against pre-defined metadata fields available through user-friendly drop-down lists. With support for filtered searches, users can perform targeted searches against specific metadata elements, collections, resource types, etc., and can limit results against defined metadata elements such as 'Peer Reviewed' and 'Full Text Availability'. Vital also allows users to re-sort the results by the metadata elements that are most important to them, such as Date, Author, Peer Reviewed Status, etc.

Browsing content – The Vital Access Portal allows users to browse content in the repository without having to perform a targeted search. Users may browse by Author, Title, Subject, Collection, Resource Type, Date, Academic Discipline, etc., and navigate through repository objects in an intuitive A-Z type display. You can also create curated collections of items for patrons to browse.

Browse categorized content 'facets' – Along with the standard listing of objects that are displayed as the result of a search query, the Vital Access Portal also displays a summary of frequently used metadata elements to help users further refine their search to return only the most relevant results. The Access Portal can be customized to display typical metadata categories like Author/Creator, Subject, Format Type, and Resource Type. Each of these categories contains a listing of these metadata 'facets' or search refinement filters which are ranked by occurrence. Clicking on an element here will return a further refined list of objects from within the original search results that adhere only to that filter. These metadata categories are updated with each set of filtered results, giving the end-user an ever-increasing level of specificity in their searches. A typical usage of this feature might be to perform a repository-wide keyword search, and then further refine the results using the search facets to show only the objects that are categorized as journal articles and written by a particular author.

Pre-defined search targets – The Vital Access Portal supports search 'tracings', which allow for certain metadata elements, such as author, subject, etc., to be searched directly from a Vital object. This enables users to generate a new search directly from a hyperlinked metadata element from within a repository object to easily discover other content of interest.

Repository highlighted content – The Vital Access Portal allows institutions to showcase repository content to their users. Such highlighted content may be in the form of a randomly selected ‘Work of the Day’, or may be focused on specific elements such as a particular collection, recent additions, a particular author’s work, or any other type of pre-defined search.

Most popular content – The Vital Access Portal provides end-users with the ability to view, at a glance, the most popular content in the repository through aggregations of usage statistics. Vital is pre-configured to display the most-accessed authors and objects, but administrators can configure the Access Portal to display other desired aggregations, such as the most-accessed content within a particular collection or academic discipline. Patrons will have direct access to the content from these screens, as well as statistics for how often the content was viewed, downloaded, cited, etc.

Highlighted author content – The Vital Portal provides support for the ability to showcase the work of your institution’s content contributors through the inclusion of an author page, which displays all repository works by the author with hyperlinks directly to the content. This page also includes access statistics to show how many hits, downloads, citations, etc. have been requested for each work in the repository. Vital also allows the authors to copy a simple URL for this page to distribute to their peers (e.g. in an e-mail footer). Each work by the author can be displayed in a bibliography format (with hyperlinks to the Vital object) for simple inclusion in a researcher’s work. Vital also provides for the direct linking to this bibliography view from an external website (e.g. from an author’s personal or departmental website). Provided as simple JavaScript for inclusion in the body of the HTML page, this Bibliography Script acts as a live feed to the author’s bibliography from that external web page.

Integrated RSS feed – The Vital Access Portal provides for an RSS feed of selected Vital content. Through this feature, researchers can choose to receive updates regarding content of interest to them instantly upon its availability in the repository. Such areas of interest might include new works by a particular author, works within a particular subject area, or new contributions from a particular academic research unit. Researchers can also perform filtered searches and set the RSS feeds to alert the researcher of new content that meets those specific search criteria.

Hi-Resolution Image Navigation – The Vital Access Portal provides support for viewing detailed imagery, large format material, digitized cartography and other wavelet-compressed images encoded in either JPEG2000 or MrSID image formats. These format types allow for very large images to be compressed with little or no loss in quality and delivered over the web without the delay typical to the download of large image files. The images can be viewed within the Vital Access Portal, which includes familiar zooming/navigation functionality,

including ‘live’ panning, without the need for patrons to install separate browser plug-in applications.

