Conference on ENTERprise Information Systems / International Conference on Project Management / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN /HCist 2015 October 7-9, 2015

An EAI based integration solution for science and research outcomes information management

Fernando Rosa Sequeira[a], Rafael Z. Frantz[b], Iryna Yevseyeva[c], Michael T. M. Emmerich[d], Vitor Basto-Fernandes[e]

[a]*School of Technology and Management, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal*
[b]*UNIJUÍ University, Department of Exact Sciences and Engineering, Ijuí, Brazil*
[c]*Cyber Security Research Institute, School of Computing Science, Newcastle University, NE1 7RU, Newcastle-upon-Tyne, UK*
[d]*Multicriteria Optimization, Design, and Analytics Group, LIACS, Leiden University, Niels Bohrweg 1, 2333-CA Leiden, The Netherlands*
[e]*School of Technology and Management, Computer Science and Communications Research Centre, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal*

Abstract

In this paper we present an Enterprise Application Integration (EAI) based proposal for research outcomes information management. The proposal is contextualized in terms of national and international science and research outcomes information management, corresponding supporting information systems and ecosystems. Information systems interoperability problems, approaches, technologies and tools are presented and applied to the research outcomes information management case. A business and technological perspective is provided, including the conceptual analysis and modelling, an integration solution based in a Domain-Specific Language (DSL) and the orchestration engine to execute the proposed solution. For illustrative purposes, the role and information system needs of a research unit is assumed as the representative case.

Peer-review under responsibility of SciKA -Association for Promotion and Dissemination of Scientific Knowledge.

Keywords: Enterprise Application Integration; Domain-Specific Language; Information Management.

1. Introduction

Organizations rely on information systems and software applications to support their business activities. Frequently, these applications are legacy systems, packages purchased from third parties, or developed internally to solve a particular problem. This usually results in heterogeneous software ecosystems, which are composed of applications that were not usually designed taking integration into account. Integration is necessary, chiefly because it allows to reuse two or more applications to support new business processes, or because the current business processes have to be optimised by interacting with other applications within the software ecosystem. Enterprise Application Integration (EAI) provides methodologies and tools to design and implement integration solutions. The goal of an EAI solution is to keep a number of applications' data in synchrony or to develop new functionality on top of them, so that applications do not have to be changed and are not disturbed by the integration solution. (Frantz, Enterprise Application Integration - An Easy-to-Maintain Model-Driven Engineering Approach, 2012)

As stated by Gregor Hohpe and Bobby Woolf (Hohpe & Woolf, 2003) application integration can be done in four different ways: File Transfer, Shared Database, Remote Procedure Invocation, and by Messaging. In File Transfer, each application produces files of shared data for others to consume, and consumes files that others have produced; in Shared Database the applications store the data they wish to share in a common database; in Remote Procedure Invocation one application exposes some of its functionalities in such a way that other applications can access them as a remote procedure, thus the communication occurs in real-time and synchronously; finally, in Messaging, each

application connects to a common messaging system, exchanges data and invokes behaviour using messages; since an application can read messages from that common messaging system in a later time after they have been published by another application, the communication is asynchronous; applications only must agree on a channel and also on the message format.

Knowing that each style has its advantages and disadvantages (Hohpe & Woolf, 2003), our approach will be done using the Messaging style. Integration solutions that are based on messaging allow for asynchronous communication between applications, which makes them loosely coupled, improving several important system quality attributes such as scalability and availability. (Frantz, Corchuelo, Roos-Frantz, & Sawicki)

The work presented here addresses an information system integration problem targeted for the analysis of national science and technology outcomes domain. We follow an EAI approach to deal with the diversity of sources and information systems available in this area, taking into account public domain national policies and known long-term decisions concerning research outcomes information management.

The amount and variety of repositories and data sources available with researchers CVs, scientific publications, research projects, scientific events, advanced training supervision and science dissemination actions, represents a huge and valuable asset for researchers, research units, Higher Education Institutions (HEI), entrepreneurs and innovation oriented businesses, to explore in an ad-hoc fashion, for many different purposes. However, those who need to have an organized, systematic and accurate perspective of research outcomes at national, research units and individual researchers levels, have difficulties on collecting, harmonize and summarize this information.

In this paper we propose an EAI based solution for national research outcome analysis at individual, research units and institutional level. The paper is structured as follows: Section 2 presents the problem domain, identifying the relevant national and international players, their software and recommendations. Section 3 introduces information systems integration from a technical and software engineering perspective. Technical approaches and tools are introduced, presented and explained, with particular emphasis on EAI. In Section 4, we present the software ecosystem perspective, its components and role in supporting the information management of the scientific and technology national system. Section 5 presents our EAI based integration solution, and finally in Section 6 conclusions and future work are provided.

