



atira**information**

PURE application overview

Created by: Atira A/S

Date: April 22, 2008

Version: 1.1

Contents

1. Introduction	3
2. Product overview	4
2.1.1. Data modeling	4
2.1.2. Replaceable meta-data model	5
2.1.3. Content families	5
2.1.4. Many-to-many relations	5
2.1.5. Relations objects	5
2.1.6. Classification editor	6
2.2. Data ingestion	6
2.2.1. Graphical user interface	7
2.2.1.1. Roles and rights	7
2.2.1.2. Customizable	8
2.2.2. Workflows	8
2.2.2.1. Example	8
2.2.2.2. Customizable	8
2.2.3. System integration	8
2.2.3.1. Plug-in architecture	9
2.2.3.2. User authentication	10
2.2.4. Data imports	10
2.2.4.1. Example	10
2.2.5. Self-import	11
2.3. Data access	11
2.4. Data exhibition	12
2.4.1. Web services	12
2.4.2. PUREportal development framework	13
2.4.3. Other exhibition technologies	13
2.4.3.1. OAI data providing client	13
2.4.3.2. Z39.50 protocol	13
2.4.3.3. Reference manager	13
2.4.4. Reporting	13
2.4.4.1. Standard reports	13
2.5. More information	14

1. Introduction

This paper describes the PURE application.

The aim with this paper is to give an functional overview, not to give detailed information about each module or feature. Some basic user interface examples are also included; mostly just to show what the user interface looks like.

It is important to notice, that many individual features exist, which is not even mentioned in this overview presentation. This goes for both the basic module and the optional modules. For more detailed information about the full feature set, please contact us using pure@atira.dk.

2. Product overview

We're going to get an overview of PURE by addressing four basic issues:

- How data can be modeled
- How PURE can be populated with data
- How data can be stored
- How data can be exhibited

2.1.1. Data modeling

An important part of PURE is the meta-data model:

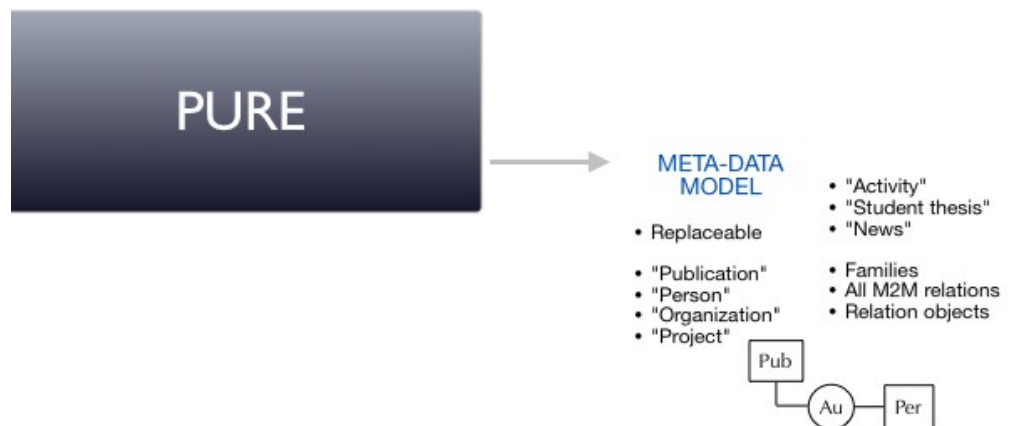


Illustration 1: The meta-data model in PURE

Briefly put, the meta-data model specifies what content objects can be registered in the system and what individual fields should be on each of those objects. Also the relations between those objects are specified. A researcher, for example, is an object of the type "Person". His or her first name would be in a field on that object called "First name". And furthermore, this "Person"-object could relate to say an object of the type "Publication" in the form of a relations' object of the type "Author".

PURE can be delivered with several standard meta-data models, which are formal structured data models such as CERIF, and which includes content types such as persons, projects, grants, organisations, publications or patents. In either of these models, these and all other content types will retain their proper interrelations.