Configurable displays for search results – The Vital Access Portal can be configured to display search results in a variety of ways, such as thumbnail images with brief data, data in paragraph form, tabular data, etc. The interface views are completely configurable through industry standard technologies such as XML, XSLT, XHTML and CSS.

Page Turning Interface – The Vital Access Portal provides the ability to view complex objects with multiple datastreams – without leaving the Access Portal. This advanced interface provides support for JPEG2000, MrSID, TIFF, GIF, JPG, and PNG files to be viewed in a predefined, sequential order. In addition, the Access Portal can be configured to automatically display PDF documents in the Page Turning Interface. This feature is ideal for displaying content such as books, historical correspondence, theses and other like content in a fashion that mirrors the original source object. Support is provided for TEI structured metadata and ‘slideshow’ of collection content. A navigation bar features the ability to jump to specific pages.

Language support & UNICODE compliance – The Vital Access Portal is fully UNICODE compliant and supports the storage and display of multilingual, non-roman character sets, including languages that read right-to-left. In addition to repository content, the Access Portal itself ships with translations for English, Spanish, Catalan, French, German, Greek and Arabic. Users simply need to select from the drop-down menu of options to enable their language of choice. Vital includes the ability to sort Chinese records as well.

Support for viewing content in external applications – Since Vital has no limitations on the types of content that can be held in the repository, the Vital Access Portal allows users to view specialized content with associated plug-in applications. This enables patrons to choose an application for which they are most familiar, or the content’s native application.

Fully customizable interface design – The Vital Access Portal has been designed to allow for flexibility in configuring the display to match a desired look and feel. The interface display may be customized via HTML and XML/XSL to blend seamlessly with your current web page or to create custom front-ends to reflect a particular project, web exhibition or other applications.

Support for thumbnail display – The Vital Access Portal allows for configurable display of thumbnails from both the search results and full-object display screens to allow users to preview repository content. Support for such image types as GIF, JPG, JPEG2000, PNG and MrSID is included. Additionally, thumbnails may be displayed for PDF files and some video file formats. The Access Portal also allows users to choose to view image-based content in a ‘lightbox’ display with minimal metadata as opposed to the standard list view.

EXPOSING YOUR CONTENT

Today it is more important than ever to make the content in your repository findable and readable not only to visitors to your site, but by harvesters. Vital supports persistent identifiers, and is compatible with standard data exposure protocols.

Handles server integration for persistent identifiers -

The Handles System is a comprehensive system for assigning, managing, and resolving persistent identifiers, known as "Handles," for digital objects and other resources on the Internet. Handles can be used as Uniform Resource Names (URNs). Once you have registered with the Corporation for National Research Initiatives (CNRI), you will be assigned a prefix. This prefix will identify your handles service to the Global Handle Registry. Vital will use this prefix to create and assign Handles to digital objects and to individual datastreams within the Fedora™ repository. Users can resolve back to digital content using the Handle that is assigned, even if the location of the content changes.

Google Indexing and Exposure – Vital facilitates the exposure of repository content to Google via a MARCXML to XHTML conversion of metadata. The system can differentiate users from a Google crawler so that users can be redirected to the object within the Vital Access Portal. You can also configure Vital to expose academic content to Google Scholar.

Exposure to Harvesters – Vital supports the exposure of repository content to harvesters through the standard OAI-PMH protocol. The system can support the exposure of individual objects or sites can define entire 'sets' such as individual collections of content for harvesting. Metadata exposed to these harvesters can be provided through the native Dublin Core metadata, or you can choose to expose indexed fields from an alternate descriptive metadata source file, such as MARCXML, for harvesting.

SRU Interface for exposure of repository content - The SRU (Search/Retrieve URL) services define a method for interacting with and retrieving information from remote databases. This technology is ideally suited for abstracting a 'gateway' to search otherwise complicated protocols or proprietary access mechanisms. Since the search results and other information are transferred using XML, SRU has become a tool for creating bridges to more technically complicated protocols, such as Z39.50. Vital includes an SRU service to expose the content of the repository to portals, federated search tools, and other search engines supporting this protocol. Vital allows institutions to configure the objects and object information that they wish to be available to an SRU search query. Administrators may choose to allow any Vital object information, including full-text datastreams, to be available for resource discovery, but still retain control over what metadata about the object is returned to the search interface (e.g. Title, Author, or Handle only).