2. National science and research outcomes information systems

The science and research outcomes information management process consists in the collection, structuring, processing and storage of information about researchers, publications, citations, projects, and other metadata about research activities and actors. Science and Technology Foundation (FCT) is the Portuguese national agency and authority for research promotion, funding and evaluation.

Although national research agencies and authorities are special observers of scientific and technical production at the national scale, other institutions also need to follow this type of information for research planning, follow up, benchmarking, etc. Among these institutions are Higher Education Institutions (HEI), research institutes and other national and regional governmental agencies, and non-governmental industry and society actors.

Several initiatives have been developed to provide features and support for the needs of such information consumers. The most relevant initiatives worth to mention at the national level are the FCTSIG and DeGois® software platforms, representing the researchers national CV repository. While the former has a simple user interaction approach, the latter has highly structured data models and advanced features for researchers CV information management. Additionally, other initiatives at the national level took place targeting bibliometric data collection and science based indicator analysis. Having an exclusive bibliographic and bibliometrics approach, these tools did not attract enough attention from the science and technology institutions, mainly due to their narrow scope for science and research outcomes analysis.

Several HEI have also developed internal systems for science and research outcomes management following their own data models, taxonomies and description syntaxes. National science and research institutions and corresponding information systems face nowadays the challenge of interoperating and exchange this type of information, in the scope of the science and technology national and international information ecosystem, for general and specific observation purposes. Among the international ecosystem components we can point out journals and conference publications repositories such as SCOPUS® and Web of Science®, international researchers information

repositories such as ORCID®, and several (less institutional) research oriented social networks. This global ecosystem is devoted to support research outcomes general information, lacks data harmonization and consistent identity management mechanisms, and raises severe difficulties for research outcomes analysis and evaluation at individual, institutional and national levels.

For the sake of simplicity and summarised description, without loss of generality, we assume in this work the role and perspective of a research unit actor. A research units needs, in a regular base (at least annually), to follow and assess its researchers activities and outcomes, whose CVs, activities, research outcomes are registered and updated in national funding agencies software platforms and international research production repositories.

In this paper we present an EAI solution proposal for science and research outcomes information system integration, from the perspective of a research centre (or a similar research organization entity), in charge of pursuing and managing science and research activities.

3. Guaraná technology

In the last years many proposals and tools have emerged to support the design and implementation of integration solutions. Hohpe and Woolf (Hohpe & Woolf, 2003) documented several integration patterns found in the domain of enterprise application integration. Camel, Spring Integration, Mule and Guaraná range among the state-of-the-art integration tools in this domain, that provide support for integration patterns. Camel provides a fluent API (Fowler, 2010) that software engineers can use programmatically or by the means of a graphical editor. In both cases, the integration solution is implemented using a Java, Scala, or XML Spring-based configuration files. Spring Integration was built on top of the Spring Framework container, and provides a command-query API (Fowler, 2010). This tool can be used programmatically or by the means of a graphical editor. Integration solutions are implemented using either Java code or an XML Spring-based configuration file. The architecture of Mule got inspiration from the concept of Enterprise Service Bus. Software engineers count on a command-query API (Fowler, 2010) to use this tool programmatically, or a workbench to design and implement integration solutions using a graphical editor. Integration solutions are implemented using either Java code or an XML Spring-based configuration file. In earlier versions, Mule supported a limited range of integration patterns; version 3.0 resulted in a complete re-design whose focus was on supporting the majority of integration patterns. Different from the previous technologies, Guaraná got inspiration from the Model-Driven Engineering discipline (Schmidt, 2006), shifting the focus from the source code to models. Models are abstractions that allow software engineers to focus on the relevant aspects of a software system while ignoring details that are irrelevant. Behind this discipline is the idea to raise the level of abstraction of the overall development process, to capture systems as a collection of reusable models, to separate business logic descriptions from a particular platform implementation, and to automate the implementation phase.

The design of integration solutions is supported by a domain-specific language (DSL) included in Guaraná technology. This language provides constructors with a visual concrete syntax, which allows modelling integration solutions in a diagrammatical form. The resulting models are platform-independent and can be automatically transformed into executable code. Guaraná also provides a software development kit (SDK), which is composed of a Java command-query API to support the implementation of the abstractions in the domain-specific language, and a runtime system that can be used to execute the integration solution. Note that the Java code plus the runtime system are not enough to implement integration solutions; it is also necessary a number of adapters. Adapters are a piece of software used for interacting with the applications being integrated (Frantz, Reina Quintero, & Corchuelo, A domain-specific language to designate enterprise application integration solutions, 2011). Figure 1 provides an overall picture of the main components in the Guaraná technology.

