Introduction

[DSpace-CRIS: a brief functional description] [Logical and physical data model] [About the CERIF compliance]

Universities and researcher centers are rethinking their communication strategies, highlighting the quality of their research output and the profiles of their best researchers. Listing publications from an Expert Finder system may represent a solution, but providing an Expert Finder system within an Institutional Repository (IR) is a more innovative approach. This idea was developed in 2009 by the University of Hong Kong Libraries, along with Cineca technicians, and applied to their IR, The HKU Scholars Hub at http://hub.hku.hk/, powered by DSpace.

In 2013, Cineca and HKU went one step further and released DSpace-CRIS, an open source general solution to enrich DSpace with CRIS entities and concepts. “A Current Research Information System, commonly known as CRIS, is any informational tool dedicated to provide access to and disseminate research information.” (www.eurocris.org)

At institutional level, a CRIS is a tool for policy making, evaluation of research based on outputs, documenting research activities and output and assistance in project planning and constitutes a formal record of research in progress. For the individual end users, a CRIS is essential to evaluate opportunities for research funding, avoid duplication of research activity, analyze trends, have references to full text or multimedia scholarly publications, locate new contacts and identify new markets for products of research. Typical forms of output are researcher CV, management information, reports to funders, research bibliography and commercial output reports.

DSpace-CRIS consists of a data model describing objects of interest to Research and Development and a set of tools to manage the data. Standard DSpace used to deal with publications and data sets, whereas DSpace-CRIS involves other CRIS entities: Researcher Pages, Projects, Organization Units and Second Level Dynamic Objects (single entities specialized by a profile, such as Journal, Prize, Event, etcetera; because any profile can define its own set of properties and nested objects).

DSpace-CRIS comply with the CERIF[1] standard indeed the key components of the CERIF Data Model are supported natively in DSpace-CRIS: universally unique identifiers (UUID), time stamped relations, semantic characterization.

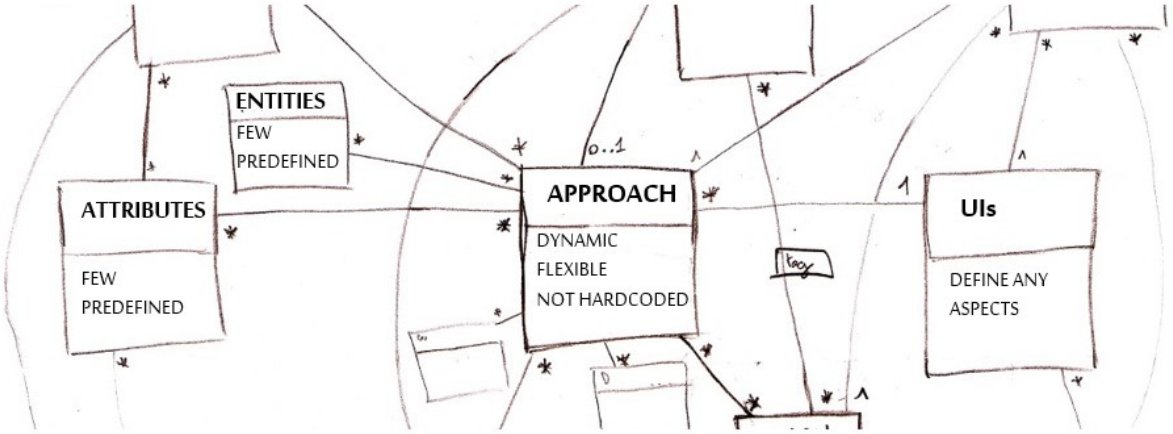
The flexibility of the DSpace-CRIS data model allows the Institutions to configure the system in several different ways, so that the level of compliance with CERIF may depend on the specific configuration adopted by the Institution. Some de-normalizations are sometimes even recommended, because they are easier to adopt at the project start-up, when data are already available in other systems even if they are not enough structured (i.e. Journal information stored in the publication record or funding information stored in the project record).

DSpace-CRIS supports interoperability through SOAP WebServices for read only access to CRIS information and import/export from CSV. An export in CERIF XML 1.6 and the support for CERIF over OAI-PMH will be added in the second half of 2016. Please refers to the project website for up-to-date information on the roadmap.

DSpace-CRIS: a brief functional description

While DSpace just allows to manage publications, DSpace-CRIS permits to handle other entities such as projects, people, departments, etc., via UI.

Every entity data structure is configurable via UI (User Interface) by adding simple or complex fields. Of course the system also allows to manage new relations among entities.



Once the data model has been configured, all entities may have a proper public page, where some or every information may be shown, and they may be searched and browsed.

