RODA

Repository of Authentic Digital Objects

Open Repositories 09
Atlanta, Georgia, USA
2009-05-20
First I will explain what is RODA, then it’s relation to Fedora Commons. Later we will enter in RODA design, based in OAIS, architecture and data model. Then we will see how RODA implements the OAIS 3 meta-processes: Ingest, Access and Administration.
What is RODA?

A digital repository specially designed for archives, with the following main features:

- Long-term Preservation and Authenticity
- Based on standards (OAIS, METS, EAD, PREMIS, etc.)
- TRAC compliant (Trustworthy Repositories Audit & Certification)
- Secure
- Scalable (SOA)
- Clean web user interface and ingest desktop tool
- For archivists, for producers, for consumers
- Open source

RODA is a digital repository created by a National Archive, specially designed archives. It primary objective is to allow long-term preservation and authenticity. For that it was based as much as possible on standards, follow international specifications to become a trustworthy repository, as the TRAC, and have high standards on security and scalability. User interface was also a important issue, as it was designed for archivists, for producers and for consumers. And it is open source!
The project team is divided by the Portuguese National Archives, with archiving consulting and development, the University of Minho, with software engineering consulting, the Assymetric Studios with design, the IDW with hardware, and Keep Solutions with maintenance and support.
In the beginning of the project, many platforms were analysed, and two came forth as the strongest, DSpace and Fedora Commons. But, even if DSpace has many features already implemented, it was not flexible enough to implement all needed premisses (like EAD support). Throughout the implementation of the project, Fedora Commons has shown that it was the best solution by far.
In the beginning of the project, many platforms were analysed, and two came forth as the strongest, DSpace and Fedora Commons. But, even if DSpace has many features already implemented, it was not flexible enough to implement all needed premisses (like EAD support). Throughout the implementation of the project, Fedora Commons has shown that it was the best solution by far.
To begin with the presentation of the RODA project, we will start with the base design that all repositories must follow: the OAIS. Many of you must already know this model very well, so I won’t take much time with it. We can identify 3 actors: producers, management and consumers. The producer sends a SIP to the repository, this SIP is ingested and becomes an AIP. Then the management uses the administration to preserve the AIP throughout the years. The consumer would search though the metadata to find the object it wants and retrieves a DIP.
RODA has a Service Oriented Architecture. Fedora serves has the base platform that accommodates the data model. The RODA Core Services implements all logic and is accessible by a programatic interface (Web Service) to any outside client. Open LDAP supports all authentication and user management back end. Migration services and preservation actions server allows for a distributed load and pluggable interface to heavy migration and preservation actions. The RODA Web User Interface gives an easy and clean interface to the end user, to all features implemented in Core Services.
This is the data model implemented with Fedora RDF relationship mechanism. It’s simple, atomistic, almost molecular. The red nodes are the description objects, that relate to each other creating the classification plan hierarchy, as in EAD. Each object is composed by an EAD Component, a node in the EAD hierarchy.

The green nodes are representation objects that have the real data (as opposed to the meta data). They can be composed by various files, but must define an entry point file.

The blue nodes are preservation objects for representations. They have the preservation metadata for each representation and files, and include technical metadata about each file.

The yellow nodes are preservation objects for events, that are created by preservation tasks like fixity checks or migrations.

The grey nodes are preservation objects for agents, that have metadata about the programs or users that performed an event.
This is the data model implemented with Fedora RDF relationship mechanism. It’s simple, atomistic, almost molecular. The red nodes are the description objects, that relate to each other creating the classification plan hierarchy, as in EAD. Each object is composed by an EAD Component, a node in the EAD hierarchy.

The green nodes are representation objects that have the real data (as opposed to the meta data). They can be composed by various files, but must define an entry point file.

The blue nodes are preservation objects for representations. They have the preservation metadata for each representation and files, and include technical metadata about each file.

The yellow nodes are preservation objects for events, that are created by preservation tasks like fixity checks or migrations.

The grey nodes are preservation objects for agents, that have metadata about the programs or users that performed an event.
This is the data model implemented with Fedora RDF relationship mechanism. It’s simple, atomistic, almost molecular. The red nodes are the description objects, that relate to each other creating the classification plan hierarchy, as in EAD. Each object is composed by an EAD Component, a node in the EAD hierarchy. The green nodes are representation objects that have the real data (as opposed to the meta data). They can be composed by various files, but must define an entry point file. The blue nodes are preservation objects for representations. They have the preservation metadata for each representation and files, and include technical metadata about each file. The yellow nodes are preservation objects for events, that are created by preservation tasks like fixity checks or migrations. The grey nodes are preservation objects for agents, that have metadata about the programs or users that performed an event.
This is the data model implemented with Fedora RDF relationship mechanism. It’s simple, atomistic, almost molecular. The red nodes are the description objects, that relate to each other creating the classification plan hierarchy, as in EAD. Each object is composed by an EAD Component, a node in the EAD hierarchy.