Content Models – Vital supports enhanced displays of objects according to a content model assigned to the object. Each content model can have a different display in the Access Portal. When browsing to an object's page, the page is automatically rendered based on the object's content model.

OpenURL Support – The Vital Access Portal can be configured to generate an OpenURL link to a target server for link resolving. The OpenURL link will allow users to find the most appropriate copy of content among the institution's resources for objects that are stored in Vital. This feature is particularly useful for repositories that contain harvested metadata only, or pre-print repositories that may not contain the most current version of a resource.

Citation Export – The Vital Access Portal contains functionality to build a citation from an object's metadata and allow for e-mail of the citation as plain text or in a standard format such as EndNote. Users can also add multiple objects to a 'cart' to allow for the export of multiple citations as a single export. Since Vital also supports the assignment of Handles identifiers to objects, users will also be able to export this persistent identifier for the content in their citations. This allows for a persistent link to the cited work to be available in the bibliography of a subsequent work.

SOPHISTICATED METADATA MANAGEMENT

In addition to the simple data display the average user has come to expect, Vital supports the storage and display of more complex data of particular interest to archivists and librarians.

Relationship Browser: The relationship display view, or relationship browser, graphically displays the RELS-EXT datastream state of the current object using arrows to indicate the horizontal and vertical relationships of the object.

Display of SEE and SEE ALSO references - When you do a search, if the search term matches any authority object, Vital will display the variants as see references and related terms as see also references for that authority.

Enhanced displays for EAD, DC and MARC metadata – The Vital Access Portal ships with customized displays for Encoded Archival Description (EAD), Dublin Core (DC) and MARCXML metadata to provide end-users with maximum utility inherent to these format types.

VITAL ACCESS PORTAL ADMINISTRATIVE MODULE

The Vital Access Portal features a web-based administrative module that allows system administrators and collection-building staff robust functionality for managing objects within the Vital repository. Functionality exists for ingesting and editing of content and metadata, searching, reporting, exporting, and more, all available through an internet browser. The administration module has been designed to empower administrators

with a complete toolset for content aggregation in an intuitive, easy-to-use interface. Since Vital has no limitations on the content formats that can be added, facilities are available for integration with third-party client applications for viewing/editing content in its native format. Administrative users can edit content in the appropriate interface and save the changes directly to the Vital repository.

Notable functionality includes:

Administrative Reporting – Vital provides administrators with a robust toolkit to perform reporting against repository content. Vital's reporting interface allows for the creation of reports against specific metadata elements, usage statistics, or a more generic report against any type of object content, including full-text, using keyword queries. The Vital web interface allows for inputting multiple search criteria to achieve the results needed, and allows for control over which data elements to include in the report. Reports may be generated on an ad-hoc basis or can be scheduled to run at a pre-defined time. The report results are archived by Vital for later use, and the report queries themselves are saved as templates and can be modified at any time to either overwrite or create a new report query. Users may choose to view/export the results of the reports as HTML, XML or CSV files. The latter may easily be viewed in a spreadsheet application such as MS-Excel for further manipulation or sorting.

Access and Usage Statistics – Vital provides automatic logging of repository usage statistics and provides a configurable interface to these statistics through the Administrative Module. Users can see, at a glance, how many hits, visitors, and links have been logged for each object page. The interface allows users to choose a time period desired for the statistics (e.g. today, this month, this year, etc.), dynamically reproduces the results both as an aggregate total and graphically as appropriate for the requested date, and provides links to additional information regarding each statistical element. Administrators have the ability to create/export reports against the usage statistics through the Administrative Reporting interface.

Consortia Support – A single Vital installation can support an unlimited number of institutions. Each individual institution can be configured to have its own users, collections and look-and-feel. Administrators have the choice to share objects that belong to its institution or only allow access to users belonging to that institution.