The result is a system that contains highly structured data, which can be used for reporting and statistical analyses and many other purposes.

2.1.2. Replaceable meta-data model

As mentioned, any meta-data model can be implemented in PURE: Actually, an integral part of PURE's application architecture is a comprehensive technical framework for that. This allows a research institution to acquire PURE and still specify their own meta-data model according to local needs and demands.

2.1.3. Content families

The most widely used of PURE's standard data models resembles CERIF very much, but has more detailed publication types. It comprises 8 content families. In principle, each family can contain any number of types, which also explains the use of the term "family". However, in the case of the PURE meta-data model, only the Publications family actually contain more than one type.

The content families in the PURE meta-data model are:

- Person
- Organisation
- Project
- Publication
- Activity
- Journal
- Publisher
- Equipment
- Funding body
- Funding programme
- Student thesis
- Press clipping
- External Publication
- Clinical trial

As mentioned, each content type specifies what fields an object of that type should contain and what relations an object of that type should have to other object in the meta-data model.

2.1.4. Many-to-many relations

In this meta-data model, all primary content types retain many-to-many relations with each other. This is easy to see if you go to one of the research portals that exhibits data exclusively from PURE¹ - you will notice how you can always link from say Person to Organization to Publication to Project and so on.

2.1.5. Relations objects

Finally about modeling data in PURE, the use of so-called "relations objects" should be mentioned briefly: Imagine a relation between a researcher and his work. In the PURE meta-data model that would be the relation between an object of the type "Person" and another object of the type "Publication".

¹ Please see the references page in the PURE section of www.atira.dk

One way of retaining this relation would be to simply link the publication to the person. That would show who wrote it. However, as the person object changed, so would that information. Imagine for example the researcher getting married and changing his or her name. Without relations objects that would cause all references to this person in the system to be updated as well, which would be unfortunate: It would look as if a paper was suddenly written by another person.

Rather than using such direct links, PURE makes use of relations objects, and in this case that would be a relation object of the type "Author". It is an object type, which only exist to describe the particular relation between persons and publications. It contains a permanent "imprint" of data from the originating persons object at the time of creation. This guarantees the permanent use of the proper author name, and it still offers current info about the person by means of a link from the author object to the persons object.

A number of other situations could initiate similar changes to objects - organizations being born, closed down or merged, project being renamed or moved between organizations, etc. The PURE meta-data model makes widespread use of such relations objects but not in each case - and thereby not to the same extend as some other meta-data models as for example the CERIF model.

2.1.6. Classification editor

The built-in classification editor allows administrators to create and edit classification schemes and individual classifications under each scheme.

By classifications is understood values such as:

- Contributor role (author, editor, illustrator, translator, etc.)
- Languages
- Employed as (local types of employment, "researcher", "educator", etc.)
- Organizational type (faculty, institute, department, etc.)
- Role on project (primary or secondary investigator, manager, participant, etc.)

These examples are implemented in PURE as classifications by default. Other examples could be:

- Keywords, scientific areas
- Publication types

These and other classifications can just be added, used and maintained by administrators.

Classifications and the classification editor is a small part of PURE. However, to universities and other research organizations, unrestricted access to creating and maintaining their own classifications is a very important part of the system adaptation.

2.2. Data ingestion

Central in PURE's application architecture is the Application Programming Interface, or the API. Among other things, it is used by each of PURE's internal clients to access the application, and that of course is also true for PURE's Graphical User Interface.

2.2.1. Graphical user interface

PURE's user interface is browser-based, allowing users to access the application through their own browsers. Both Firefox and Explorer are supported on Windows and Mac, though only in relatively recent versions.

2.2.1.1. Roles and rights

Also, users are assigned roles. Roles define two things: What functionality a user should have access to, and what the user's rights are.

For example, the simpler roles have by default been limited to editing only their own content, whereas the more advanced administrator-roles are able to access and edit content from several organizations - such as faculties, institutes or departments.