The green nodes are representation objects that have the real data (as opposed to the meta data). They can be composed by various files, but must define an entry point file.

The blue nodes are preservation objects for representations. They have the preservation metadata for each representation and files, and include technical metadata about each file.

The yellow nodes are preservation objects for events, that are created by preservation tasks like fixity checks or migrations.

The grey nodes are preservation objects for agents, that have metadata about the programs or users that performed an event.
This is the data model implemented with Fedora RDF relationship mechanism. It’s simple, atomistic, almost molecular. The red nodes are the description objects, that relate to each other creating the classification plan hierarchy, as in EAD. Each object is composed by an EAD Component, a node in the EAD hierarchy.

The green nodes are representation objects that have the real data (as opposed to the meta data). They can be composed by various files, but must define an entry point file.

The blue nodes are preservation objets for representations. They have the preservation metadata for each representation and files, and include technical metadata about each file.

The yellow nodes are preservation objects for events, that are created by preservation tasks like fixity checks or migrations.

The grey nodes are preservation objects for agents, that have metadata about the programs or users that performed an event.
This is the data model implemented with Fedora RDF relationship mechanism. It’s simple, atomistic, almost molecular. The red nodes are the description objects, that relate to each other creating the classification plan hierarchy, as in EAD. Each object is composed by an EAD Component, a node in the EAD hierarchy.

The green nodes are representation objects that have the real data (as opposed to the meta data). They can be composed by various files, but must define an entry point file.

The blue nodes are preservation objects for representations. They have the preservation metadata for each representation and files, and include technical metadata about each file.

The yellow nodes are preservation objects for events, that are created by preservation tasks like fixity checks or migrations.

The grey nodes are preservation objects for agents, that have metadata about the programs or users that performed an event.
First we will talk how RODA implements the ingest meta-process.
One of the premisses of preservation is to define a small group of formats that can represent each class of objects, defined as preservation formats. In RODA, the defined formats are PDF/A for text documents, TIFF for raster images, MPEG–2 for video, WAV for audio, and DBML for relational databases. DBML is an XML schema devised by the team for this purpose. RODA can ingest any one of this types, and with the normalization plugins already created, can ingest many other formats. Furthermore, plugins can easily be created to allow ingestion of other formats not in the list.
One of the premisses of preservation is to define a small group of formats that can represent each class of objects, defined as preservation formats. In RODA, the defined formats are PDF/A for text documents, TIFF for raster images, MPEG-2 for video, WAV for audio, and DBML for relational databases. DBML is an XML schema devised by the team for this purpose. RODA can ingest any one of these types, and with the normalization plugins already created, can ingest many other formats. Furthermore, plugins can easily be created to allow ingestion of other formats not in the list.
A cross-platform desktop application allows the producers to create, describe, manage and upload objects to the RODA server. The classification plan can be downloaded from the server. Then the producer can create a SIP with a file or item inside of it. A panel allows for the insertion of the descriptive metadata and its validation. Afterwards a representation can be created, choosing the object class. For each object class there is a different creator panel. This is the creator for raster images. The order and structure of the images can be defined and a preview and metadata are presented. After the object is created and described, it can be sent to the RODA server: online by HTTP or FTP, or by postal mail.
A cross-platform desktop application allows the producers to create, describe, manage and upload objects to the RODA server. The classification plan can be downloaded from the server. Then the producer can create a SIP with a file or item inside of it. A panel allows for the insertion of the descriptive metadata and its validation. Afterwords a representation can be created, choosing the object class. For each object class there is a different creator panel. This is the creator for raster images. The order and structure of the images can be defined and a preview and metadata are presented. After the object is created and described, it can be sent to the RODA server: online by HTTP or FTP, or by postal mail.
A cross-platform desktop application allows the producers to create, describe, manage and upload objects to the RODA server. The classification plan can be downloaded from the server. Then the producer can create a SIP with a file or item inside of it. A panel allows for the insertion of the descriptive metadata and its validation. Afterwards a representation can be created, choosing the object class. For each object class there is a different creator panel. This is the creator for raster images. The order and structure of the images can be defined and a preview and metadata are presented. After the object is created and described, it can be sent to the RODA server: online by HTTP or FTP, or by postal mail.
A cross-platform desktop application allows the producers to create, describe, manage and upload objects to the RODA server. The classification plan can be downloaded from the server. Then the producer can create a SIP with a file or item inside of it. A panel allows for the insertion of the descriptive metadata and its validation. Afterwords a representation can be created, choosing the object class. For each object class there is a different creator panel. This is the creator for raster images. The order and structure of the images can be defined and a preview and metadata are presented. After the object is created and described, it can be sent to the RODA server: online by HTTP or FTP, or by postal mail.
A cross-platform desktop application allows the producers to create, describe, manage and upload objects to the RODA server. The classification plan can be downloaded from the server. Then the producer can create a SIP with a file or item inside of it. A panel allows for the insertion of the descriptive metadata and its validation. Afterwards a representation can be created, choosing the object class. For each object class there is a different creator panel. This is the creator for raster images. The order and structure of the images can be defined and a preview and metadata are presented. After the object is created and described, it can be sent to the RODA server: online by HTTP or FTP, or by postal mail.
A cross-platform desktop application allows the producers to create, describe, manage and upload objects to the RODA server. The classification plan can be downloaded from the server. Then the producer can create a SIP with a file or item inside of it. A panel allows for the insertion of the descriptive metadata and its validation. Afterwards a representation can be created, choosing the object class. For each object class there is a different creator panel. This is the creator for raster images. The order and structure of the images can be defined and a preview and metadata are presented. After the object is created and described, it can be sent to the RODA server: online by HTTP or FTP, or by postal mail.
SIP structure