Automated verification of linked resources – Vital allows administrators to run a service that queries the repository content for external URLs to verify the integrity of these links. The system will generate a report of any problems noted with these links, including external URLs that may have moved or changed since the content was originally added to Vital. The report provides detailed information regarding the noted problem for the link (e.g. 404 page not found error, 500 internal server error, etc.) and can be regenerated on the fly to ensure that attempts to correct problems have been successful.

User Activity Logging – Vital allows administrators to view recent activity for users of the repository. Session information is available for each user to show the occurrence and timing of events and links to the event pages within the system.

Global Diagnostics Page – A simple interface that displays repository status information and diagnostics for the Vital system, making the first step of troubleshooting as simple as possible.

Repository Indexing – Vital allows administrators to control the status of background indexing services directly from the Administrator Module to ensure that the new content is indexed in a timely fashion and at the administrator's discretion.

Create custom indexes to enhance resource discovery – Vital allows for customized indexes to be created for granular control over resource discovery. Features of indexing allow the repository administrators greater control in content exposure, thus empowering end-users more accurate results when searching and retrieving content. XPath definitions can be created to allow specific portions of XML content in objects and datastreams to be targeted for searching.

Annotation of page content – Vital allows administrators a quick and easy way to add additional content to Vital screens through a Wiki-style editor. Administrators can add specific information to the top of every page visible to end-users either as plain text or as HTML content. This feature allows for quickly adding news, help information, links to new or highlighted content, repository maintenance notices, etc. The interface also allows administrative users to preview modifications as they would be seen to end-users before applying the changes.

CONTROLLING ACCESS TO CONTENT

Vital has tools that let you control who has access to your content. This means that you can control which staff members have which administrative controls, as well as protect access to privileged content via log-in.

Users and Groups – Vital provides an interface for creating users and groups, setting users to groups, defining the permissions for each group, as well as defining permissions at the object and datastream levels. This lets you control patron access to particular parts of your collection, and lets you assign different administrative functions to different administrative users.

User Authentication via LDAP - Institutions need a way to ensure the identity of certain users within the Vital System. By supporting the Lightweight Directory Access Protocol (LDAP) as a mechanism to authenticate users, Vital allows sites to use their existing LDAP-based authentication system to manage users, thus reducing the overhead of maintaining separate systems. System administrators have the ability to see users logged into the system and report on user activity.

User Authentication via Shibboleth – Vital can also be configured for users to authenticate via Shibboleth. Shibboleth is an open source implementation of the Security Assertion Markup Language (SAML), which provides a Web Single Sign-On (SSO) for end users both within their own institution as well as across institutions. For more information about Shibboleth, please see <http://shibboleth.internet2.edu>

CONTENT CREATION AND MODIFICATION

Vital can store any type of content, and supports the creation and editing of XML-based metadata right in the web interface. You can also store

Notable functionality includes:

Support for any content type in its native format - When you import image, text, sound, video and other digital files, basic metadata and XML content is created dynamically based on MIME-Type. Upon importation of text-based objects encoded in Dublin Core (DC), EAD or MARCXML, Vital will automatically extract content data and use it to catalog the object. Vital can convert EAD and MARCXML metadata into DC metadata for the purposes of cataloging and indexing a digital object. Upon import of generic objects, Vital will automatically catalog them by creating Dublin Core metadata for Title, Format, Identifier, and Source, using customizable XSL stylesheets.

Relationships Management – Vital provides support for the assignment of relationships between repository objects. Management tools allow for like content to be grouped as an institution desires (by academic department, content type, research unit, etc.) and provide end-users with the ability to search and browse these collections. Vital supports unlimited levels of collection 'nesting' (collections within other collections).

Previewing content directly from Vital – Vital provides staff the ability to preview some content formats directly from the repository. Large thumbnail displays of image content, such as JPEG2000, BMP and GIF are available. Other content types with supported previews include PDF documents and select video formats.

Viewing, Editing and Saving content directly from Vital – Since Vital has no limitations on the content formats that can be added, facilities are available for integration with third-party client applications for viewing/editing content in its native format and to save changes directly to the Vital repository.

Editing XML content in your preferred interface – The Vital Manager allows for integration of commercial XML editors such as Xmetal, XMLSpy, Authentic and InfoPath so that users can edit XML content directly from the repository with their editor of choice.

