



13th International Conference on Current Research Information Systems, CRIS2016, 9-11 June
2016, Scotland, UK

Application of Kroki Mockup Tool to Implementation of Executable CERIF Specification

Milorad Filipović*, Renata Vaderna, Željko Ivković, Sebastijan Kaplar, Željko Vuković,
Igor Dejanović, Gordana Milosavljević, Dragan Ivanović

University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad, 21000, Serbia

Abstract

The paper presents application of Kroki tool to creation of executable specification of CERIF standard. Kroki (fr. *croquis* – sketch) is an open-source tool that is being developed in order to foster development agility and better communication among team members with different specialties. It can be used for creating of specification of data-driven applications using two different notations (UML based and mockup based) and for generating application over the specification. This paper presents usage of Kroki tool for visualization and further development of CERIF data model, as well as using of Kroki for generating a prototype of CRIS system over CERIF data model. CERIF 1.5 specification was taken from euroCRIS web site and entered into Kroki tool. More than 280 CERIF classes are divided into 31 packages, in order to make CERIF model easier to read and comprehend. Thanks to its mockup and lightweight UML editor, Kroki enables each user, with or without background in information technologies, to understand and change CERIF specification.

© 2016 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the Organizing Committee of CRIS2016.

Keywords: CERIF, model-driven development, CRIS applications

* Corresponding author. Tel.: 381-21-485-2422; *E-mail address:* mfili@uns.ac.rs

1. Introduction

Research management systems take significant role in the development of science (Zimmerman, 2002). Those systems collect, preserve and disseminate data about scientific institutions, scientists and researchers, research projects, published scientific research outputs other relevant data for scientific-research domain. The European Union encourages the development of national research management systems in accordance with the CERIF standard (the Common European Research Information Format - <http://www.eurocris.org/cerif/introduction/>) because the European Union wants to achieve maximum competitiveness of Europe at all levels of research activity. Research management systems compliant with CERIF standard are called CRIS (Current Research Information System). CERIF standard provides a standardized data model. Standardization of those systems data is very important, because it could enable data exchange between those systems and researchers could search research data in various research management systems.

The first version of CERIF was produced in 1991 by European Commission (EC). It had a single-entry focus and a simple record format: the project was an entity with persons, organizations and other information represented as attributes. In 2000 the EC handed over the custodianship of CERIF to euroCRIS non-profit organization (www.eurocris.org) dedicated to the development of research management systems and their interoperability. Since then, CERIF has gone through various development stages and expanded a lot. Eight version of CERIF has been published by euroCRIS organization in the period 2000-2013. CERIF version 1.5 was published in 2013 and this model contains over 280 entities related to scientific-research domain.

By 2015, there are more than 200 euroCRIS institutional, personal and affiliated members from all over the Europe. Those members belong to various scientific fields and have different specialties including different level of technical skills.

Kroki^{11,12} (fr. *croquis* – sketch) is an open-source tool that is being developed in order to foster development agility and better communication among team members with different specialties. It can be used for creating of specification of data-driven applications using two different notations (mockup based and UML based) and for generating application over the specification.

This paper presents usage of Kroki tool for visualization and further development of big CERIF data model which has had very fast expansion in last 15 years, as well as using of Kroki for generating a prototype of CRIS system over CERIF data model.

The paper is structured as follows. Section 2 gives a brief overview of Kroki tool. Section 3 explains creation of CERIF specification and CRIS system prototype using Kroki tool. Section 4 presents related work. Section 5 concludes the paper.

2. Kroki tool

Kroki enables requirements elicitation based on executable prototypes, using the means familiar to the end users - drawing user interface (UI) mockups (see Figure 2). Contrary to the approaches where mockups are created by general-purpose drawing tools and then manually or semi-automatically transformed to formal models¹⁴, mockups created by Kroki are already elements of the UI model. Kroki's mockup editor actually implements the concrete syntax of our EUIS⁶ (Enterprise User Interface Specification) DSL (Domain Specific Language) for specifying UIs of enterprise applications at a high-level of abstraction. Drawing is based on several types of forms and its elements, whose layout and functionality have been defined by our user interface guidelines^{2,3}. Its architecture is presented in Figure 1.