- Object class and format
- Place in classification plan
- Descriptive metadata
- Preservation metadata
- Technical metadata
- Representation files
- Identification of root file

To send an object to RODA is used a Submission Information Package. This package defines the object class and format, its place in the classification plan, the metadata and files, all inside a zip with this structure.
Ingest workflow

<table>
<thead>
<tr>
<th>Step</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Upload</td>
<td>Upload SIP to the ingest stage area by HTTP, FTP or local copy</td>
</tr>
<tr>
<td>2</td>
<td>Unzip</td>
<td>Open ZIP package</td>
</tr>
<tr>
<td>3</td>
<td>Virus Check</td>
<td>Check all files for viruses with Clam Anti Virus</td>
</tr>
<tr>
<td>4</td>
<td>Syntax and fixity check</td>
<td>Check if metadata is well formed and integrity of all files against checksums in METS envelope</td>
</tr>
<tr>
<td>5</td>
<td>Authorization check</td>
<td>Check if user has authorization to ingest into the defined classification plan</td>
</tr>
<tr>
<td>6</td>
<td>Ingest</td>
<td>Ingest original into Fedora. This ingestion is not final, can be undone in next steps.</td>
</tr>
<tr>
<td>7</td>
<td>Normalization</td>
<td>Check if representation needs normalization, and enforce it as needed</td>
</tr>
<tr>
<td>8</td>
<td>Semantic check</td>
<td>Manual semantic check by an archivist, then object is marked active, indexed and published.</td>
</tr>
</tbody>
</table>

When the SIP arrives RODA, it goes throw an ingest workflow that checks if everything is correct. The SIP is check for virus, many validation checks are performed, the authorization is checked and the representations are normalized if needed. Finally, an archivist checks the semantic and approves or rejects the SIP. The producer is automatically warned, if selected.
This is the RODA Web User Interface, showing the ingest status menu option. This is the list of sent SIPs, describing each SIP name, submission date, current ingest state and the producer. Highlighted in green there is the SIP we have just sent.
This is the RODA Web User Interface, showing the ingest status menu option. This is the list of sent SIPs, describing each SIP name, submission date, current ingest state and the producer. Highlighted in green there is the SIP we have just sent.
This is the RODA Web User Interface, showing the ingest status menu option. This is the list of sent SIPs, describing each SIP name, submission date, current ingest state and the producer. Highlighted in green there is the SIP we have just sent.

