Towards the Refactored CERIF Model
Proposal for a presentation at the CRIS 2022 conference

Jan Dvořák
Dragan Ivanović

Extended abstract

The paper informs on the on-going activity at euroCRIS to refactor the CERIF model. The features of the present model are briefly explained, and one also presents the shortcomings that motivate the refactoring effort. We state the goals and the ways to achieve these: the new modelling approach for CERIF, introducing a modular structure and improving documentation.

The current CERIF

CERIF 1.x (Common European Research Information Format) is basically a graph model of the domain of research information. The model is presented as an Entity-Relationship one, with flexible and semantically rich “linking entities” that realise the edges in the graph.

The universal structure of linking relationships in combination with storing the semantics of the relationships in a specialised Semantic Layer leads to a great flexibility of CERIF. As a rule, there is typically more than one way of expressing facts.

Shortcomings:
- Monolithic structure, non-scalable development process
- Entity-Relationship modelling approach not a good fit to support information interchange
- The model sometimes needs to take care of technicalities (the Semantic Layer to store semantic classifications, the Federated Identifier infrastructure to store identifiers), which makes it miss the point: which semantic classifications containing which classifications are needed, which identifiers are supported. The semantic information is then not part of the model.
- The great flexibility does not help information interchange. Particularly, consuming CERIF information can be challenging.
- Sometimes the model hides important concepts in technical details: e.g. a research dataset is represented as a ResultProduct with a classification that it is of type “dataset”. This contributes to a perception of CERIF as a complex model.

Goals

The following goals of the CERIF Refactoring Pilot project have been formulated:

1. Modernise CERIF
   a. Change the modelling notation
   b. Adapt CERIF for usage in APIs and for Linked Open Data
c. Modern serialisation formats
2. Make the model easier to use
   a. Emphasise the conceptual model
   b. Improve documentation
   c. Take away the perceived complexity of CERIF
3. Involve the community in further development of CERIF
   a. Modularity
   b. Open source software development practices

The strong features of current CERIF will be kept. Multilinguality is one such feature.

New approach to modelling CERIF

The new model aims at capturing more of the important semantics than was possible with CERIF 1.x. In particular, many relationships between different semantic terms can now be represented as associations between classes in the model itself, e.g. the generalisation-specialisation relationship between Authorship and Contributorship. Parts of this were previously represented by relationships between classifications stored in the Semantic Layer, so they were not part of the model. It also introduced the need to have certain “anchor” classifications, such as Part (is-part-of / has-part) or Specialisation (is-a / can-be-a) to be introduced in the Semantic Layer itself to make such linking possible.

We had two candidates for modelling approach: the Object Oriented Modelling and the Semantic Web modelling. The Object Oriented approach was selected as the primary one, the Semantic Web will be a secondary one with its main use in supporting RDF-based serialisations.

Modularity

The refactoring proposes to divide the CERIF model in a Core that will contain the essential entities, and then the definition of specific areas of entities that completes the model and that should be defined by experts in the area.

Commonly used functionalities and items are placed in the CERIF Core (e.g., Person and Organization). The Core also contains high-level abstractions that are expected to be specialised in the modules (e.g. Document).

The Core also defines some datatypes that will be used. These are primitive types (such as String, MultilingualString, Integer or Date), commonly used vocabularies (such as Country and Language Tag) and structures (such as Monetary Amount). Modules are expected to use these types. If they find a new one, they can propose it to be added in the Core.

A module is described by its aim and scope. A module provides new entities and relationships and puts them in the context of other entities and relationships from the Core and from other modules it depends on.

We expect CERIF modules will be built by communities of users interested in a particular perspective of research information.
Documentation

Formalised descriptions of the entities in Markdown files on Github, accompanied by UML diagrams produced by PlantUML.

A Markdown file describing an entity contains the following sections:
- Heading: The name of the entity (e.g. Affiliation Statement).
- Definition: The scope of the entity and its meaning.
- Usage Notes: An additional description to communicate the purpose of the entity.
- Specialization of: The link to the super class
- Attributes: The list of attributes: properties whose values are simple or composite datatypes.
- Relationships: The list of relationships that are important for this entity.
- Illustrative Diagram: A UML diagram showing this entity in its context if one is available.
- Matches: Describe any external entities or classes this entity matches.
- References: References to external resources in case these are referenced.

A Markdown file describing a datatype contains the following sections:
- Heading: The name of the datatype (e.g. ORCID iD).
- Definition: The scope of the datatype and its meaning. The definition can be omitted for simple types.
- Notes: An additional description to communicate the purpose of the datatype.
- Components: The list of fields the datatype is composed of. Similar to attributes for entities.
- Matches: Describe any external datatypes this datatype matches.
- References: References to external resources in case these are referenced.

Verification

The model use-cases used to verify our approach:

1. A researcher is willing to move from a country to other (e.g. France to Australia) to a different institution. It is expected that his profile and vitae may be transferred electronically to the new institution via a standardised model and format that can be digitally signed as a verifiable claim or verifiable credential (following the directions of the W3C working group of the same name).

2. Partners in a multinational project funded under the EU H2020 programme need to share project participation data electronically with the coordinator for reporting under the particular rules of a scheme or call. The partners include non-EU members, e.g. from China.

3. A researcher is looking for a research infrastructure for a very concrete physics experiment and wants to know if there are e-infrastructures available in other countries that he could eventually use, with or without paying for it.

The refactored CERIF will also support widely used serialisation formats: JSON for APIs and RDF formats (RDF/XML, Turtle, JSON-LD, …) for use in Linked Open Data scenarios.
Resources

The CERIF Core repository on Github -> https://github.com/EuroCRIS/CERIF-Core
A sample module to represent scholarly publications ->
https://github.com/EuroCRIS/CERIF-ScholarlyPublicationsModule

Brief outline of the talk & paper

- CERIF 1.x features and shortcomings
- Goals for the refactoring
- The modelling approach used
- The modularity
- The description of the model
- Examples