The DSpace-CRIS entities can be used as authority file for publication’s metadata (dspace items), thus producing manageable lists for Authors, Journals, Events, Projects, Funders, etc.

Any relation between a DSpace item and a DSpace-CRIS entity or among DSpace-CRIS entities can be automatically explored in the inverse manner so that it can be produced inside the Researcher Page a list of publications authored by the researcher, or the list of researcher members of the organization, and so on.

Among other advantages, DSpace-CRIS has a unique, unambiguous way to assert that a publication (i.e. a dspace item) is related to a person or to a project without rely on the mere “string value” of the metadata, thus solving the problem of the homonym attribution issues and allowing the building of more advanced functionalities such as:

- aggregation of statistics at different levels: publication lists for researchers, projects for researchers, researchers for OrgUnits, etcetera;
- graphically display and analyze various relationship networks such as co-authorship, collaborative projects, departmental interactions, etcetera.

Moreover, in order to manage information such as the publication citation index, the analysis of usage statistics, the public statistics, and so on, it provides an automatic data retrieving service from external systems[2], actually from PubMed Central, Scopus®, Web of Science® and Google Scholar®.

Logical and physical data model

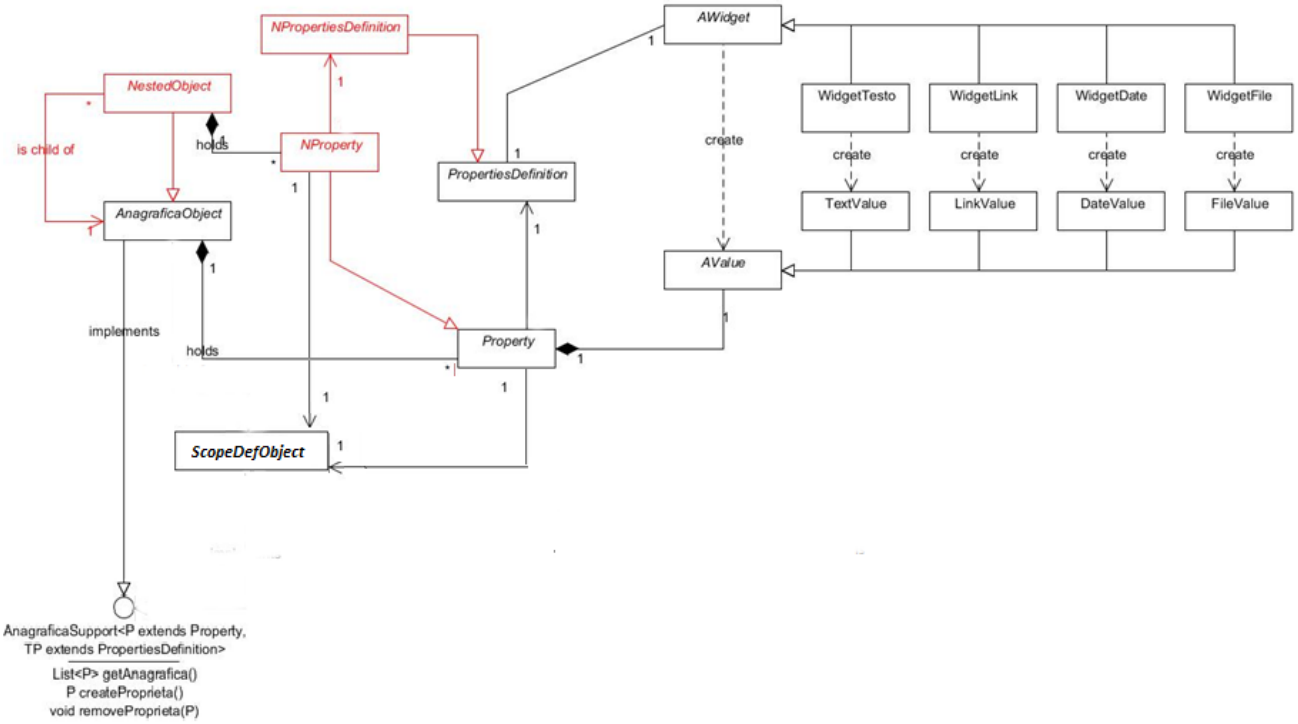
In order to manage the persistence of the "dynamic registries" DSpace-CRIS embrace JDynA[3], an open source JAVA library whose aim is to provide JPA persistence for such dynamic structure.

As of DSpace 5, the DSpace-CRIS database now upgrades itself **automatically**.

