CRIS2016

FRIS R3 - CERIF XML in large scale exchange of research information

Thomas Vestdam[a,*], Brian Plauborg[a], Leen Van Campe[b]

[a]*Elsevier, Niels Jernes Vej 10, 9220 Aalborg, Denmark*
[b]*EWI, Koning Albert-II laan 35 box 10, 1030 Brussels, Belgium*

Abstract

The ambition of FRIS R3 is to take responsibility for a broader regional research information infrastructure that enables the participating Flemish research institutions, and other service users like funders and assessment entities, to uniquely identify and reuse relevant "research entities" such as researchers, organisations, projects, research output as well as other research related aspects – as well as showcasing Flemish Research. In this paper we present the motivation, development processes, technical architecture and implementation behind the new Flemish infrastructure for research information called FRIS R3. FRIS R3 also serves as an example of a real world implementation of CERIF XML that uncovers challenges that must be addressed if CERIF XML is intended to stay relevant in commercial and large scale contexts.

Keywords: CERIF; CERIF XML; FRIS (The Flanders Research Information Space); Exchange of research information; Research information infrastructure

1. Introduction and background

The Department of Economy, Science and Innovation [1] in Flanders (EWI) has for a number of years run the Flanders Research Information Space [2] (FRIS) R2 portal [3]. The FRIS R2 portal aims to be a simple, transparent and open platform in order to make Flemish research information publicly available. For years data have been fed into the portal as a semi-manual process, where meta-data on research production was uploaded every 6 months by the participating research institutions – i.e. the *data-providers*. FRIS R2 is built on top of the CERIF 2006 relational data structure, and the data-providers use bulk CERIF XML [19,20] uploads to update the research information at their institution. Due to a number of factors, both with the underlying CERIF data structure and the implementation of the FRIS R2 portal, a number of pressing use cases could no longer be satisfied.

Data timeliness; the research information of each data-provider was updated as a manual bulk upload. This resulted in a labour intensive update process, and FRIS R2 data is not at all times an accurate reflection of the state of research in the region. Thus hampering both research discovery in the public portal but also the statistics and analysis produced for use in political decision making at EWI.

Data quality; data validation was static and the bulk submission process was not conducive for evolving the data

* Corresponding author.
E-mail address: T.Vestdam@elsevier.com

quality over time. A number of research institutions were still not actively managing their research information and the batch oriented approach was not encouraging them to change to a management model with a focus on data quality.

Entity identification; a number of the submitted research entities would be duplicates, for example the same researcher working for different universities at different times or a project performed in collaboration at multiple Flemish research institutions. The existing data-structures and functionality did not provide any mechanisms for identifying and associating these "logical duplicates" with each other.

Open data; there was no harvest functionality if users wanted to use the aggregated research information for other purposes. Specific use cases were to use the FRIS data as a part of the funding submission process for the regional funders like FWO [3] or facilitate transforming the aggregate research information set to a RDF [4] format and exposing it as a SPARQL [5] end point [18].

In 2009 EWI formulated the scope for an ambitious successor to the old portal; the FRIS R3 project. The goal of this project is to rebuild both the FRIS portal and its underlying infrastructure, in order to support system-to-system integration, improve quality of the data ingested into FRIS, and supply options for "real-time" updates. More importantly, the ambition was to expand the scope of FRIS by evolving the responsibility from mostly showcasing Flemish research into taking responsibility for a broader regional research information infrastructure that enables the participating research institutions, and other *service users* like funders and assessment entities, to uniquely identify and reuse relevant "research entities" such as researchers, organisations, projects, research output as well as other research related aspects. Hence, FRIS R3 aimed to support a number of different scenarios:

- Increase visibility of Flemish research, potentially increasing the citation index of Flemish knowledge institutions
- Enabling easy location of individual experts and provide meaningful inter-disciplinary collaboration suggestions
- Provide reports and benchmark indicators to support government policies
- Coupling of financial information to research information enabling budget analysis for administrators and easy funding discovery and application for researchers
- Integration with other research networks potentially linking the research information for all European knowledge institutions

Specifically, the requirements to FRIS R3 involved the following:

Data timeliness; enable full system-to-system integration. The goal was to transition all data-providers from a bulk update process to an incremental process where changes are published to FRIS as they happen in the data-provider systems – e.g. their CRIS. For some data-providers this would involve starting to manage research information directly in order to be able to satisfy this requirement. However, in order to support the needs of smaller institutions that might not have the necessary resources to invest into a system-to-system integration, it was decided to also support manual file upload and manual entry of data in the admin user interface of FRIS R3.