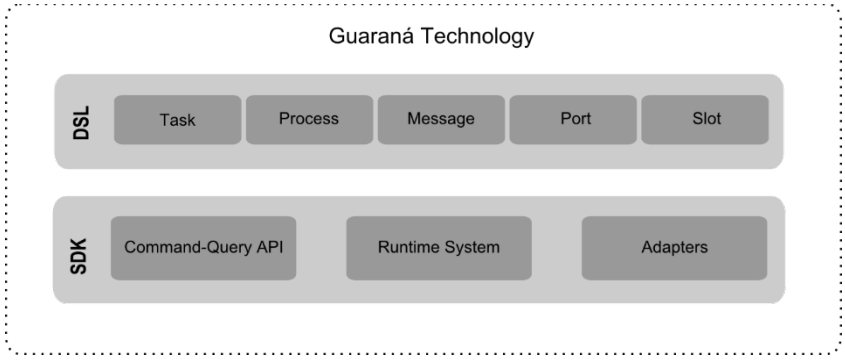

Figure 1. Guaraná overall picture.

1.1. Concrete Syntax

Guaraná DSL has an easy-to-learn and intuitive graphical notation. Through every constructor in the language, flows messages. In Guaraná, a message is an abstraction of a piece of information that is exchanged and transformed across an integration solution. It is composed of a header, a body, and one or more attachments. The header includes custom properties and frequently the following pre-defined properties: message identifier, correlation identifier, sequence size, sequence number, return address, expiration date, and message priority. The body holds the payload data, whose type is defined by the template parameter in the message class. Attachments allow messages to carry extra pieces of data associated with the payload, e.g., an image or an e-mail message. Figure 2 illustrates the concrete syntax of the main constructors in Guaraná.

Notation	Constructor	Notation	Constructor
	Resource		Solicitor Port
	Integration Process		Responder Port
	Entry Port		Task
	Exit Port		Slot

Figure 2. Guaraná constructors (from (Frantz, Sawicki, Roos-Frantz, Yevseyeva, & Emmerich, 2015)).

Task: Represents an atomic operation that can be executed on messages, such as split, aggregate, translate, chop, filter, correlate, merge, resequence, replicate, dispatch, enrich, slim, promote, demote, and delay. Roughly speaking, a task may have one or more inputs from which it receives messages, and one or more outputs by means of which messages depart. Depending on the kind of operation, a task may be stateless or stateful. In a stateless task, the completion of its operation does not depend on previous or future messages; contrarily, the operation of a stateful task depends on previous or future messages to be completed, such as the case of the aggregator task, which has to collect the different correlated inbound messages to produce a single outbound message. The vast majority of tasks in Guaraná technology are stateless.

Slot: A buffer connecting an output of one task to the input of another task aiming at messages to be processed asynchronously by tasks. A slot can follow different policies to serve messages to tasks, such as a priority-based output or a first-come, first-served. If a priority is defined in the message, slots follow the former policy; otherwise, the latter policy is adopted. In this paper, it is assumed that messages have no priority and the slot serves them in a first-come, first-served policy.

Port: Abstracts away from the details required to interact with resources within the software ecosystem. Roughly speaking, by means of a port it is possible to establish read, write, solicit, and respond communication operations with the resources being integrated.

Integration Process: Contains integration logic that executes transformation, routing, modification, and time-related operations over messages. An integration process is composed of ports that allow it to communicate with the resources being integrated, slots and a set of tasks to specify the integration logic.

Conceptually, an integration solution aggregates one or more integration processes through which messages flow and are processed asynchronously. The integration flow is actually implemented as a Pipe and Filter architecture, in which the pipes are implemented by Slots and the filters are implemented by Tasks. Every task realizes an integration pattern, (Hohpe & Woolf, 2003) and its execution depends on the availability of messages in all slots connected to its inputs. Slots are key constructors to enable asynchrony in an integration solution, thus messages are stored on them until they can be read by the next task in the integration flow. Messages do not appear in Figure 2, because they are not part of the conceptual model, they only exist and flow in the constructed and running integration solution (Frantz, Sawicki, Roos-Frantz, Yevseyeva, & Emmerich, 2015).

4. Software Ecosystem

Our proposal involves the interaction with four applications external to the integration solution: "Local Research Unit Characterisation", "PlataformaDeGois®", "CMS Application" and “Web of Science®”. As the name suggests, "Local Research Unit Characterization" refers to the file system in the computer running the integration process, containing information about the research unit for this specific solution. "PlataformaDeGois®" represents in our solution the CVs repository of Portuguese researchers. (Plataforma Degóis) Among other things, "PlataformaDeGois®" can be queried for a researcher curriculum in XML format.