We now use Flyway DB along with the SQL scripts embedded in the dspace-api.jar to automatically keep your DSpace database up-to-date. These scripts are now located in the source code at: **[dspace-src]**/dspace-api/src/main/resources/org/dspace/storage/rdbms/sqlmigration/postgres

As Flyway automates the upgrade process, you should NEVER run these SQL scripts manually. For more information, please see the README.md in the scripts directory.

The core entities of JDynA are shown in the following diagram.



Abstract class AnagraficaObject is inherited by all the objects with dynamic registry. Property class contains both the field definition (PropertiesDefinition) and its value (AValue). AWidget is a framework inner class born to instantiate the correct data types (String, Data, etc.) and to manage the data input modes (autocomplete, dropdown, etc.).

The diagram does not show all the data types that may be handled. It is also possible to support numeric values, links to other entities, Boolean data, classifications, opened or closed subjects list.

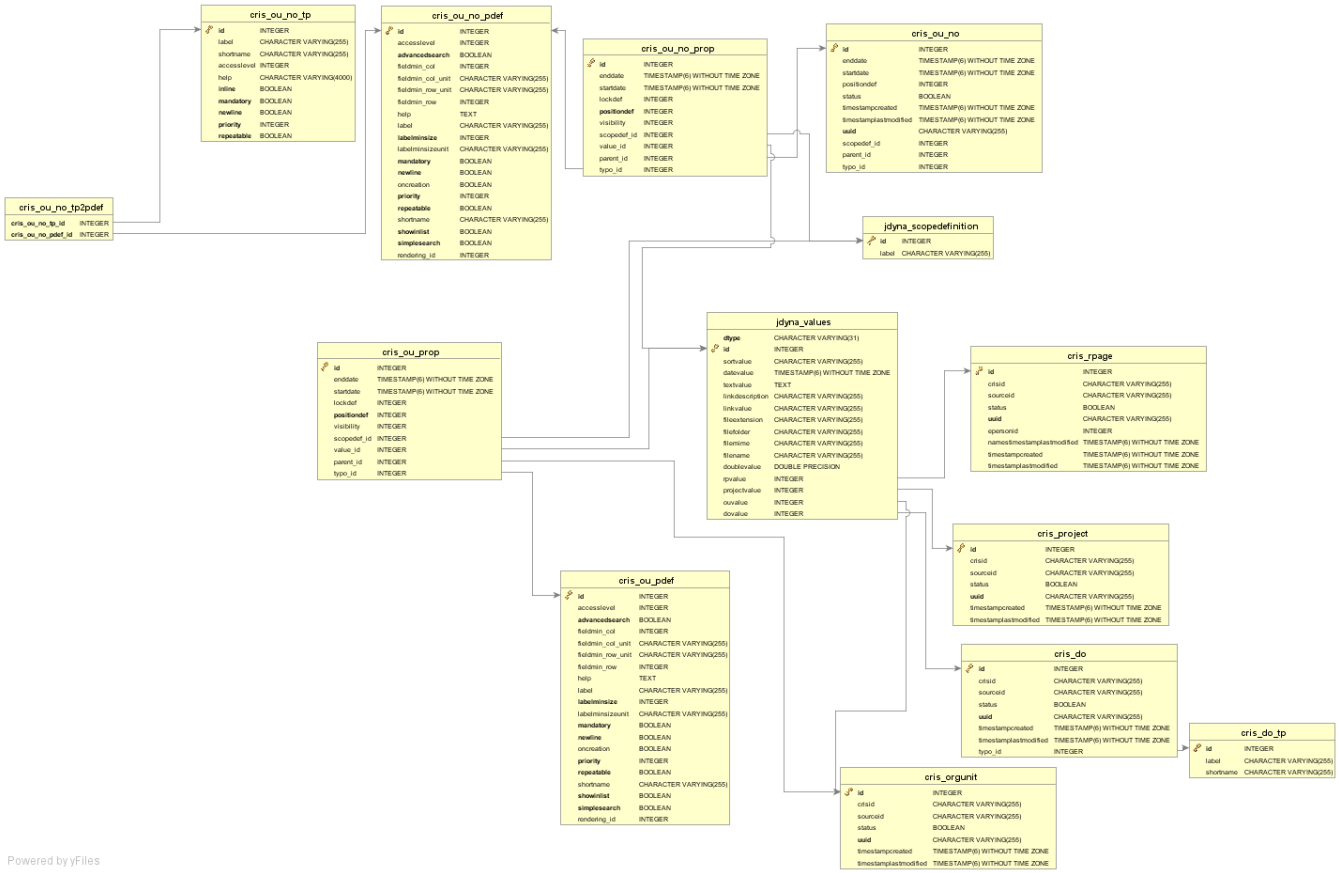
JDynA may be extended, in order to create other data types. At the end, the DB structure can be seen as a richer key-value storage.