In addition, EWI also established a governance structure in order to support the transition process, including securing funding for the universities in Flanders, so they could invest in incremental system-to-system integrations between their local system and FRIS R3. Furthermore, EWI play an active role in the Pure user community where integration with FRIS is a major topic, as FRIS integration have been implemented as an out-of-the-box feature in Pure [9] in collaboration between the Pure User community and EWI.

Data quality; EWI wanted to be able to set data quality goals in collaboration with the data-providers and evolve these business rules over time in "real time" directly in FRIS R3 without having to any involve developers.

***Entity identification*;** it was important to EWI to be able to improve entity disambiguation across data-providers by promoting a *bi-directional relationship* between the FRIS system and the data-provider systems. By allowing data-providers to look up research information entities in FRIS, it would be easier for the individual researchers to add quality data to the data-provider systems - simply by importing relevant entities from FRIS, such as persons, organisations, and journals, or even projects and publications. FRIS will in turn get source identifiers for entity disambiguation when the data-providers push new data into FRIS.

Open data; aggregated FRIS data should be made easily available for use by other service users - i.e. the public. A set of standards compliant web services must enable data lookup for the bi-directional data-provider integration use-case, and facilitate harvesting. Hence, the next generation of the FRIS portal would be a client in the new infrastructure, in contrast to being built on top of the infrastructure. The FRIS R3 portal is scheduled to be available late 2016.

The use of existing standards whenever possible was a very high on EWI's agenda. Specifically this means using CERIF XML as the exchange format between FRIS R3 and the data-providers and other service users, but it also means utilising other technical standards – see sections "3 Architecture" and "4 Implementation". Surprisingly it was not possible to find a suitable standard data-model for FRIS R3 – see section “3 Architecture”.

After a long ideation, design and project planning process the implementation project was launched in 2014. During 2014 and 2015 the new FRIS infrastructure was implemented, and FRIS R3 is now being rolled out to all universities (and Pure institutions) in Flanders - deadline for their integration has been set to the beginning of 2017.

In the rest of this paper we present the FRIS R3 development process as inspiration for adaptation of an agile development process even for larger development projects. In section 3, we present the architecture of FRIS R3, again meant as inspiration for anyone working with large scale research information systems. In section 4 we present a number of implementation details, most significantly challenges we have encountered when using CERIF XML for large scale exchange of research information. In section 5 we will recap the results and present a set of recommendations that outline future directions for CERIF XML. Finally, we conclude on the FRIS R3 project so far, and outline future direction for the FRIS R3 project.

2. Process

The initial requirements for the FRIS R3 project did not fully cover all scenarios that the project group envisioned for the system, and there was an expectation that the requirements would change anyway, as part of the ongoing discussions with the project stakeholders. Furthermore, 2014 marked the year where EWI started new contracts with the universities in the region. EWI therefore initiated a new integration project with all the universities. However, a number of details of this project where still undecided. Due to these uncertainties it was decided to structure the implementation project as an agile process backed by a time and material contract instead of a fixed-price contract. This process would allow EWI to delay detailed specification until the time just before it was to be implemented, and more importantly change development priorities as the discussions with the stakeholders evolved.

An agile process requires a great deal of trust and respect between the product owner (EWI) and the development team (Elsevier). This trust is created by having a great degree of transparency into the project activities and outcomes.

A core feature of the agile process is the, in this case monthly, *iteration meeting*. This is a face-to-face meeting in which the development team showcases the outcomes of the preceding *iteration* and plans the subsequent one. The big advantage by having a software deliverable each iteration is that it makes it easy for the stakeholder to gain a meaningful insight in the abstract development process. This also enables the stakeholders to ensure that the requirement or use case interpretation reflected by the implemented functionality is in line with their expectations, or be able to adjust the requirements and priorities while the invested development time is still manageable.

The iteration meetings in conjunction with a high degree of transparency into the development activities, through full access to the development team management tool (Jira), and access to the software quality metrics, combine to hold the development team accountable for the deliverables that have been agreed upon, and enables the stakeholders to verify these deliverables.