About 10 pre-defined roles comes with PURE, which fit most universities. They are widely used as they are, but they can also be changed. Custom roles can be defined and added, too.

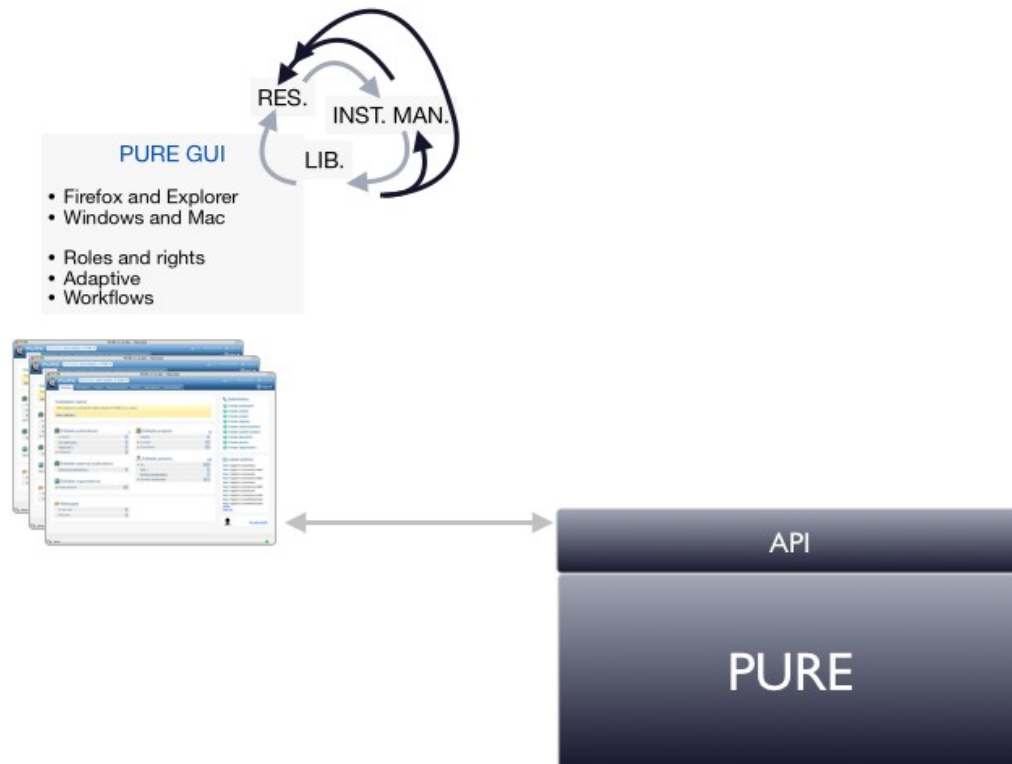


Illustration 2: The meta-data model in PURE

An easy example from the set of standard roles would be the role called "Personal user". It is a very simple role that allows only basic entering of data and administration of the users own data. Usually, this role is assigned to regular researchers that doesn't have manger- or administrator responsibilities.

Further, roles can be combined. It is not uncommon that users will have two or even three roles assigned to them at the same time. This allows much easier administration of roles - because complex custom roles can be avoided, at least to some extend.

Finally, PURE's user interface is adaptive. This means that a user only will see exactly the features that he or she actually needs according to the role. For the simpler roles, this leaves the main screen and other parts of PURE's user interface clean and easy to overview and remember. In many cases, users with simple roles will not be very frequent users, which emphasize the need for the user interface to be easy to overview and remember from work session to work session.

2.2.1.2. Customizable

Roles and rights in PURE are customizable to individual research organisations' needs.

2.2.2. Workflows

Workflows allow several users to participate in different parts of the same work process - for example the process of registering a publication. Imagine an example where a researcher, an institutional secretary and a librarian all take part in registering a publication. Lets say that the researcher will create it, the secretary approve and enrich it and the librarian validate it.

Please refer to Illustration 2 on page 7 to follow this example.

2.2.2.1. Example

In this example, the researcher will be the initial step in the workflow. First, he or she will start the registration by choosing "New publication" and filling out some fields.