<table>
<thead>
<tr>
<th>File</th>
<th>Submission date</th>
<th>Current state</th>
<th>%</th>
<th>Producer</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST_sf1_c1_sr1Ngu.sip</td>
<td>2009-05-11 14:41:47</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_sf1_c1_sr1_fox.sip</td>
<td>2009-05-07 12:19:36</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_MP48298.sip</td>
<td>2009-03-20 17:50:39</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_PNG2.sip</td>
<td>2009-03-20 15:06:14</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_PNG492.sip</td>
<td>2009-03-20 15:05:28</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_XPM.sip</td>
<td>2009-03-20 11:40:25</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_WAV.sip</td>
<td>2009-03-20 11:40:24</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_TXT.sip</td>
<td>2009-03-20 11:40:16</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_TIFF.sip</td>
<td>2009-03-20 11:40:15</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_TGA.sip</td>
<td>2009-03-20 11:40:15</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_RTF.sip</td>
<td>2009-03-20 11:40:14</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_PDF.sip</td>
<td>2009-03-20 11:40:13</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_OGG.sip</td>
<td>2009-03-20 11:40:06</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_ODT.sip</td>
<td>2009-03-20 11:40:05</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_MP3.sip</td>
<td>2009-03-20 11:40:01</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_JPEG.sip</td>
<td>2009-03-20 11:39:28</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_ICO.sip</td>
<td>2009-03-20 11:39:25</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_GIF.sip</td>
<td>2009-03-20 11:39:24</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
<tr>
<td>TEST_LFARIA_FLAC.sip</td>
<td>2009-03-20 11:39:22</td>
<td>accepted</td>
<td>100%</td>
<td>Ifaria</td>
</tr>
</tbody>
</table>
If you click the report icon, you can see the details of the outcome of each ingest workflow task.
If you click the report icon, you can see the details of the outcome of each ingest workflow task.
Next we will talk about the Access meta-process.
Access

The ability to find and retrieve objects in a format that is easily and widely accessible by the technologically ever changing target community.

- Basic and advanced search
- Tree-driven browsing of the classification plan
- View and download:
  - descriptive metadata
  - preservation metadata
  - digital representations (actual data)
- Multiple viewers per content type

As the repository content grows, any consumer can access the information easily. For the brave ones, one can browse the classification plan tree. For the impatient ones, a powerful search mechanism based on Lucene can rapidly find what you are looking for in a glimpse of an eye.

In every object, the descriptive and preservation metadata can be viewed and easily downloaded. Also the original and normalized versions of the representation can be downloaded, or just use one of the previews to rapidly see the content.
A simple search bar enables the user to find anything in the repository. In this example the “or09” keyword was search, and it found it in the scope and content field. There it is, highlighted. The results show the score of the search, and a summary of each indexed fields, highlighting the keywords.
A simple search bar enables the user to find anything in the repository. In this example the “or09” keyword was search, and it found it in the scope and content field. There it is, highlighted. The results show the score of the search, and a summary of each indexed fields, highlighting the keywords.
But if the basic search is not enough, there is an advanced search where the user can specify the description level, date range, and fields to search in. Clicking in the consult button, the object is shown in the browser.
But if the basic search is not enough, there is a advanced search where the user can specify the description level, date range, and fields to search in. Clicking in the consult button, the object is shown in the browser.
But if the basic search is not enough, there is an advanced search where the user can specify the description level, date range, and fields to search in. Clicking in the consult button, the object is shown in the browser.
But if the basic search is not enough, there is a advanced search where the user can specify the description level, date range, and fields to search in. Clicking in the consult button, the object is shown in the browser.
The classification plan is expanded to focus in the object. All descriptive metadata is shown. Clicking in the Visualization we jump the the Viewers tab.
The classification plan is expanded to focus in the object. All descriptive metadata is shown. Clicking in the Visualization we jump the the Viewers tab.
The classification plan is expanded to focus in the object. All descriptive metadata is shown. Clicking in the Visualization we jump the the Viewers tab.
The original and normalized representation are available for download. But, to quickly check the content, choose one of the beautiful previewers. Let's check the photo previewer.
The original and normalized representation are available for download. But, to quickly check the content, choose one of the beautiful previewers. Let’s check the photo previewer.
All the images are shown in a flash previewer, where the images added in the SIP.
In the browser, one can also see all the preservation metadata, laid down in a dynamic timeline viewer.
In the browser, one can also see all the preservation metadata, laid down in a dynamic timeline viewer.
Finally, we will talk about the administration meta-process.
Preservation

• Format migration
• Fixity check (MD5, SHA)
• Backups, RAID
• Automatic generation of preservation metadata for every action taken (PREMIS events)

• Succession plan
• Trustworthy repository

RODA uses the migration preservation strategy. It also keeps the integrity by a frequent fixity check, backups, and RAID. Every action taken automatically generates preservation metadata.

