IRMI – An open metadata catalogue for information and resource management

Hannes Kleindienst

Abstract

Today, internet based information systems are standard solutions for collaborative management of resources and information. Experiences with several applications like the Swiss environmental catalogue envirocat or the Biosafety Clearing House led to the compilation of a list of common requirements. Based on these requirements, IRMI (Information and Resource Management Interface) was developed. IRMI is solely build upon open source components, with PostgreSQL and ZOPE as the major contributions. The primary goal of the IRMI development is to get a flexible metadata catalogue systems which is easily adaptable to application specific requirements. The current development of IRMI focuses on the integration of GIS data.

1. Introduction

Many organisations deal with the management of information and resources. It is often a group of people or even a large community being involved in the management process. The organisation of such a community can be a time consuming and costly task.

The internet and specific IT solutions often help maintaining the resource documentation. In order to be successful, such tools must be accepted by all team members, which can be a challenging requirement. They must meet the specific requirements of all members, they must be simple to use, with a flat learning curve and they must be easily extendable, since the list of specific features might grow with time.

However, the general requirements for such tools are surprisingly similar, they usually include an internet based access to the system, the integration of a workflow including user- or group-based access control and some more features. International applications or countries like Switzerland also require multilingual support, meaning that the content of the information system can be management in several languages.

This paper aims at presenting a flexible solution for such tasks, targeting a broad range of possible applications. As Information and Resource Management Interface

1 GRID-IT GmbH, Mitterweg 24, A-6020 Innsbruck, Austria, www.grid-it.at
(thus being named IRMI), it defines a framework for the general purpose of managing different types of resources. Targeting large groups of users, IRMI contributes to the general theme of the enviro-info conference 2004, by providing a technical background to sharing information.

2. Requirements and application examples

2.1 Common requirements

The Swiss environmental data catalogue envirocat was one of the key drivers for the development of IRMI. Based on the experience with this and other applications, a list of requirements could be identified for a system such as IRMI. Some of the requirements are common to many content management systems; others are more difficult to be found in standard solutions. The following paragraphs discuss the individual requirements.

1. **Action Log** – All modifications to the information need to be logged with information about the time, user and element. This allows tracking the cause of a potential error in the content.

2. **Workflow** – Every information, which is entered in the system, must be reviewed and approved before it is published. The states "open", "submitted", "published" and "expired" can be distinguished.

3. **Access control** – Even if an information is published, the access can be restricted to certain groups or users.

4. **User and group management** – The access permissions need to be defined on a group level to allow several members with the same permission pattern to be organised in groups. A user can belong to one or several groups.

5. **Automatic metadata generation** – The date of creation, last modification and some other information about the history of information is automatically created.

6. **Management via internet** – The information can be managed via internet, with a secure login, if required. No specific programs or plug-in's are required.

7. **Multilingual support** – Every resource description can be translated in other languages. However, only some attributes need to be translated. Dates, numbers and other non-verbal content remains unchanged in all languages. Selections from a list of possible choices should be automatically translated.

8. **Relation management** – Every resource can be linked to other resources based on a specified relation. A person can be member of an organisation, a document can be based on a dataset or a person or organisation can be the owner of a building.
9. **Hierarchies** – All information can be organised in form of structure hierarchies. Such hierarchies can represent a folder like structure, a topic list or administrative units within a country, like cantons, regions and communities.

10. **Interoperability** – The system should allow for data exchange with other systems to avoid redundancy.

11. **Flexible rules** – For every generalisation, there is an exception. Therefore, the system should allow for flexible rules, for example to check the validity of the information entered into the system.

   The most important requirement is hard to measure and is not mentioned in the list. It is simplicity. As already mentioned, a successful system must be easy to use with a flat or even not existing learning curve.

   In many cases, only few of these requirements might be asked for, but IRMI was designed to provide all these features. Some examples of applications are described in the following sections.

### 2.2 Environmental data catalogue

The Swiss environmental data catalogue envirocat\(^2\) is the first application being based on IRMI. It contains information about environmentally relevant data, projects, documents and other resources. One of the main focuses lies on the relations between different objects, listing the data basis of a document, documents and datasets as results of a project or the tools used to derive the data.

   With about 150 groups involved in the management of the information, an easy to use publication process is required, including specific access permissions for each group. Every information within the envirocat application is linked with two hierarchy structures, one defining the administrative units (cantons, regions, communities), the other defining a list of topics based on specifications of the European Environment Agency (EEA).

   More information about the envirocat application can be found in Fink (2004).

### 2.3 Clearing houses

The Swiss Clearing House Mechanism site (CH-CHM)\(^3\) is an information platform for the application of the Convention on Biological Diversity of Rio 1992. It is mainly a collection of information notes and internet resources, but also contains details about organisations and some document and project descriptions.