Mandatory fields can be defined per organization by PURE administrators. So in this example, once the researcher finishes filling in what ever fields are mandatory, the workflow will forward the uncompleted publication to the institutional secretary at the researcher's institute.

Now the institutional secretary can complete the registration and by clicking the OK-button he or she can validate and forward the it to the library. Here, a librarian can approve the entire publication and finally submit it to the institutional repository. As this happens, a message is sent to the original researcher confirming that the publication have been accepted.

At any step in the workflow, the registration can be sent back to any prior step with information about required changes or other comments.

By deploying de-centralized data ingestion across the organization, and by basing rights-management and validation on roles and workflows, PURE becomes a tool for planning and executing de-centralized ingestion of rich and valid meta-data to a central source.

The description of data ingestion is divided into two separate sections, one about dynamic integration and one about conversion and import of historic data.

2.2.2.2. Customizable

Workflows in PURE are customizable to individual research organisations' needs.

2.2.3. System integration

In this section, focus is on how data from existing systems can be used in PURE.

With some object types in PURE, information doesn't exist anywhere and must be entered manually. That would be the case with the title of a brand new publication, for instance. But with other object types, much information already exists in other systems.

And that is usually the case with Person-objects: On objects of this type, it would not be uncommon to have information about a person's name, title, room number, employment number, direct telephone number, e-mail address, etc. And at most research institutions, such data about persons already exists in one or more systems. Therefore, using these systems as a data source for PURE will be the most appealing solution in many situations.

The same can be the case with other of PURE's object types as well. Organizations would be a typical example; usually, research organizations will have data about their institutes and departments available in one or more systems such as LDAPs or Active Directories, and - again - having such systems provide data for PURE will be the best way to go.

As these data might not be available from the same source, integration between PURE and any number of local sources is possible.

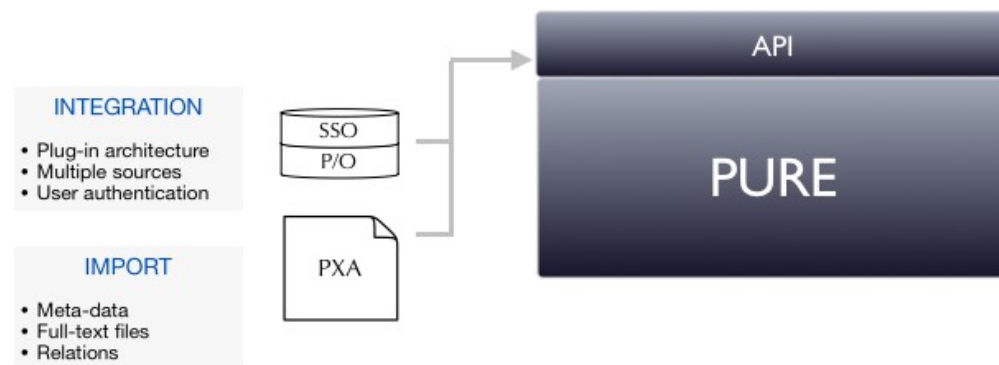


Illustration 3: External data sources typically integrated with PURE

2.2.3.1. Plug-in architecture

These local sources can be any type of system from any vendor, exhibiting their data in any structure and format. Even if global standards are used, proprietary schemas and naming conventions are still very likely to be found at different research institutions across the world. And for that reason, we added a plug-in architecture to PURE, which allows rapid and cost-effective development of custom plug-ins to PURE.

A plug-in contains a data mapping definition. It will request data from the source however data is made available, and it will correctly place data in PURE according to the mapping definition.

Further, a plug-in will schedule it's processes depending on what is needed - hourly, daily, weekly. In addition, PURE's integration plug-ins facilitate 2-way integration; if for some reason data from PURE is desired in the local system, such integration can be set up, too.