The chosen project process was by any measure a success, after a short learning period the EWI project team was fully engaged in the iteration planning and meetings. The benefits of the agile project can be summarised as follows:

- EWI got the functionality they needed, at the time they needed it – not what they thought they needed, at a point in time where it was too late to undo the decision
- EWI re-prioritized their “current needs” several times throughout the process
- EWI where able to scrap functionally they thought they needed, but discovered was irrelevant, and could simply use the “hours” on something else, without having to renegotiate any contract or payment
- EWI was able to see actual progress in form of a running, albeit very limited, system early on, and was able to test increments after each iteration – i.e. FRIS R3 got tangible very early on
- EWI was able to track time used on development (on daily basis, if they wanted to). EWI knew exactly what was going on the development team. This allowed EWI and the development team to openly and honestly prioritize and plan at iteration meetings. Hence, EWI could stop development of functionality if they deemed that it would be come too “expensive” compared to the gain – either based on estimates, or simply based on progress

3. Architecture

FRIS R3 must support a large number of diverse integrations to not only the research institutions, but also to other

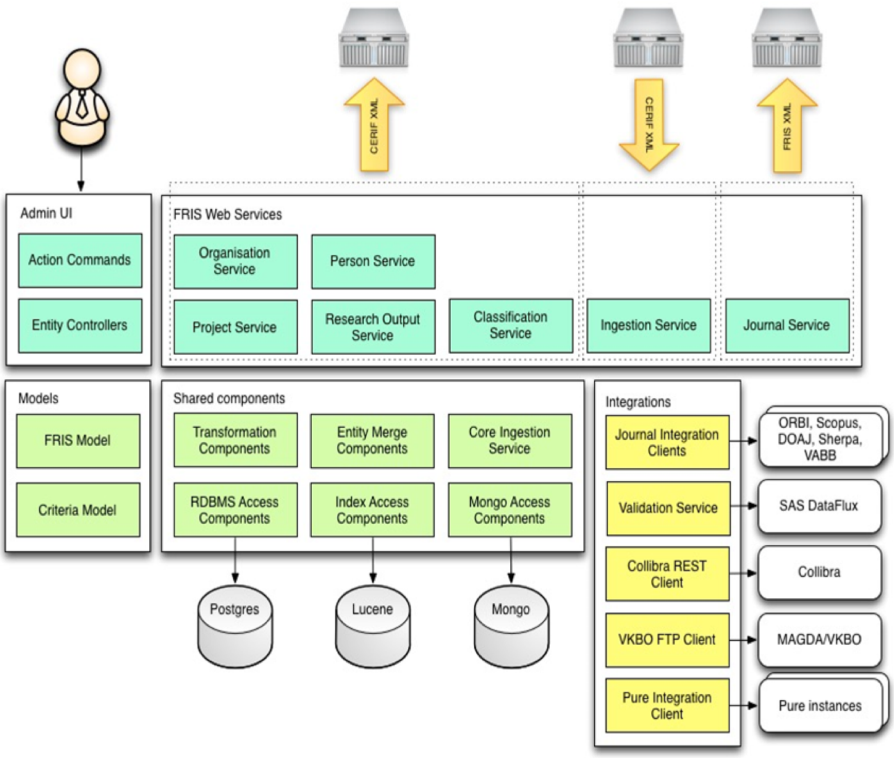

Fig. 1. The FRIS R3 Architecture

data sources, in order to aggregate or enrich the FRIS data set (see section "4 Implementation" for examples of these data sources). In addition, similar diversity exists when exposing FRIS R3 data to users. FRIS R3 is therefore a layered architecture, where the domain and business logic can be kept separate from the technological concerns of the integrations (see figure 1). Each integration point in turn was to be encapsulated so model translations and other technological constraints pertaining to the integration do not leak into the broader system. This type of architecture is also known as a "Ports and Adapters" or "Hexagonal" architecture [7].

Furthermore, the FRIS infrastructure basically comprise a set of *FRIS services* each with a publicly facing web-services, and a set of infrastructure services and storage services. One of the FRIS services is responsible for ingestion of data, whereas the remaining services each are "responsible" for the management of information stored per specific entity type: Researchers, Organisations, Projects, and Research output (see figure 1).

Another aspect that ended up having an architecturally significant impact was the vision of using FRIS entity lookups as part of a data-providers normal research information flow. When a researcher updates a project or research output in the data-provider systems any external collaborators can be looked up in the FRIS web services as well as any internal representations in order to increase data quality. For this scenario to provide an acceptable usability it would need to be extremely fast. Early prototyping showed clearly that a highly normalised relational data model had a very detrimental effect on the median request times for simple entity lookups due to the high cost of the many joins necessary in order to build the full entity representation. The current solution is to synchronise a de-normalised representation of the publicly available data to a document store, and have the FRIS services source data from the document store. This ensures minimal server side processing times when serving public data.