---

\(^2\) [http://www.envirocat.ch](http://www.envirocat.ch), see also Fink, K. et al. (2004)

\(^3\) [http://www.ch-chm.ch](http://www.ch-chm.ch)
The Swiss Biosafety Clearing House (CH-BCH)⁴ facilitates the exchange of information concerning living modified organisms (LMOs) based on the Cartagena Protocol on Biosafety. This information platform addresses both citizens and specialists as a tool to search for related data and information. It contains descriptions of LMOs with mainly technical details as well as summary information about the decisions on the application of the LMOs. For each decision, the legal basis is also documented, including a link to the complete text of the single laws. Links to the responsible organisations provide points of contact for additional information if required.

Both clearing house platforms manage their information in several languages. Each piece of information is organised in hierarchical folders, which define the structure of the internet presentation. The Biosafety Clearing House is connected with the international database in Montreal and frequently uploads new information to the international server.

![Figure 1: Screenshots of the BCH and envirocat application](image)

2.4 Management of underground shelters

A first application of IRMI, which includes GIS features, was developed as prototype for the management of First and Second World War underground shelters in Vorarlberg, Austria. The application is used to manage the state, ownership and responsibility of the objects, along with some other attributes. Persons and organisations, lots including ownership and some general information are additional object types to be described by the system. Because of the sensitive information, access control is given special attention. The combination with additional GIS-Data (eleva-

---

⁴ http://www.ch-bch.ch
tion, topographic map, lots, etc.) allows additional queries based on the spatial relations (e.g. list of all lots in proximity to the underground shelter).

3. **Technical description**

3.1 **Overview**

The implementation of IRMI is based only on Open-Source-Products. PostgreSQL\(^5\) serves as database backend. The middleware is build upon ZOPE\(^6\), a Python-based application server (for an introduction see Latteier & Pelletier 2001\(^7\)). The presentation of GIS data is realised using the UMN Mapserver.

In its current state, IRMI is not a externalised ZOPE product, but a set of objects within the ZOPE framework, providing the required functionality. As shown in the figure below, IRMI consists of a core program and an application specific extension. The core program provides all features to interact with the database and the system, whereas the extension contains the application specific configuration, like special SQL queries, templates for presenting the content or configuration files.

IRMI currently runs under ZOPE 2.6.4, with several ZOPE products used within the IRMI framework. For the connection to the database, the psycopg2-Databaseadapter is used. TinyTablePlus\(^8\) is used for storing small amount of data in form of tables within ZOPE. Xron\(^9\) provides scheduler functions and SOAPMethod\(^10\) is used as basis for the SOAP client implementation.

With all current implementations of IRMI, the user interface is also realised based on ZOPE. This simplifies the interaction with IRMI, since it is possible to directly access the session content. A fully featured interface for any external application is already available for the database query, but not yet for content modification and user management.

\(^{5}\) [http://www.postgresql.org](http://www.postgresql.org)
\(^{6}\) [http://www.zope.org](http://www.zope.org)
\(^{8}\) [http://zope.org/Members/hathawsh/TinyTablePlus](http://zope.org/Members/hathawsh/TinyTablePlus)
\(^{9}\) [http://zope.org/Members/lstaffor/Xron](http://zope.org/Members/lstaffor/Xron)
\(^{10}\) [http://zope.org/Members/EIONET/SOAPMethod](http://zope.org/Members/EIONET/SOAPMethod)
3.2 Database backend

The database model is well structured and can be organised in several areas. An object or resource to be described is called element and is specified in the element definition area. A number of tables define name, label and help texts as well as the attributes for all element types. For each attribute, the format, special conditions, multilingual support, label and help texts as well as some other specifications are defined.

3.2.1 Database structure

The core area holds all common attributes such as title or abstract as well as meta-information about the elements such as owner, creator or date of creation and last modification. Sub-elements define sets of attributes within a common context. The temporal extent, for example, includes a free text definition as well as date attributes for the range and one attribute for the interval of repeated observation. The communication sub-element specifies the communication method (telephone, fax, email) and the number or address. Additional common sub-elements are postal address, spatial extent, distribution information, download link, and some more. Every sub-element is linked to the element table via a one-to-many relation, allowing multiple occurrences of a sub-element type for one element.
The hierarchy systems are defined as a separate area. A hierarchy is a structured container for elements with parent-child relationships between hierarchy entries. IRMI allows for different hierarchy types, each with specific behaviour. The folder hierarchy is a standard container and provides access rules depending on the ownership of a folder. The location hierarchy is used to define a spatially dependent hierarchy like administrative areas within a country. It includes a spatial reference for each hierarchy entry. Topic hierarchies are selections of themes or keyword, which might be linked to a thesaurus. A relation table allows each element to be placed in one or several hierarchy entries.

The relation between elements is managed in form of a simple relation table. For each relation, the relation type is defined, as well as the direction of the relation using references to the "from"- and "to"-element. A comment on the relation can provide further explanation.

The multilingual support is managed in a way that for every language dependent content, a second table with the language code as additional primary key is defined. This almost doubles the number of tables within the data model, but following a strict naming convention, the complexity of the data model is not that much affected.

In order not to restrict the definition of new element types, specific database tables can be defined for each type. This is the only area, which can be extended depending on the specifications of the application and its element types.
The GIS data is currently stored in external shape files, linked via unique ID's. The integration of the GIS data in the PostgreSQL database using the PostGIS extension is currently under development.