QuickEdit XML – Vital provides administrators the ability to edit existing XML Metadata – without having to know XML. The

QuickEdit XML Editor allows administrators to edit the XML datastream in its native format, or to edit using an HTML Form.

Tracking content changes through versioning – Vital keeps track of changes to digital objects and associated datastreams, records metadata related to the modification of the object/datastream, and preserves each version for later retrieval. Additionally, Vital supports version roll-back so that older versions can be restored.

Automated capture of technical metadata for preservation purposes - Vital can automatically capture still-image metadata according to the MIX standard (XML expression of Z39.87 NISO Standard for Technical Metadata from Digital Still Images). This functionality is enabled as a Vital web service to automatically capture technical metadata from popular image file formats such as TIFF, JPEG, GIF and others. This allows Vital to handle and populate object metadata directly from the source image to further improve search results and to preserve information about the creation of the file.

Automated text capture for full-text searching – The system provides advanced metadata capture for PDF, DOC, RTF and XHTML. To provide full-text searching capabilities against the content of an imported text document, Vital accommodates the metadata transformation and full text extraction and indexing from PDF (Portable Document Format), DOC (MS-Word), RTF (Rich Text Format) and XHTML (eXtensible Hyper Text Markup Language) document types, in addition to the native XML indexing functionality.

Automatic validation of content via JHOVE – Upon ingest into the repository, Vital will automatically validate specific objects using JHOVE (JSTOR/Harvard Object Validation Environment). This feature allows administrators to trust that the content being added to the repository can be validated as a known format type, that it is not masquerading as another format type, and that metadata configured for that format type can be extracted by the Vital system to be used for preservation purposes, for future version migrations and for advanced resource discovery.

Support for PREMIS preservation metadata – Upon ingest into the repository, Vital creates a PREMIS metadata file for each datastream and uses PREMIS to store an SHA-1 hash checksum for the datastream, creating and preserving a unique alphanumeric key that can be used in the future to ensure the integrity of the content over time. The checksum is regenerated and the PREMIS metadata updated with a new value whenever the datastream content is edited and saved. Vital also supports the ability to ingest and store PREMIS metadata the institution may have created in advance as a way to preserve information relevant to the creation of the datastream. Vital also allows for PREMIS metadata fields to be set as XPATH definitions to be used for searching. These fields can then be used as targets for administrative reports.

XML Validation – To ensure the integrity of XML metadata added directly to the repository, Vital checks to ensure that the data is not malformed. When an error is detected, the system will warn the user of the problem and will provide information pertaining to the source of the problem within the file.

Metadata Synchronization – Vital objects contain a Dublin Core descriptive metadata datastream for every item that can

be used for indexing, searching, harvesting, and as source metadata for gathering citations. Since Vital also supports other XML formatted metadata for object description, such as MARCXML, Vital automatically updates the Dublin Core metadata whenever changes are made to the source descriptive metadata file. Additionally, Vital regenerates other appropriate 'derived' system data, such as JHOVE, PREMIS, full-text, thumbnails, etc., whenever modification to the source content is made.

VITAL BATCH INGEST UTILITY

The Vital Batch Ingest Utility allows system administrators to ingest larger quantities of content into the Vital repository via command-line tools and XML-based configuration files. The toolset has been designed to allow for considerable flexibility in configuring the system to ingest content en masse. The system allows for simple or compound object creation/ingest of any content type, and includes the same robust functionality as other import methods, including PDF text extraction, Handles assignment and technical metadata extraction.

Notable functionality includes:

Flexible definition of ingest source targets – The Vital Batch Ingest Utility allows administrators the ability to choose from a variety of import target types, including a single file, a directory of files, a single XML file with multiple records, a MARC ISO2709 file of multiple records, or a URL for harvesting content from a remote location.

File extension and filename filtering – Administrators can choose to define a target directory but limit the batch ingest to target files with a specific format extension (e.g. PDF files only) or filename token (e.g. 96.pdf, 96a.pdf, 96.rtf, etc.).