In order to ease both subsequent development and use of the FRIS services a point was made to use applicable de jure or de facto standards wherever possible. Since the FRIS solution is developed in mainly Java this means JAXB, JAX-WS 2.0, SOAP, WS-I Basic Profile 1.1, WSDL 1.1, Oasis WS-SecurityPolicy 1.2 and FastInfoSet for the web services. Focusing on these standards ensure that integrators will be able to use the framework support built in to their implementation language of choice, and spend most of their time on the actual business logic instead of the technological plumbing.

EWI had a strong ambition to evolve a FRIS specific data model, instead of co-opting an existing research information data model. EWI wanted to deviate from the CERIF model because their need was to express a model that in terms of semantics and expression that are much closer to the real domain model than to a database model with highly complex structure. Furthermore, it proved too complex to translate the model for “Funding” in Flanders into a CERIF representation - e.g. modeling Policy Level, Funding Organisation, Type of research (directed or not), money streams, etc. For this reason, the Funding web-service in FRIS is not able to deliver responses in CERIF, as the other web-services are. The FRIS data model was evolved in close collaboration between the EWI project team and the development team. The EWI team would provide the semantic context to be modelled, and the development team would adapt that to the technological constraints. The detailed modelling process followed the iterative, evolutionary structure implied by the Agile [8] project process. This facilitated concentrating on the entities being fleshed out as the use cases involving the entities was being developed. This evolutionary approach assured that the EWI team did not suffer semantic overload, and would be able to redefine concepts as new use cases appeared later in the process, or even redefine concepts if use cases changed later the process. The focus on a data model defined by the FRIS domain has impacted the use of CERIF XML as exchange format between FRIS and the outside world – i.e. support for ternary relationships and participation was added, and an entity for impact reviews was added (the former two concepts are discussed in the next section).

4. Implementation

The FRIS R3 system comprise of a large number of encapsulated adapters, each concerned with the integration towards data-providers (i.e. the participating research institutions), specific data-sources and services users (e.g. clients). Each of these adapters uses one or more *shared core services* in order to modify the underlying data-sets that a specific data-provider is the owner of. That all adapters use the same shared services ensure that behaviour is consistent between different data-sources (and protocols). The list of data-sources below exemplifies how varied the adapters can be (see also figure 1):

- Bulk and incremental import of organisation data from VKBO (Database of companies in Flanders). Periodic FTP transfer and processing of XML files containing both organisation data and associated NACE codes.
- Bulk and incremental import of research institution data sets in CERIF XML from requests to an authenticated SOAP ingestion web service.
- Import of research institution data sets from Pure [9] (bulk & incremental) based on the Pure change stream with configurable semantic mapping.
- Lookup of research entities (organisation, person, journal) in FRIS SOAP web service delivering FRIS model XML as part of the Pure “add external entity flow”.
- Import of curated set of journals from the Orbi [10] OAI-PMH service created by the University of Liège.
- Addition of Scopus IPP/SNIP/SJR [11] metrics from a REST XML service with custom authentication to Journal data sourced from Orbi.
- Addition of VABB [12] metrics (Flemish Academic Bibliographic Database for the Social Sciences and Humanities) from a spreadsheet upload to the journal data sourced from Orbi.
- Addition of SHERPA/RoMEO [13] information from a periodic CSV file download to the journal data sourced from Orbi.

Each of these adapters encapsulate all concerns that are specific to them, whether its pertaining to the interaction protocol or the structural/semantic model transformation. This enables maintaining a coherent core focused on the domain. Other aspects, such as business validation is delegated to a SAS DataFlux [14] instance managed by EWI, and persistence is delegated to relational- (Postgres [15]), indexing- (Lucene [16]), and document- (MongoDB [17]) data stores.

As outlined in section “3 Architecture”, the FRIS model is the result of an analysis driven by the EWI project team through discussions with their domain stakeholders (e.g. select data-providers). In collaboration with the development team the results of the analysis were converted into an appropriate Java model representation. As a part of this design a comprehensive specification of the CERIF XML needed to represent the domain concepts was formulated.