The following E-R diagram shows the DSpace-CRIS core entities.

JDYNA_VALUES is included in the core entities diagram because it collects the value of all the information type managed by DSpace-CRIS (in particular the references to system entities: RPVALUE, PROJECTVALUE, OUVALUE and DOVALUE). Thanks to JDYNA_VALUES DSpace-CRIS allows to manage unlimited relations among any object within the module.

CRIS_DO table represents the "Dynamic Object", later renamed "Research Object". It permits to handle all the research entities dedicated to dissemination information process, such as labs, instruments, awards, etc. In fact JDynA supports the definition of object whose configuration depends on the typology (different set of fields for labs, awards, etc.)

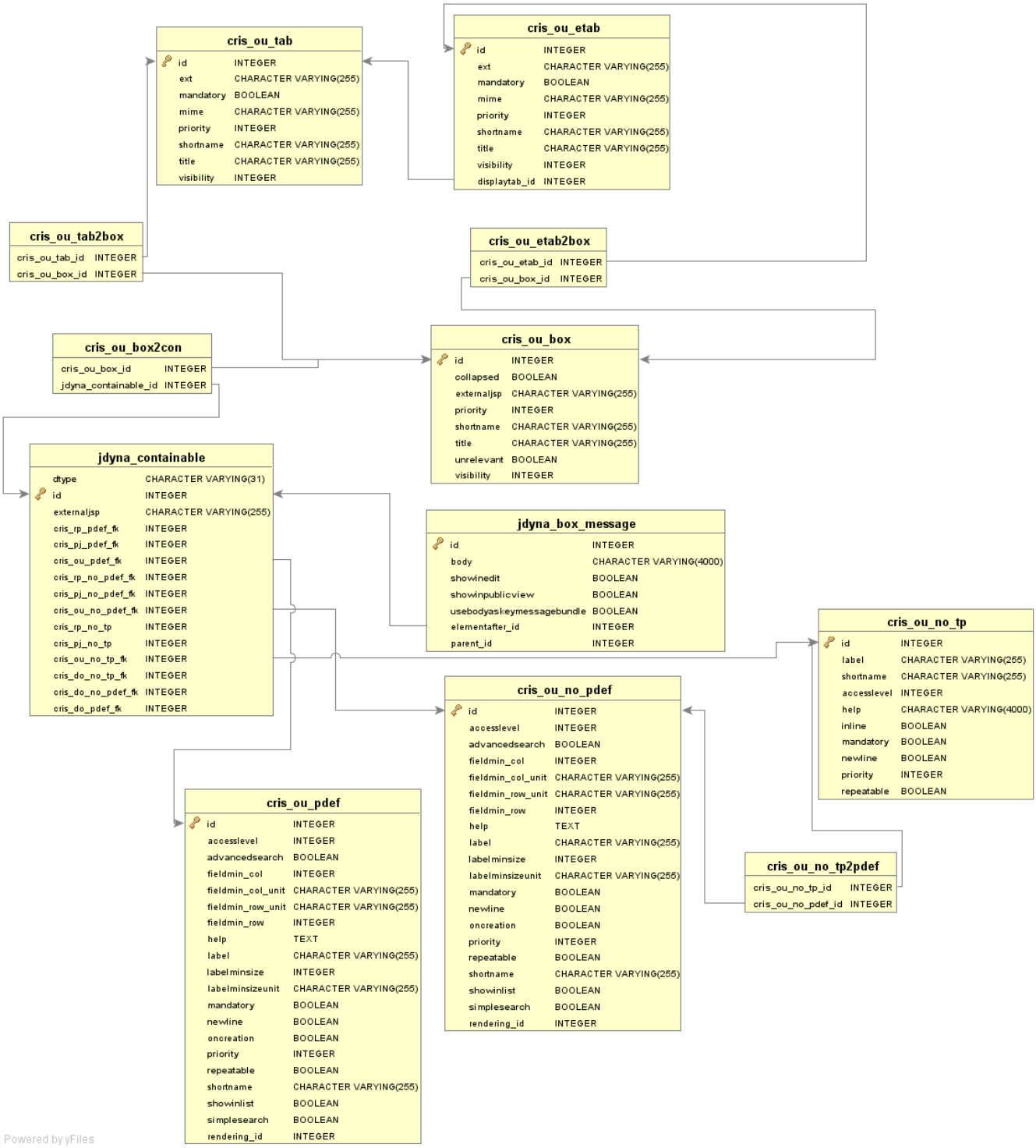
The following E-R diagram shows the tables related to OrgUnit, ResearchPage (the equivalent of CERIF Person) and Project entities.