2.2.3.2. User authentication

Also user authentication and Single Sign-On capabilities should be mentioned. Authenticating users can be done in three ways. First, PURE can store user information, allowing users to enter a user name and password in PURE's login fields in order to be authenticated against the information stored in PURE. Next, PURE can look up users in external systems and authenticate them against information found there.

Typically, that would be an LDAP or an Active Directory, but any other system with securely stored user information can be used in this way. And last, PURE can be integrated under any formal Single Sign-On system that might be in service at the local research organization. In this case, users wouldn't have to enter any user information: All authentication would be done automatically and securely. An example of such a SSO-system could be CAS from YALE, but it could also be any other such system.

Common for the last two ways of authenticating users is that also here, custom plug-ins under PURE's plug-in architecture is used.

2.2.4. Data imports

So far on ingestion, we have covered how data can be manually entered and how data can be dynamically integrated with PURE from external sources. Next, we're going to have a look at PURE's options for importing data.

Imports are different from dynamic integration; they deal only with historic data. For that reason, imports are carried out only once per data set, which is what distinct them from the dynamic integration, where the same data sets are requested continuously.

Please refer to Illustration 3 on page 9.

2.2.4.1. Example

The perfect example of historic data is publication registrations from an old research registration system, which is going to be replaced by PURE: In that case, you would want all publications, which had already been registered in the old system, to be available in PURE as well.

To facilitate easy and cost-effective import of such historic data, we developed the PURE XML Archive format or just PXA. Based on the ZIP standard and XML, a PXA format file will contain both meta-data from the old publications, full-text files (DOC and PDF) and relations between all entities.

PXA files are created outside of PURE and imported via an XML I/O unit, developed for that purpose.

Based on specifications or analysis of the data set to be imported, data is mapped to match the meta-data model in PURE. The possibility to create relations between a) publications to be imported and b) organizations and persons already in PURE is important. In some cases, sufficient information to create those relations is not available in the import data set. In such cases, manual work must be done, usually be employees at the organizations in question - typically at institute or department level. Once ready for import, the validity of the PXA file will be checked by PURE. Only valid entities in the PXA file are imported. In case of invalid entities, an error log is created and saved, describing what is wrong. By fixing those entities and then re-creating and re-importing the PXA file, all entities can be imported.

2.2.5. Self-import

Self-import is an ad-on module that allows users to import publication and project data from external sources – rather than manually entering the same data. So if for example a researcher must enter his or her publication, he or she will have the option of importing it from e.g. PubMed if it already exists there. It will be imported directly into the PURE publication editor just as if it had been typed in manually, and it will proceed in the workflow from there just as if it had been typed in manually.

As standard, the following resources and formats are supported in the self-import module: PubMed, ArXiv, Reference Manager, BibTeX, BibCode, PURE and DDF-MXD.

Importantly, support for new sources can be added within 3-5 man-days, provided that the source in question provides an data access interface of some kind.

This module can also be used to facilitate access to internal system, that otherwise would not be accessible to researchers or other users.

2.3. Data access

In PURE, data access is encapsulated using an object/relational persistence and query service. Among other benefits, this allows the use of any SQL-database environment with PURE - Microsoft SQL-Server, Oracle and PostgreSQL are the three most used SQL-environments among current PURE-users.

Apart from storage, PURE also maintains an index to support searches, which is also accessed at the data access layer.