The FRIS domain model is designed for use in a bulk streaming or incremental ingestion model. This means that there are clear boundaries between research domain entities, all associations between entities are by design unidirectional so there is a clear ownership of the association. In addition, all inter-entity associations have been designed to allow single pass, i.e. streaming, processing of an ordered set of organisation -> person -> project -> research output data.

While working with defining and implementing the FRIS domain model, and implementing exchange via CERIF XML a number of observation about CERIF where made. These observations are summarised in the following.

Ambiguous entity boundaries; The CERIF data model is, at its core, a graph model with ambiguous entity boundaries. Not having clear entity boundaries or association owners, makes it hard to decide how much of the object-graph should actually be modified when receiving a CERIF entity representation. For example, if the submitted CERIF entity representation does not include a particular bi-directional association, does that mean that any pre-existing association should be deleted, or that they should be ignored because they haven't been modified?

Read all processing model; The CERIF model is not designed for use in a single-pass processing model. The contained XML elements may appear in any order and the model entity parts may be embedded or stand-alone. This forces any system integrator to use an inefficient processing model, where *all* the exchanged data represented in CERIF XML must be passed into an intermediate form, processed (validated) and then stored. For exchange of large "data-sets" this simply becomes impractical, to the level of using up all available hardware resources (memory).

Complex framework; The CERIF model is less a research information model that can readily be used than a framework where one can specify a research information model. Writing algorithms that can reason about research information stored in CERIF quickly becomes very complex, simply because CERIF is more of a generic framework. It is often fairly easy, for a human, to devise a rule for understanding certain aspects of CERIF, however, translating such a rule to an algorithm in software is more challenging. This challenge becomes impractical when considering all the possible rules needed to reason about data stored in CERIF. Note that this algorithm needs to be able to address a lot of "generic" reasoning as well. In fact, this algorithm needs to be used in almost all interaction with CERIF data, hence, the algorithm, should rather be seen as a "CERIF reasoning engine". The truly devastating fact is that no such readily available reasoning engines exists – every developer needs to build their own. However, since CERIF semantics are fluid, then there is little hope that such "reasoning engines" become an "off the self" software component. Even with well-defined semantics and a rigid governance structure, building such a common "reasoning engine" would be a tremendous achievement. An achievement that would require a lot of commitment from several software developers. In the end it begs the question, why should you spend time modelling your domain in CERIF when you can use a model that expresses the domain directly within its own structure, as opposed to a companion semantic description document? Since CERIF cannot be parsed or understood without extensive custom semantic specification, then the value added by using the model will often even be seen as very low.

Domain Decoupling; A common use-case when modelling research information is the need to identify the specific employment or tenure by which work on a project, publication or patent was performed. This is, however, difficult to to do in CERIF in a precise manner. Using a publication as example - representation of employment would require three binary relationships between CfOrgUnit, CfResPubl and CfPers respectively. However, if there are additional relationships between CfOrgUnit and CfPers, say due to past or future employment, then how can you tell which relationship was relevant at the time the Publication was written (CfResPubl)? Note, that using data/time information stored in the model is inherently error prone, and complex to get right. Furthermore, ensuring that the data/time information you have in your CRIS, or even your HR system is correct, or even at the right detail level, is just a notorious pain point for anyone who have worked with these kinds of systems. To overcome this in FRIS R3 CERIF XML, an *Assignment* "link" entity have been introduced. The Assignment entity is basically a CfOrgUnit_CfPers link entity, but with one notable differences – it has an identifier – and, is often termed a *Ternary Relationship*.

Another example of a common use-case in a CRIS is being able to distinguish between internal and external Persons (or internal and external Organisations). That would be the ability to explicitly model the relations to internally managed persons (where your CRIS manage the assignments), external authors (perhaps you only have their name), external organisations (like project collaborators), or group authors (perhaps this is just a name/string). Hence, a *Participation* "link" entity that encapsulates the relation from a CfProj/CfResPubl/CfResPat to a choice between an Assignment entity, CfPers, CfOrgUnit or group author association was introduced in order to model such concerns.

A more fundamental example is the modelling of the order of authors by use of integers in link entities, rather than utilising the well-known concept of a list. A more general example is that even small, well-known bits of information, say a publication, modelled in CERIF, requires a lot of entities, link entities and classifications. sAll of these are related to *domain decupling*, and combined with the *complex framework* that is CERIF, CERIF just becomes too difficult to use and understand for non-experts, and even developers are challenged.