CRIS_OU_PDEF, CRIS_OU_NO_PDEF and ***CRIS_OU_NO_TP*** tables contain the semantic of the organization unit defined by the institution. In JDynA the object semantic is persisted in the data structure.

The following E-R diagram shows the widget JDynA structure used by DSpace-CRIS.

The following E-R diagram shows the configuration tables of OrgUnit entity.



The configuration process allows defining tab to display the entity data and their groupings (box).

About the CERIF compliance

The CERIF export is not yet released. This documentation describes the approach that we are using for the feature development that will be released in the next version.

Since the actual data model is effectively build upon the specific need of the institution through the configuration process, the actual degree of compliance with the CERIF data model can vary. The system provides a general infrastructure that allow to translate different configurations to the CERIF data model, see “Configure the CERIF Mapping”. The basic configuration provided out-of-box is a simplified implementation of the key CERIF concepts and entities, it could be extended to support furthers CERIF entities and relationship.

For any CERIF Entity is possible to define one or more corresponding DSpace-CRIS entities (1:N), in this way is for example possible to better characterize a publication from a journal or a grant from the underline project.

Once that the entities mapping CERIF vs DSpace-CRIS has been configured, i.e.:

- People à ResearcherPage

- OrgUnit à OrgUnit

- Publication à DSpace items; Dynamic Object: “Journal”; etc.

The property definition of the DSpace-CRIS Entity can be mapped to the attribute of the CERIF Entity.

To manage the multilingual feature of CERIF there are two possible approach:

- The first one, often the most practical, is to define separate property definitions for each languages that the Institution like to support in the system;

- The second one, more accurate, is to use a nested object with two properties one to hold the language and another for the actual value. The language can be stored as simple string holding the ISO code or as a pointer to a dynamic object “ISO Language” where all the available languages are stored. The second one should be preferred where the data is edited via the DSpace-CRIS UI.

To manage the relationship between CERIF entities and specifically the semantic characterization of such relationship there are two possibilities:

- The first one, often the most practical, is to define separate properties definition for each relation meaning so to have, for example, a specific property definition that track the “principal investigator” relation between a project and a people. This mean that for all the relationship meaning that an Insitution want to support a property definition need to be separately configured, i.e. principal investigator, co-investigator, director, member, associate member, etc.

- The second one, more accurate, is to use a nested object with two properties one to hold the relation meaning (semantics) and another for the actual value. The first pointer should be configured to link to the dynamic object “CERIF Semantics Classification” that are providen out-of-box in the system filtering on a specif CERIF Semantic Schema, i.e. Person role. The second pointer is the link to the target entity of the relation.

For the relationship is usual to use in the same system both approach, keeping the most relevant relationship as separated property definition and tracking the other as nested object. The main reason to make such difference is related to the major flexibility that a separate property definition provides in terms of UI configuration (edit/visualization permission and positioning) and UI customization simplicity.

Finally, the JDynA data model has been extended introducing a ScopeDef object, actually at most a placeholder. The property level holds attributes “startdate”, “enddate” (validity dates) and a reference to such object.

Even if these attributes have not been managed by JDynA/DSpace-CRIS yet (they depend on an ad hoc customization), this activity is on the DSpace-CRIS roadmap, and will provide a third way to manage the CERIF mapping in a more natural way. The “scopedef” ref attribute may be used in order to manage multilingual contents or the relation semantics instead of setting specific propertydefinition or use nested objects.

The three different ways to manage the DSpace-CRIS / CERIF mapping allow to choose the better balance between the complete representation of the research world that CERIF aims to achieve and the data that the Institution really holds or is able to provide maximizing performance and end-users functionalities.

Property Definition, Nested Object and, in future, ScopeDef and validity time stamp attributes will all provide different behaviour in terms of performance[4], UI configuration and indexing.

---

[1] http://www.eurocris.org/Index.php?page=CERIFreleases&t=1
[2] PubMed Central, Scopus, Web of Science and Google Scholar are trademark of their respective owners, the use of the data in dspace-cris is subject to specific agreements with the data owners that may require subscription or other fees payment.
[3] https://github.com/Cineca/JDynA/wiki
[4] Specifically property are retrieved in eager mode with the object that they belong to; instead, nested objects are loaded in lazy mode when needed. That mean the information that are not alway required for display or indexing work well as nested object than as property from a performance point of view. The extended CERIF attributes (scopedef, start/end date) are expected to be in eager mode with the scopedef objects cached to avoid database query.