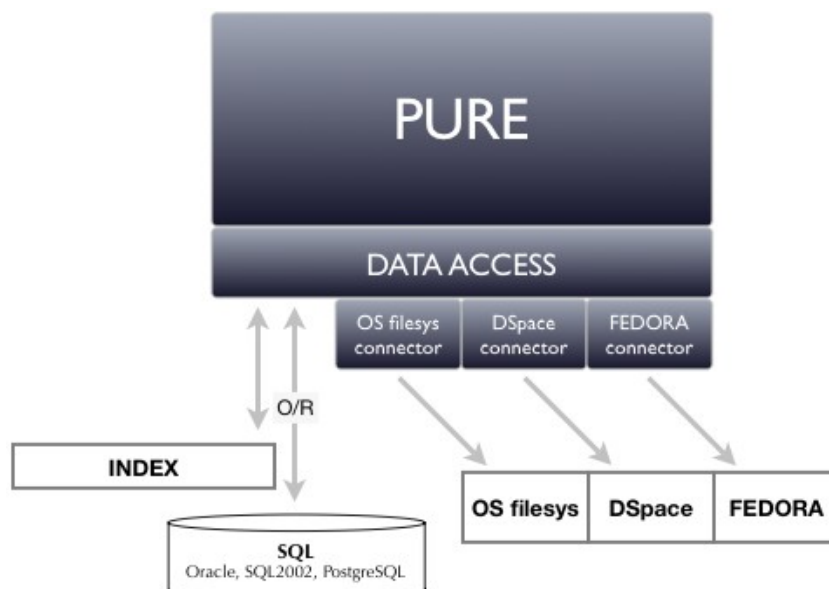


Illustration 4: Data access in PURE3

Furthermore, a so-called file connector interfaces PURE with the OS file system. When a user uploads a full-text file as part of a publication registration, the connector will store that file to the file-system of the operating system in service while still retaining relations to the relevant meta-data objects.

However, a research organization might desire more advanced storing of its full-text files. For that reason two additional connectors are also available for PURE: One for storing to a DSpace environment and one for storing to a FEDORA environment. This means that whether for long term preservation or for other reasons, PURE will work fine in a joint environment with either of these two repository platforms.

And for now, that covers all options for data storage in PURE.

2.4. Data exhibition

2.4.1. Web services

Centrally in PURE's abilities to exhibit data are the two web services. One is a Document/Literal web service, the other is an REST web service.

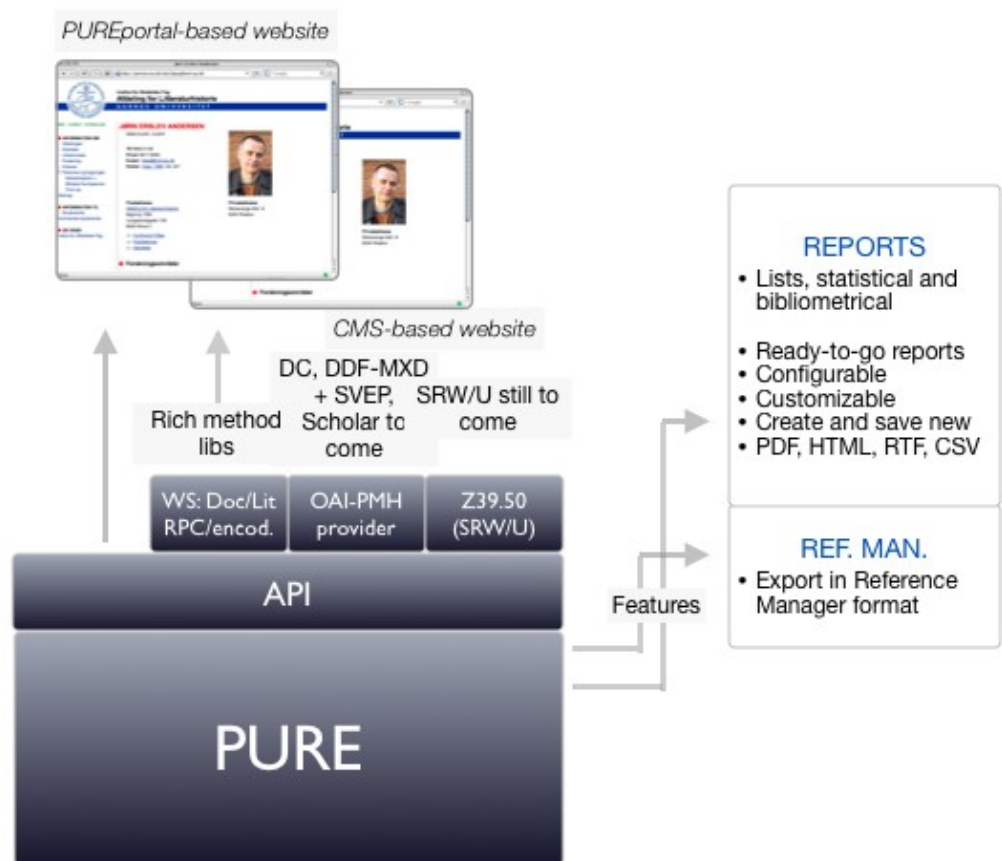


Illustration 5: Different services facilitate data exhibition from PURE using different technologies