Lack of tooling awareness; From a purely technical perspective the current CERIF XML XSD was a bit challenging, due to the use of an unbounded xs:choice as a representation for unordered set semantics for all collections of elements. The interpretation of this particular XSD type is awkward when used with most XML frameworks since they will, rightly, lose all type specificity when generating a programming language specific model of the XSD. This places an additional burden on the system integrator to explicitly type check all objects returned by the generated model. If CERIF XML is to be of use for developers, then it must be possible to use common, say open source, frameworks and tools when working with CERIF XML.

5. Recommended future direction for CERIF XML

The CERIF model, although as argued in this paper, is decoupled from the domain and complex, still serves a very important purpose. The CERIF model is a very useful framework that provide common ground and a common vocabulary for experts to discuss the domain of research information. However, as a vessel for exchange of research information, or even a vessel for knowledge exchange in a scope beyond experts, the CERIF model is challenging. If CERIF – in the form of CERIF XML – is to be *the* standard for research information, then CERIF needs to be detached and decupled from the CERIF model. Once detached, then in order to prepare CERIF XML for “industrial/commercial” use, as well as large scale use, then CERIF XML must be propelled into a sustainable format that can be easily utilised by software developers to exchange research information with the use of standard tools. Furthermore, it is crucial to evolve CERIF XML in order to alleviate developers from having to devise complex custom-per-project algorithms to interpret the information at hand.

Streaming is a crucial exchange mechanism; The concept of being able to stream data, one coherent chuck at time, must be addressed. This is the key to dissolve *ambiguous entity boundaries* and leave the *read all processing* model in favour of a simple one-pass processing model. However, in order to truly address *ambiguous entity boundaries* some kind of additional meta-data layer is needed in order to signal deletes, omissions, update, additions, etc. In short, streaming needs to be a handled explicitly in CERIF XML.

Move towards an intuitive domain driven format; Are we really speaking the same language as real-world domain experts (e.g. librarians)? We truly believe that it is time to leave the generic CERIF framework model, and start working on a streamlined exchange format that is domain-near. The aim should be to move away from the *complex framework (the CERIF model)*, to a format is 100% geared towards exchange of research information. This format should rely on the principles of CERIF, but allow a different structure that is self-documenting, and can be used directly without having to understand much other than the domain. Such a format be totally detached from the concept of a database layer – this is an implementation concern. E.g., no more “technical terms” like CfPers, CfOrgUnit, CfResPubl. Introduce real domain concepts instead – E.g. Publication, Person, OrganisationUnit etc. The concept of *CERIF Profiles* is being coined as a way to specify how CERIF should be used in a given context. Maybe such profiles should be taken one step further, and be tangible concepts in the exchange format? Maybe a “publication” should not be a complex graph of CfPers, CfOrgUnit, CfResPubl, CfMeas, CfPaddr, CfEaddr, CfFedId, CfSrv, cfPers_OrgUnit, cfResPubl_ResPubl and the list goes on (and on), but rather be a more concise domain near format that directly addresses the needed meta-data when exchanging “publications”. Granted that by doing so, there may be scenarios that cannot be handled is such a format. However, these scenarios will be edge-cases, and we should allow ourselves to strive towards simplicity, ease of use, and solving the most common use-cases in the context of exchange of research information. We actually truly do understand the domain and its complexity, due to CERIF as our “theoretical” background and our model to understand research information. We should therefore be able to design a format that is a simplified abstraction of the complexity that works in most situations.

We also discussed *ternary relationships* and *participation* in the previous section. These are symptoms of a *decoupling* between the CERIF model and the domain. For example, the lack of ability to directly address which CfPers-CfOrgUnit relationship is relevant in a given context, at a given point in time should not be dismissed lightly. This is a day-to-day real-world concern. We do not claim that our modelling approach is *the* solution – however, the underlying challenge of the domain decoupling and associated symptoms should certainly be discussed more openly. A standard for research information exchange should – ideally – model day-to-day real-world concern in a comprehensible, precise and transparent way.