EUIS DSL also has concrete syntax designed to look like a simplified UML (Unified Modeling Language) class diagram notation with stereotypes, which enables modeling in the “classical” way (see Figure 3). Changes made in Kroki's class diagram editor are immediately visible in the mockup editor and vice versa, with automatic layouting performed for newly created elements⁵. This enables each participant (with or without background in information technologies) to use preferred way of development, with changes immediately communicated to all concerned parties.

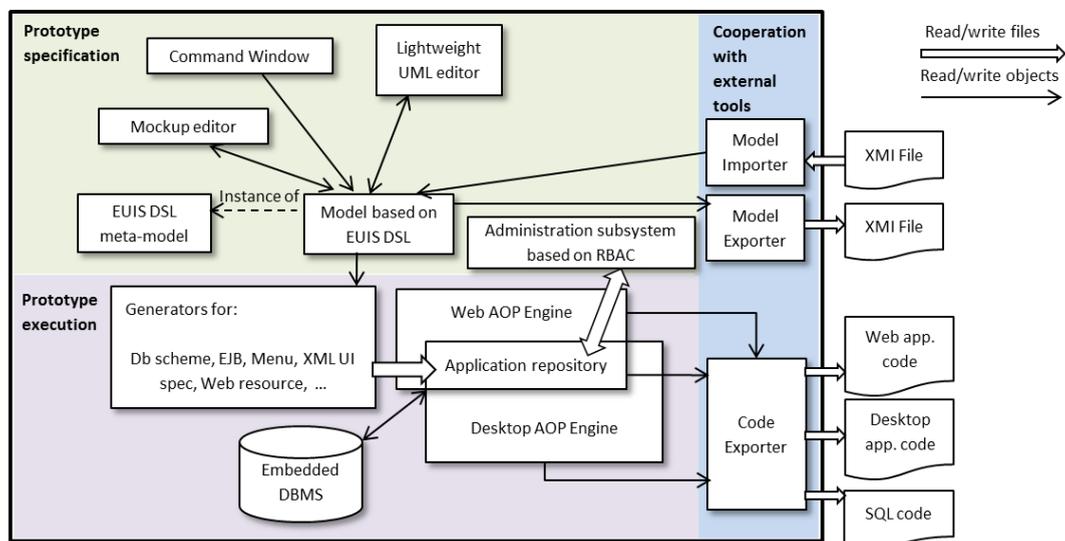


Fig. 1. Kroki tool architecture

Development agility is provided by: (1) Kroki's user interface which enables fast modeling³, (2) code generators which produces a fully-functional, three-tiered application prototype even when minimal specification details are obtained (default code generation action exists for every non-specified functionality)¹, (3) administration subsystem based on RBAC (Role Based Access Control) standard that enables flexible access control policies and dynamic configuration of deployed application based on user roles⁷, and (4) the option of reusing artifacts across development phases, in order to reduce waste of time and effort.

The reuse is supported by exporting class diagrams and application prototypes to general purpose modeling and programming tools, and by importing models from general purpose modeling tools (Figure 1). Thus, a created prototype can be used for requirements elicitation, and can also evolve to the final application using the preferred tool chain (currently supported target language is Java). The application prototype, web or desktop, is generated on top of our aspect-oriented (AOP) frameworks that enable easy integration of generated and hand-written code¹.

Interchange of Kroki models with other modeling tools is based on Eclipse UML2⁸ libraries. Eclipse UML2 is an EMF-based implementation of the UML 2.x⁹ metamodel for the Eclipse platform. Eclipse UML2 provides an API for creating a UML2 diagram using Java classes that contain convenience methods for creating Java objects that map to corresponding UML metamodel elements. The API also provides methods for saving and loading UML2 diagrams to files in XMI format. XMI is a model driven XML Integration framework for defining, interchanging, manipulating and integrating XML data and objects¹⁰.