### 3.2.2 Querying the database

The database structure described above allows for many different view angles of the database content. In addition to the standard searches on the common attributes of the core element tables, the hierarchies as well as temporal or spatial context of the elements can be used as search criteria. With the PostGIS extension, even spatial searches can be defined, listing elements within or in proximity to a predefined area.

### 3.3 ZOPE application

IRMI is a collection of ZOPE objects such as Python Scripts, Zope Page Templates and some more, in combination with some Zope External Methods. In order to be included in existing ZOPE applications, the IRMI folder must be imported and placed somewhere within the application. The inheritance feature of the ZOPE Object Database allows any IRMI function to be called from anywhere within the application by simply adding the IRMI path to the current location path. IRMI provides methods for the following tasks:

1. **Query the database** – IRMI accepts a number of search conditions, like keywords to be searched for in title and abstract, search within a hierarchy structure, ownership, date of last modification, and so on. The search functions are available via CGI calls and, therefore, can also be accessed from external applications.

2. **Authenticate user** – With username and password provided, the authentication of a user can be verified. If the login is successful, the respective flag will be set in the session, which is valid for a predefined period or until logout of the user.

3. **Retrieve single elements** – Standard methods allow reading all attributes for a specific element. If the application works with multilingual content, also the language of the element needs to be defined.

4. **Modify and write element content** – The content of an element can be changed via CGI calls. Every change leads to an update of the session content. After all modifications are done, the element can be written to the database. The previous version will be kept in a backup database (currently under development).

5. **Manage users and groups** – A number of methods provide access to the users and groups defined within the database. Depending on the role of the current user, new users and groups can be defined, membership and roles can be set or existing users and groups can be modified or deleted.
In order to accelerate the implementation of new IRMI applications, templates are provided for most of the views. This includes the display of search results, predefined forms for login, user and group management as well as management of the elements content. The appearance of the templates can be controlled via CSS. The inheritance feature of the ZOPE database allows creating application specific templates, which override the predefined versions.

3.4 Interoperability

Many existing information systems deal with the management of metadata. In order to avoid redundancy in the different databases, the interoperability between the systems is a major requirement.

IRMI includes a SOAP client implementation, which allows calling external servers, assuming they provide a SOAP interface. Within the BCH application, this feature is used to synchronise the content of the Swiss server with the international server located in Montreal, Canada. A SOAP server is not yet implemented, but is planned for one the next versions.

In addition to SOAP, IRMI also supports CGI-based communication with external servers. Via CGI, it is possible to query the IRMI database using all available query features. The management of the IRMI content via CGI is currently under development.

4. Five steps to an IRMI application

IRMI is not a product, which can be installed and used right away. It requires some configuration and adaptation depending on the application, for which it is supposed to be used. However, some standard configurations and predefined templates enable a fast implementation within only few days. This chapter explains the steps, how to create an IRMI application from scratch.

a) Content definition

Depending on the objective of the application, the resources to be managed need to be defined. For each resource, the required attributes have to be specified including attribute type, format, special conditions, labels and help texts. For free text attributes, a flag can be set to indicate a multilingual attribute. Attribute types can be standard text (including date or number), sub-elements (each with a set of attributes), selections from pick lists, relations to other elements or a link to one of the hierarchy systems.

A set of predefined attributes can be used, including several sub-elements, which consists of a number of attributes. A postal address for example is defined by delivery point, city, postal code, region and country.
b) **Database creation**

For the default configuration of the database, a predefined script can be used in order to create all required tables and queries. Additional tables for all element types need to be created separately.

c) **Application layout**

One of the major tasks is the specification of the layout and the development of the application front end. The following features or screens should be included: login and logout, database search and list view, element view and edit pages, user and group management and editing personal settings. For most of these components, predefined HTML templates can be used and formatted via CSS. If a specific layout is required, own templates can be defined.

If the application is realised based on ZOPE, it is possible to access all features of IRMI as well as the session content. Otherwise, IRMI can be accessed via CGI, returning search results, element content or other information as HTML or XML formatted text.

d) **Definition of users and groups**

Before starting to work with the systems, the organisational structure needs to be set up. Some consideration must be given to the definition of groups, since they control part of the access permissions. A list of users needs to be defined including the group membership and role for each user.

e) **Configuration**

Finally, some general settings need to be defined, like whether the system supports workflow or multilingual content, the behaviour after login or logout, the default language, and so on.

5. **Outlook**

At the moment, IRMI is used for the management of data catalogues like *envirocat* and information exchange platforms like the BCH. For the near future, a new application area is targeted: it focuses on tourism, providing information about the infrastructure of a region. For this application, the integration of GIS data as well as GIS-based features such as spatial searches is a major requirement. Therefore the current development concentrates on the integration of the GIS-module, utilizing the PostGIS-extension of PostgreSQL, UMN Mapserver for visualisation and GRASS as GIS backend for more sophisticated queries.
Bibliography


Related internet resources

http://www.envirocat.ch
http://www.ch-bch.ch
http://www.ch-chm.ch
http://www.zope.org
http://www.postgres.org
http://mapserver.gis.umn.edu