Ingest content interpreted/derived from a source target – Administrators have the flexibility to define a 'data token' extracted from a target XML file to discover other files for import. To illustrate, a user could define a particular datafield tag from a MARCXML file as the defined data token to discover referenced files to be ingested as separate datastreams.

Ingest output from executable program – Administrators have the ability to define the output of an executed program to be imported into the repository as separate datastreams. To illustrate, a user could choose to have the JPEG2000 derivative images from a TIFF-to-JPEG2000 conversion program ingested

into the repository as separate datastreams, or to migrate content of varying versions to a single output version and ingest these files en masse.

Metadata mapping – The Vital Batch Utility comes packaged with stylesheets to map various fields within metadata in its native format into DC metadata. These stylesheets are customizable to allow for robust cataloging.

VALET ELECTRONIC SUBMISSION TOOL

Valet is a submission interface that allows users to submit content into the repository through a web-based interface. The interface can handle submission of any content type, and allows contributors and other content creators to enter metadata into configurable form-based templates. Valet can handle a staged submission process where any number of review stages can be integrated to ensure that staff have the ability to edit, delete, or approve submitted content prior to ingestion into the repository. Additional features such as form validation, configurable submission workflows, user authentication and facilities for reviewer comments make this tool a critical asset for distributed content submission.

Notable functionality includes:

Pre-configured templates for electronic theses and dissertations – Innovative has enhanced the functionality present in the Electronic Theses & Dissertations (ETD – created by Virginia Tech) and Australian Digital Theses Program (ADT – created by the Council of Australian University Librarians) solutions for submission of electronic theses and dissertations to create a 'best of breed' solution for submission of these content types.

Pre-configured templates for various content types – In addition to support for ETDs, VALET ships with pre-configured templates for journal articles, conference papers, working papers, books and book chapters. While pre-configured, these templates may still be modified to meet your institution's exact needs for these content types.

Configurable templates and workflows – Administrators can configure the system to allow for the required and optional fields that meet their institution's needs for web-submitted content. Additionally, the system allows for administrators to define a specific workflow for submission, which could include number of metadata screens, role-based stages for content review, or required authorization screens.

Configurable, staged content aggregation – Valet supports the ability for administrators to configure required stages for content review, approval/rejection, additional metadata entry for cataloging requirements or reviewer comments, send e-mail to contributing author/creator or other purposes.

Automatic metadata transformation for any XML schema – Valet provides administrators with facilities to define a transforma-

tion, allowing data entered by content authors to be converted into any XML schema for inclusion into the repository object (e.g. Dublin Core, MARCXML, ETDMS, etc.).

Automatic assignment of Handles persistent identifiers for content – Used in conjunction with Vital, Valet allows for the automatic assignment of a Handles persistent identifier to the object and its associated datastreams.

Automatically extract full-text for PDF content – Used in conjunction with Vital, Valet allows for the automatic extraction of data from PDF files and other text-based file formats to create a full-text datastream within the repository to enhance resource discovery from the Vital Access Portal.

Pop-up Help windows – Valet provides facilities for administrators to configure pop-up help windows to guide content creators through the submission process.

ABOUT FEDORA™

Fedora™ was selected for Vital's repository because of its support of Service-Oriented Architecture and because of its scalability over the long-term. The following is some background information on Fedora:

In September 2001 the Mellon Foundation funded a project to build a digital object repository management system. The project was based on and became known as the Flexible Extensible Digital Object Repository Architecture (Fedora™). Jointly developed by the University of Virginia and Cornell University, Fedora™ is designed to be a foundation upon which interoperable web-based digital libraries, institutional repositories and other information management systems can be built.

At its core is a powerful digital object model that supports multiple views of each digital object and the relationships among digital objects. Digital objects can encapsulate locally-managed content or make reference to remote content. Dynamic views are possible by associating web services with objects. Digital objects exist within a repository architecture that supports a variety of management functions. All functions of Fedora, both at the object and repository level, are exposed as web services. These functions can be protected with fine-grained access control policies.