The application modelled in Kroki can also be directly invoked as desktop or web application, by clicking the buttons on the main toolbar (Figure 8c). Invoking a prototype will generate a middle-tier Java persistence layer using JPA (Java Persistence API) which is used to automatically create database schema deployed in a H2 database management system included in Kroki. Other information contained in mockups is stored in configuration files which are used as inputs to our desktop-based or web-based aspect-oriented frameworks. The frameworks are using these files to dynamically create the remaining elements of the application (forms or pages, menus, links, etc.). The dynamic creation is needed in order to enable role-based access control.

Kroki currently comprises the following tools: mockup editor, lightweight UML class diagram editor, and lightweight business process modeling editor. Business process modeling is still under development.

The mockup editor (see Figure 2) consists of: mockup drawing area, UI component palettes, and property editor panels used for setting the properties of mockup components. Property editor is divided into two tabs: the first contains basic settings, which can be adjusted by non-programmers; the second contains advanced settings, intended

for advanced users and developers. Unless the advanced properties are set, defaults are used, so that prototype execution is always possible.

3. Application of Kroki tool to CERIF model

The CERIF 1.5 specification was taken from euroCRIS web site and entered into Kroki tool. More than 280 CERIF classes are divided into 31 packages, in order to make CERIF model easier to read and comprehend. The specification is available in Kroki mockup editor and Kroki lightweight UML editor (presented in Figures 2 and 3, respectively).