On the Organizational level, there are also some needed premisses for long-term preservation. The existence of a succession plan, and all needed attributes for RODA to be a trustworthy repository.
Scheduler

*Scheduling mechanism to run specific actions on the repository*

- Actions are developed as plugins
  - Easy to install
  - Easy to develop
  - Easy to maintain

- Many plugins already created
  - Fixity check
  - Virus check
  - Normalization plugins
  - Migration plugins
  - Alerts
  - etc.

The plugin schedule mechanism allows new migrators to be easily added to the system. Install a new plugin is just to copy a jar to a particular folder. To create a new plugin one just has to implement a simple interface, and add a line in the jar manifest. Many plugins are already created, like ...
Here are the list of scheduled tasks. To create a new one just click new.
Here are the list of scheduled tasks. To create a new one just click new.
Here are the list of scheduled tasks. To create a new one just click new.

<table>
<thead>
<tr>
<th>Name</th>
<th>Start date</th>
<th>Repeat</th>
<th>Username</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estatísticas</td>
<td>2009-01-01 03:00:00</td>
<td>repeat every 1 days para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Ingest/Authorization</td>
<td>2009-03-11 16:46:49</td>
<td>repeat every 5 minutes para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Ingest/Check SIP syntax</td>
<td>2009-03-11 16:12:48</td>
<td>repeat every 5 minutes para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Ingest/Create AIP</td>
<td>2009-03-20 20:12:06</td>
<td>repeat every 5 minutes para sempre</td>
<td>rcastro</td>
</tr>
<tr>
<td>Ingest/Monitor FTP</td>
<td>2009-03-11 16:13:35</td>
<td>repeat every 5 minutes para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Ingest/Normalize</td>
<td>2009-03-20 20:12:34</td>
<td>repeat every 5 minutes para sempre</td>
<td>rcastro</td>
</tr>
<tr>
<td>Ingest/Unpack</td>
<td>2009-03-11 16:39:00</td>
<td>repeat every 5 minutes para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Ingest/Unpack FTP</td>
<td>2009-03-11 16:38:38</td>
<td>repeat every 5 minutes para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Ingest/Virus</td>
<td>2009-03-11 16:39:20</td>
<td>repeat every 5 minutes para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Notificar produtores</td>
<td>2009-04-06 10:47:00</td>
<td>repeat every 1 days para sempre</td>
<td>mferreira</td>
</tr>
<tr>
<td>Preservation/Fixity</td>
<td>2009-03-11 04:00:00</td>
<td>repeat every 7 days para sempre</td>
<td>Ifaria</td>
</tr>
<tr>
<td>Reject 'AUTHORIZED' SIP</td>
<td>2009-04-29 03:58:35</td>
<td>repeat every 5 minutes para sempre</td>
<td>rcastro</td>
</tr>
<tr>
<td>Reject 'DROPPED_FTP' SIP</td>
<td>2009-04-29 03:59:00</td>
<td>repeat every 5 minutes para sempre</td>
<td>rcastro</td>
</tr>
<tr>
<td>Reject 'SIP_INGESTED' S</td>
<td>2009-04-29 04:00:09</td>
<td>repeat every 5 minutes para sempre</td>
<td>rcastro</td>
</tr>
<tr>
<td>Reject 'SIP_NORMALIZED'</td>
<td>2009-04-29 04:01:02</td>
<td>repeat every 5 minutes para sempre</td>
<td>rcastro</td>
</tr>
<tr>
<td>Reject 'SIP_VALID' SIPs</td>
<td>2009-04-29 04:02:02</td>
<td>repeat every 5 minutes para sempre</td>
<td>rcastro</td>
</tr>
</tbody>
</table>
Each task created needs a name and description. The start date can be scheduled for some date and time. And the plugin can be set to repeat. The list of installed plugins is presented, there are several already installed. Let's choose an ingest tool, Notify producers.
Each plugin can set a different set of parameters. Let's create a producers notification that will be sent once a month with a list of the SIP accepted or rejected in that period of time.
Security

- Advanced user management with LDAP
  - Groups, nested groups, users and permissions

- Fine-grain permissions
  - from repository to object level

- Transparent permission integration in browse and search

- ALL user actions are logged

- ALL plugin actions are logged

- Detailed statistics and reports

- Anonymous users are not allowed
  - For security and accountability reasons, all user actions are logged