Cater for standard tools and frameworks; It is absolutely crucial that an exchange format like CERIF XML is matured and tested with popular third party tools and frameworks in order to make sure that CERIF XML can be used efficiently with these tools. Developers must be able to build software for exchange of data using CERIF XML with readily available frameworks (e.g. XML parsers, XML modelling and mapping frameworks, etc.), and be able to realise the exchange without having to write complicated algorithms in order to interpret the data. This will allow developers to build well-performing software efficiently. Hence, *lack of tooling awareness* must be addressed by either actively testing with best practice tools, or by making sure that CERIF XML is well-designed by adhering to best practice XML schema design, and simply allow re-factoring of the CERIF XML schema when developers discover inefficiencies when using common tools. However, caution must be exercised, as it is most likely impossible to accommodate all possible quirks of various tools, but attempts should be made to accommodate standard tool and framework usage whenever possible.

6. Conclusion

As a project, the FRIS R3 project was rather successful – it clearly demonstrated the usefulness of the agile development process, with substantial customer involvement throughout the entire project. The FRIS R3 project have also demonstrated that developing a regional infrastructure for research information can actually be done without having to resort to a complex architecture or technology. However, we are still awaiting the full roll-out to the entire Flemish region, hence, it is too early to state anything about the successfulness of project from the "users" perspective. However, the idea that ownership of data lies with data-providers should mean that FIRS can be seen as a trusted information base, could lead to FRIS R3 becoming a central place to follow up on projects funded by the FWO.

The driver behind the FRIS R3 was to be able to support system-to-system integration where up-to-date, qualitative, complete data can be sent to FRIS as soon as they are created in the systems of the data-providers - without a huge administrative burden at the side of the researcher/research institution. This has indeed been achieved, through best practice software engineering and the use of standards, including CERIF XML.

However, as part of our work with CERIF, specifically CERIF XML, we have encountered various challenges – nothing that could not be solved, however, some solutions did require customisation of the CERIF XML format. More notably, based on our experience with CERIF XML in the FRIS R3 project, but also past projects, we do see a clear need to rethink CERIF XML as an exchange format. In principle, we have challenged the close relation between CERIF XML and the CERIF model – we see a need to move towards a less generic format that is more domain near. This will be at the expense of some of the power of CERIF, but will allow a much simpler exchange format that can archive the most common exchange use-cases, while also being very easy for developers to process and understand.

The next generation of the FRIS portal will be the first client in the new infrastructure, and will become *the* expertise platform for research in Flanders. The next challenge for FRIS R3 will be to look into the way metadata of datasets can be integrated in FRIS as well.

References

1. Department of Economy, Science and Innovation (EWI), http://www.ewi-vlaanderen.be/en
2. Flanders Research Information Space (FRIS), http://www.ewi-vlaanderen.be/en/fris-flanders-research-information-space
3. The FRIS R2 portal, http://www.researchportal.be/en/index.html
4. Research Fund - Flanders (FWO), http://www.fwo.be/en/
5. Resource Description Framework (RDF), http://www.w3.org/RDF/
6. SPARQL, https://www.w3.org/2009/sparql/wiki/Main_Page
7. The Pattern: Ports and Adapters (''Object Structural''), http://alistair.cockburn.us/Hexagonal+architecture
8. Martin Fowler, The New Methodology, http://martinfowler.com/articles/newMethodology.html
9. Pure, https://www.elsevier.com/solutions/pure
10. Open Repository and Bibliography (ORBi), http://orbi.ulg.ac.be
11. Journal Metrics from Elsevier, http://www.journalmetrics.com
12. Flemish Academic Bibliographic Database for the Social Sciences and Humanities (VABB-SHW), https://www.ecoom.be/en/services/vabb/
13. SHERPA/RoMEO, http://www.sherpa.ac.uk/romeo/about.php
14. SAS Data Management, SAS Institute Inc., http://www.sas.com/en_us/software/data-management.html
15. PostgreSQL, http://www.postgresql.org
16. Apache Lucene, https://lucene.apache.org
17. MongoDB, http://www.mongodb.org

18. A. Dimou, L. De Vocht, G. Van Grootel, L. Van Campe, J. Latour, E. Mannens, R. Van de Walle, Visualizing the information of a linked open data enabled research information system, 12th International Conference on Current Research Information Systems (CRIS 2014), volume 33, pp. 245-252, doi:10.1016/j.procs.2014.06.039
19. The CERIF standard, euroCRIS, http://www.eurocris.org/cerif/main-features-cerif
20. B. Joerg, J. Dvořák, T. Vestdam, Streamlining the CERIF XML Data Exchange Format Towards CERIF 2.0, 2012, In 11th International Conference on Current Research Information Systems (CRIS2012), Prague, Czech Republic, June 6-9, 2012