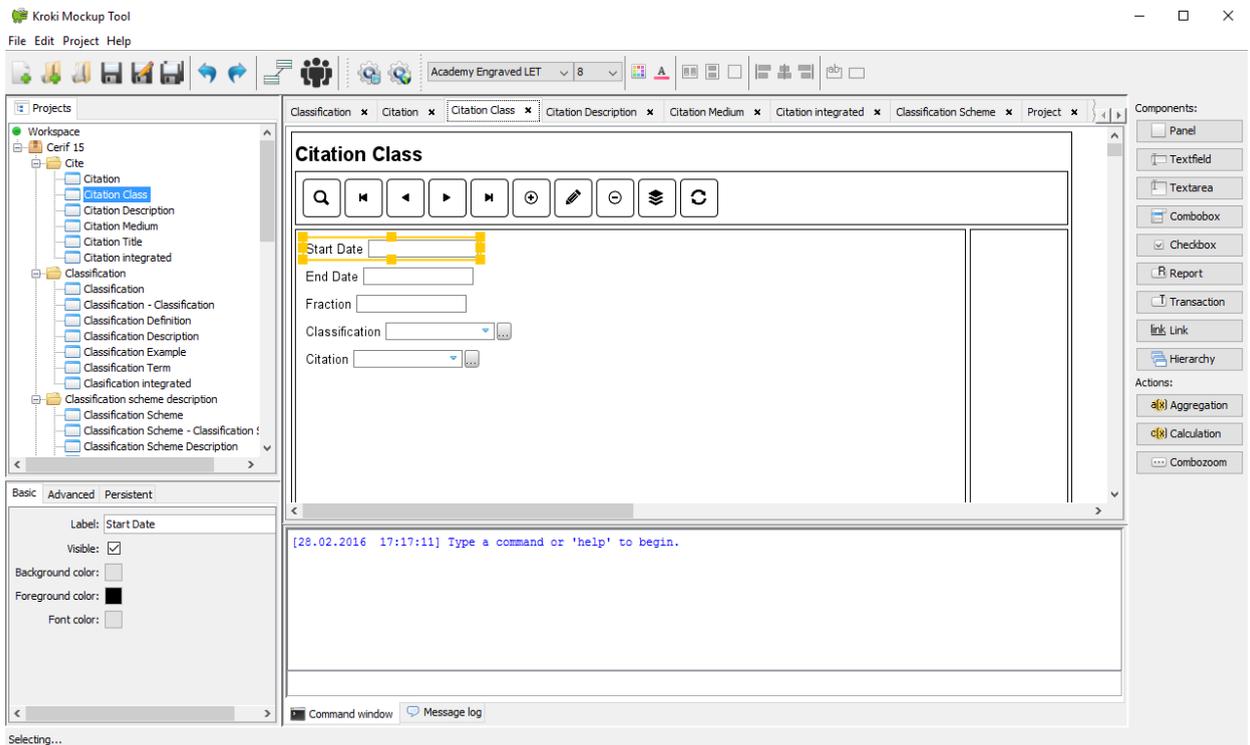


Fig. 2. CERIF 1.5 specification in Kroki mockup editor

A developer can enhance or update CERIF specification using both mentioned Kroki tools. Since Kroki provides automatic execution of the specified application, developer has to decide how to map CERIF classes to user interface of the generated CRIS application. Available course-grained components of the user interface are standard panel (Figure 4 and Figure 5) and parent-child panel (Figure 6).

Standard panel is designed to display data and all available operations so the user can choose a data item and invoke an operation on it without memorizing commands. Standard operations common to all entities are represented by buttons/icons at the top of the form, while specific operations (if they exist) are represented by links/buttons at the right hand side. Operations common to all entities include search (query by form), display, addition, update, and removal. Specific operations include complex data processing procedures associated with the given entity (transactions), invocation of related (*next*) screen forms, and invocation of reports. Standard panels can also be used as building elements of parent-child panels.

A parent-child form is used for data that have hierarchical structure, where each element in the hierarchy is modelled as an entity in the database. Each element in the hierarchy is represented by a standard panel, where a panel at the n -th hierarchy level filters its content according to the selected data item at the level $n-1$. The hierarchy

must have at least two levels. The position of a panel in the hierarchy is defined by hierarchical association between it and the container panel (see Figure 7).