In the case of the Document/Literal service, rich libraries of methods are available. Each method offers an easy, thought-through way of requesting specific data from PURE. Parameters such as date-range or organization can be added. Further, rich documentation of these web services and their methods is available.

Of course, one of the primary uses of PURE's web services is when data is requested from PURE to be exhibited on a website - very often a CMS-based website, to that. At least, that is the case among current PURE owners.

2.4.2. PUREportal development framework

When speaking of websites, PURE's own portal-framework should also be mentioned. It is called PUREportal, and it is an internal framework in which we at Atira offer to build customized websites for exhibiting data from PURE. The framework itself is free and comes with PURE, only man-hours spent building the website is charged for.

A PUREportal-based website is limited to exhibiting data. No editors for manual article entry are available and no other means of manual data entry exist other than PURE itself. And this is exactly what distinguishes PUREportal-based websites from CMS-based websites with PURE-data.

Together, the two web services and the PUREportal framework are how PURE exhibits data to websites.

2.4.3. Other exhibition technologies

Two more services help exhibit data from PURE. One is an OAI-PMH data providing services, the other is a Z39.50 based service.

2.4.3.1. OAI data providing client

The Open Archives Initiative defines a protocol for meta-data harvesting, usually abbreviated PMH, and the OAI-PMH service in PURE makes PURE function as a data provider under this standard. Different formats can be defined under OAI-PMH. Dublin Core is mandatory, but other formats can be added, too. Currently, PURE provides data under OAI-PMH in Dublin Core format. It also provides data under OAI in the DDF-MXD format, which is much richer than Dublin Core, but which is also much less known and primarily serves national purposes in Denmark. It is our aim to add more standard formats as they emerge.

2.4.3.2. Z39.50 protocol

Z39.50 is an old-timer among service standards. But we added it to PURE because so many library systems still interface nicely with it. What it does is to allow searches in PURE from library systems. Not everybody might need that, but in cases where the library system is an important and well-used source of information for students and researchers, the option to search the organization's own publications from within that same source can be of value.

2.4.3.3. Reference manager

Still about exhibiting data, the Reference Manager export feature should be briefly mentioned. It simply allows data from PURE to be exported in a Reference Manager native format. That's great for some users because then they don't have to register their publications twice - once in their reference system and once in the institutional repository. But as mentioned, this is not a core feature.

2.4.4. Reporting

Contrary to Z39.50, no doubt the report generator in PURE is in fact a core feature. It is a complete reporting and statistics tool, that will respond to all data in the entire PURE database.

2.4.4.1. Standard reports

It comes with a number of standard "ready-to-use" reports. They are available in three categories: Lists, Analyses and Bibliometrics. To use these standard reports, only a time interval and an organization must be chosen, and the report will run. Further, each standard report can be customized.

This means that users can add or remove criteria to the report before running it, and thereby modifying the result according to their needs. Finally, custom reports can be build from scratch. Most user interaction regarding reports is carried out either from the report tab in PURE. Normally, more than basic rights are required to do that.

But in the later versions of PURE, we added little reporting features to every role in PURE, allowing for example basic researchers to run reports and statistics on their own data.

2.5. More information

This section has briefly gone through the basics of the PURE application. With the exception of the reporting section above, only features from the basic module has been mentioned. A number of other modules exist too. Please refer to the website at atira.dk/en or the price-list.

For more information about the other modules, please refer to the PURE section of www.atira.dk or call us at +45 96 356 100.