This unique combination of features makes Fedora an attractive solution in a variety of domains. Some examples of applications that are built upon Fedora include library collections management, multimedia authoring systems, archival repositories, institutional repositories, and digital libraries for education.

A fundamental requirement of an open architecture for digital libraries is a reliable and secure means to store and access digital content. Fedora™ is a digital object and repository architecture designed to achieve these requirements, while at the

same time providing extensibility and interoperability. The key features of the architecture are: support for heterogeneous data types; accommodation of new types as they emerge; aggregation of mixed, possibly distributed data into complex objects; the ability to specify multiple content disseminations of these objects; and the ability to associate rights management schemes with these disseminations. [Sandra Payette and Carl Lagoze Fedora™ ECDL98 Paper]

■ TECHNICAL INFORMATION

System Architecture

The Vital Access Portal web component serves as the public online interface for searching the objects within the Vital repository, and can be extensively customized to adapt the software for your own use. The Fedora™ repository includes a basic relational database (PostgreSQL) as part of the repository, but you have the option of using other RDBMS's such as MySQL or Oracle.

[The following section describing the Fedora™ architecture is from an article in D-Lib Magazine – April 2003 by Thornton Staples, Ross Wayland and Sandra Payette – An Open-source Digital Object Repository Management System]

The Fedora™ architecture is based on object models that by definition are templates for units of content, called data objects, which can include digital resources, metadata about the resources, and linkages to software tools and services that have been configured to deliver the content in desired ways. These software connections are provided as methods encoded into two kinds of inter-related behavior objects as described below. A Fedora™ repository provides access to the data objects by leveraging tools and services that are described by the behavior objects. The behavior objects store metadata that describes the operations of the tool/service and the runtime bindings for running the operations. The Web Services Description Language (WSDL) is used to describe the tool/service bindings.

The digital resources and the metadata are datastreams in an object model, definitions of which connect the content model either to internal content under the direct control of the repository or to external content that is delivered via HTTP servers. The content of a datastream is identified using a URL. When an object is ingested into a Fedora™ repository, a URL for a managed datastream is used by the repository system to retrieve the content and store it in the file space under its control; the datastream in the object is updated to be this internal address. When an object contains a datastream defined as external, the URL is stored in the datastream and used by the repository to access the data whenever necessary. An in-line metadata datastream is a bytestream that is name-spaced XML encoded data stored in the XML instantiation of the object directly, rather than as remote or managed content.

From the user's point of view, the linkages to software tools and services (via disseminators) are seen as behaviors upon the units of content. These behaviors can be exploited to deliver varieties of prepared content directly to a web browser. They can also be used to prepare or configure content to be used through some external software application. In a sense, these object models can be thought of as containers that give a useful shape to information poured into them; if the information fits the container, it can immediately be used in predefined ways.

Fedora™ makes it possible to describe abstract sets of behaviors that constrain a corresponding set of specific processes or mechanisms delivering the behavior described for a given unit of content. One abstract set of behaviors, a behavior definition (bdef) object, can be used to constrain many mechanism sets, or behavior mechanism (bmech) objects, ensuring a standardization of behaviors for different units of content that are equivalent in type, but differing in format. A bdef object formally defines the terms of a behavior contract that must be upheld by any bmech object to be paired with it. In turn, the bmech object contains a data contract, the terms of which any data object model subscribing to it must meet. Bdef objects and bmech objects are analogous to interfaces and implementations in object-oriented programming.

A data object model subscribes to a set of behaviors by linking to a bdef object and pairing it with a link to an appropriate bmech object. This pair of links defines a disseminator; an object model can contain any number of disseminators. In practical terms, this means a specific data object conforming to the model can have sets of behaviors for a variety of purposes, or sets of behaviors equivalent in purpose but that prepare the object's content to be delivered to applications with different format requirements. In summary, a data object model specifies the number and types of datastreams as well as the set of disseminators every conforming data object will have.

■ FOR MORE INFORMATION :

© Innovative Interfaces, Inc.
1900 Powell St., Suite 400
Emeryville, CA 94608

web: www.iii.com
email: info@iii.com
call: 800.878.6600



twitter.com/iii_Innovative



facebook.com/innovativeinterfaces