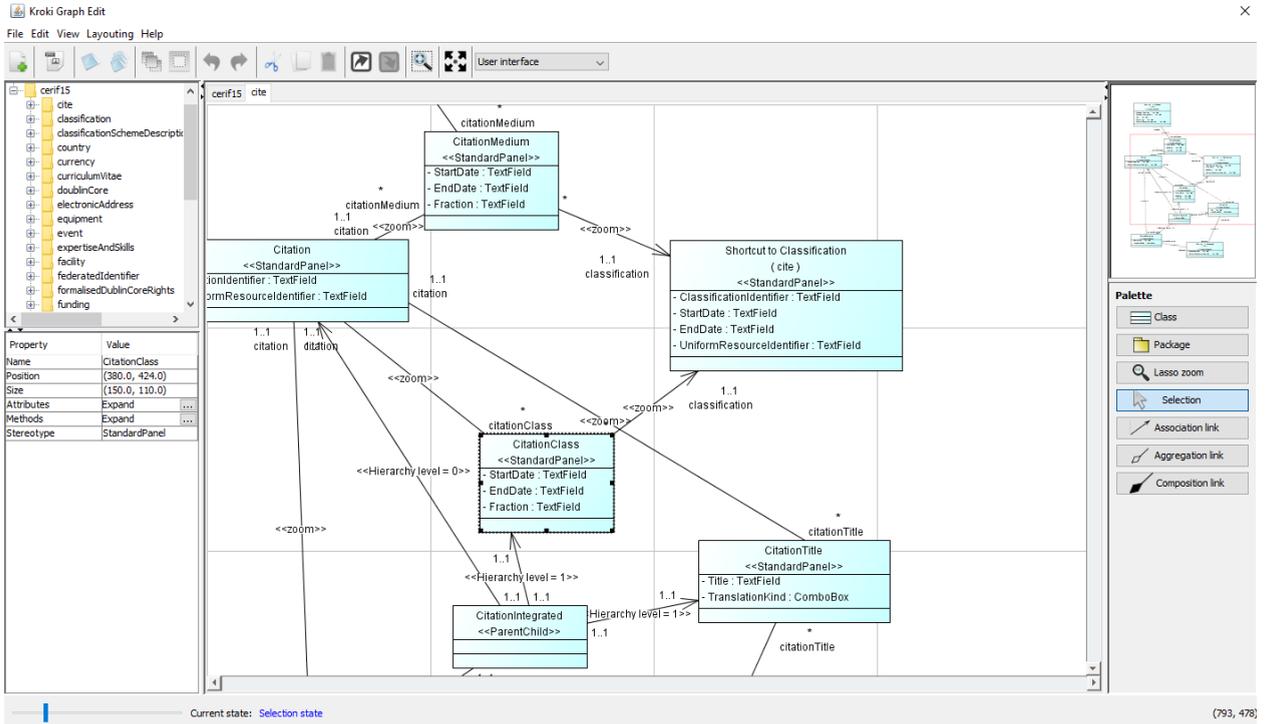


Fig. 3. CERIF 1.5 specification in Kroki lightweight UML editor

The specified CERIF application can be invoked as a desktop or web application, by clicking the buttons on the main toolbar (Figure 8 c). Invoking a prototype will generate a middle-tier Java persistence layer using JPA which is used to automatically create database schema. Database schema can be deployed in a chosen DBMS, or in a H2 DBMS which is included in Kroki.

If it is needed to change the appearance or functionality of generated prototype of CRIS application, there are two ways to achieve this. If minor changes are needed, CRIS application code can be exported in Eclipse project (Figure 8 a). Aspects¹⁷ can be used to weave the generated and manually written code, without need to directly change the generated code¹. This way, subsequent code generation couldn't damage manual changes.

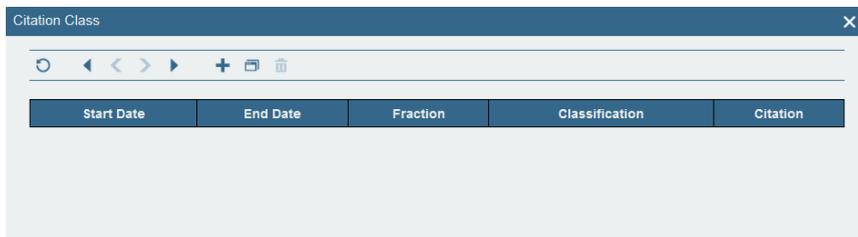


Fig. 4. An example of generated standard panel in view mode for CitationClass entity

Fig. 5. An example of generated standard panel in add mode for CitationClass entity

Fig. 6. An example of generated parent-child panel for Currency entity

The CERIF specification can also be exported to general-purpose modeling tool, as presented in Figure 8 b).

A developed CERIF 1.5 specification, generated application, and associated Eclipse project are available at: <https://github.com/KROKIteam/KROKI-CERIF-Model>. Kroki installation is available at: www.kroki-mde.net/download/. A short movie that explains Kroki usage is available at: <http://youtu.be/r2eQr11bzA>.

4. Related work

The paper presents an extension of our previous work on automated development of CERIF-compliant research management system². The main difference with the previous work is usage of our Kroki tool that enables users, with or without background in information technologies, to understand and change CERIF specification, in order to automatically produce a prototype of a CRIS application.

Approaches and tools similar to our Kroki tools are discussed below.

Störrle¹⁶ presents approach to use mockups as a base to generate the code that helps user to investigate both the graphical layout of the application and the behavior of the user interface elements. It is still not a complete solution and the implementation of other aspects of the application (persistence layer, for example) requires manual intervention.

Rivero et al.¹⁴ propose approach where mockups created by general purpose mockup tools are parsed, manually enriched with tags that add semantics to its elements and then translated to an abstract UI model that is further translated to presentation and navigation models of a web application. The content model can be derived as well, but it is not always precise enough since derivation is based on heuristics. After a set of transformation, developers can obtain an executable prototype that user can try out, similar to our solution.