A trusted repository must have high standards in security and system management. The user management supported by LDAP allows definition of users and groups, each with a set of permissions (or roles) that are recursively inherited. Above this repository wide permissions, fine-grain permission can be added to each object. The change of permissions integrated seamlessly in the browsing and searching of items. If one does not have permissions to an object, its just like it doesn’t exist. Furthermore, very detailed logs are kept about all actions the user does, and, has anonymous user is not allowed, that means everyone. All actions executed by plugins are also logged, because they are treated just like users. Nevertheless, refined statistics and reports about every aspect of the repository are shown and an history of them kept.
This is the list of users, with the groups the users belong to nested below. Here the users and group can be created, edited, activated and removed. Lets edit my user.
This is the list of users, with the groups the users belong to nested below. Here the users and group can be created, edited, activated and removed. Lets edit my user.
Many personal fields are available. The groups this user belongs to can be edited in the bottom boxes. In the other tab we have the permissions.
Many personal fields are available. The groups this user belongs to can be edited in the bottom boxes. In the other tab we have the permissions.
This are the repository level permissions. As I am an administrator, I can do everything, but every group of users, like archivists, producers and consumers have a different set of permissions, in accord with their role in the repository.
Fine grain permissions can be set into the object level directly in the browser. The permissions can be No Access, Access to metadata and disseminations, access and metadata edition, or full control, as control to change permissions. Permissions can be recursively set to all sub-levels.
On each fonds, or collection, the list of authorized producers can be set.
Every single action made by a user is logged with detail, including parameters. The log can be listed and filtered by action, date range or user. Here you can see all the actions I was doing to create this screenshots.
Also, every time a plugin runs a report is created and the result is kept in the scheduler history.
Furthermore, very detailed statistics are available with lots of specific information about the repository. The statistics are divided into repository (descriptive, preservation and representation objects, representation class and format distributions), ingest (SIP state distribution, mean processing time), producers (SIP state per producer), users (user and groups size), access (page views, metadata views, representation views, disseminators views distribution, etc.) actions (metadata edition distributed by users, user edited, task edited) system (CPU time, Memory usage, Disk usage, etc.)
Furthermore, very detailed statistics are available with lots of specific information about the repository. The statistics are divided into repository (descriptive, preservation and representation objects, representation class and format distributions), ingest (SIP state distribution, mean processing time), producers (SIP state per producer), users (user and groups size), access (page views, metadata views, representation views, disseminators views distribution, etc.) actions (metadata edition distributed by users, user edited, task edited) system (CPU time, Memory usage, Disk usage, etc.)
The history of each statistic item is kept and viewed as a graphic or a table.
Conclusion

- Centralized digital repository held at the National Archives
- Its main audience is
  - Public Administration institutions (as producers)
  - General public (as consumers)
- Based on Fedora Commons
- State-of-the-art technologies
  - Fedora Commons, Java, MySQL, LDAP, GWT, etc.
- On the final testing phase, soon going into production (June 2009)
- Open source

- Looking for partners and developers to keep the wheel spinning...

All this is installed and being tested in the Portuguese National Archives, all will be public until June 2009. The main audience is the public administration as producers and everyone as consumers. The repository is based in Fedora Commons and uses state-of-the-art, free, technologies. RODA is Open Source, and we are currently looking for partners and more developers to work in RODA.
Future work

- Conclude the internationalization of the repository GUI
- Support for SWORD ingest protocol
- Support more digital formats
- Development of more plugins
- Documentation in english

The future work includes finishing the internationalization of the user interface, support for SWORD ingest protocol, support ingest of more formats and more plugins. We also need to increase the available documentation in english.
Questions?

National Archives  http://www.dgarq.gov.pt
   Luís Faria  luis.faria@dgarq.gov.pt
   Rui Castro  rui.castro@dgarq.gov.pt
   Francisco Barbedo  francisco.barbedo@dgarq.gov.pt
   Cecília Henriques  cecilia.henriques@dgarq.gov.pt
   Luís Corujo  luis.corujo@dgarq.gov.pt

University of Minho  http://www.uminho.pt
   Miguel Ferreira  mferreira@dsi.uminho.pt
   José Carlos Ramalho  jcr@di.uminho.pt

http://portal.roda.dgarq.gov.pt

http://roda.dgarq.gov.pt

So here are our contacts, we can check the more information in the portal, or send me an email. Any Questions?
References