Forward et al.¹⁵ present UI prototype generator named UIGU. The UIGU is based on set of heuristics for transforming Uml models to a usable UI.

All presented solutions are using heuristics to produce some kind of model (UI model, presentation model, data model, etc.) from the mockup or vice versa. This process can be prone to errors and usually requires manual intervention to improve archived results.

5. Conclusion

The paper presented application of Kroki tool to creation of executable specification of CERIF standard. CERIF 1.5 specification was taken from euroCRIS web site and entered into Kroki tool. More than 280 CERIF classes are divided into 31 packages, in order to make CERIF model easier to read and comprehend. Thanks to its mockup and lightweight UML editor, Kroki enables each user, with or without background in information technologies, to understand and change big CERIF specification. Also, Kroki enables automatic code generation of a prototype of CRIS application.

Further development will be related to manual fine-tuning of the generated CRIS prototype in order to improve its look and usability. Also, we plan to incorporate the changes proposed in CERIF 1.6 specification.

References

1. Filipović M, Kaplar S, Vaderna R, Ivković Ž, Milosavljević G, Dejanović I. Aspect-Oriented Engines for Kroki Models Execution. 5th International Conference on Information Society Technology and Management (ICIST) 2015. p. 502 – 507
2. Milosavljević G, Ivanović D, Surla D, Milosavljević B. Automated construction of the user interface for a CERIF-compliant research management system, *The Electronic Library* 2011. Vol. 29. Iss: 5. p. 565 – 588
3. Milosavljević G, Filipović M, Marsenić V, Pejaković D, Dejanović I. Kroki: A Mockup-based Tool for Participatory Development of Business Applications, IEEE 12th International Conference on Intelligent Software Methodologies, Tools, Techniques 2013.
4. Zimmerman E. CRIS-Cross: Current Research Information Systems at a Crossroads, Proceedings of the 6th International Conference on Current Research Information Systems 2002, p. 11-20
5. Vaderna R, Milosavljević G, Dejanović I. Graph Layout Algorithms and Libraries: Overview and Improvements. 5th International Conference on Information Society Technology and Management (ICIST) 2015. p. 55 – 60
6. Perišić B, Milosavljević G, Dejanović I, Milosavljević B. UML Profile for Specifying User Interfaces of Business Applications, *Computer Science and Information Systems* 2011, Vol. 8, No. 2, p. 405-426.
7. Kaplar S, Filipović M, Milosavljević G, Sladić G. Kroki Administration Subsystem Based on RBAC Standard and Aspects. 5th International Conference on Information Society Technology and Management (ICIST) 2015. p. 61 – 66
8. Eclipse MDT/UML2, <http://wiki.eclipse.org/MDT-UML2>
9. Unified Modeling Language (UML), http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML
10. MOF 2 XMI Mapping (XMI®), http://www.omg.org/technology/documents/modeling_spec_catalog.htm#XMI
11. Kroki source code. <https://github.com/KROKIteam/KROKI-mockup-tool>
12. Kroki tool web site. www.kroki-mde.net/
13. Freemarker java template engine. <http://freemarker.org/>
14. Rivero JM, Grigera J, Rossi G, Robles Luna E, Koch N. Towards Agile Model-Driven Web Engineering. IS Olympics: Information Systems in a Diverse World, Lecture Notes in Business Information Processing 2012. Springer, p. 142-155
15. Forward A, Badreddin O, Lethbridge TC, Solano J. Model-driven rapid prototyping with Umple. *Software Practice & Experience* 2012, Vol.42 Issue 7, p. 781-797
16. Störrle H. Model driven development of user interface prototypes: an integrated approach, Proceedings of the Fourth European Conference on Software Architecture 2010. p 261-268
17. AspectJ Project. <https://eclipse.org/aspectj/>