The XML format CV of a researcher, returned by “PlataformaDeGois®” through SOAP Web Services and RESTful Web Services requests and responses, follows the schema presented in Figure 3. "CMS Application" refers to a Content Management System application (Joomla®), where the integration solution resulting HTML files should be stored.

DeGois® CV XML structure covers a wide range of research activities and outcomes, which are essential for a complete and comprehensive analysis of researchers, research units and institutional results analysis and assessment.

Other sources of information are used in our integration solution to be merged and to enrich information coming from “PlataformaDeGois®”. These sources include information collected from the Web of Science® concerning Journal Citation Reports® (JCR) indexes, ORCID® platform for researchers international IDs management, SCOPUS® to collect citations data for researchers specific papers, etc. For the sake of clarity and restricted space we do not represent all these sources in the current integration solution.

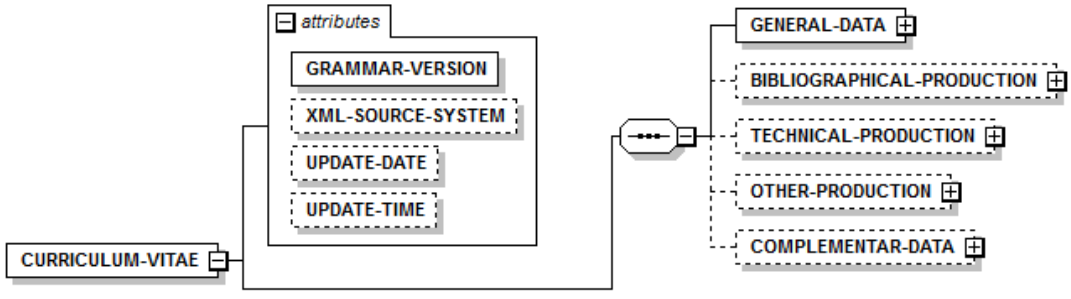

Figure 3. DeGois® CV XML Schema

"Researchers" XML document available at “Local Research Unit Characterisation” represents the main initial input for our integration solution. The structure of this document covers information about researchers, their institutional affiliations, research groups memberships, etc., as shown in Figure 4.

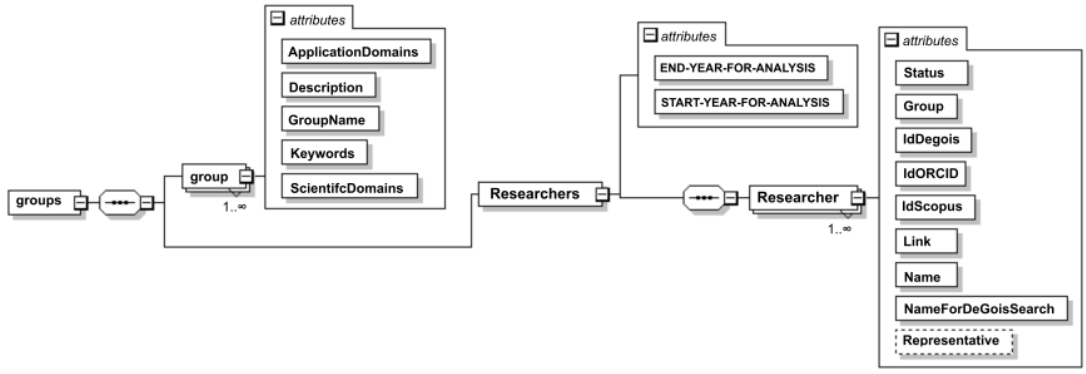

Figure 4. Researchers.xml Schema

5. Integration Solution

The proposed integration solution modelled using Guaraná DSL is introduced in Figure 5. An input XML document named Investigadores.xml, c.f. Figure 4, is present in "Local Research Unit Characterization" and stores the initial and main input for the integration solution. This document contains information about researchers (tag "Researchers"), namely their status (attribute "Status"), the research group he or she belongs (attribute "Group"), the researcher name to do a search his/her CV in “PlataformaDeGois®” (attribute "NameForDeGoisSearch"), etc. This document is updated with data from "PlataformaDeGois®"(e.g. researcher CV last update at "PlataformaDeGois®"), and its information about researchers feeds and triggers all transformations taking place inside the integration solution. The output of the integration solution consists in a set of HTML documents that are forwarded to a "Joomla®" CMS instance in the form of "Joomla®" articles.

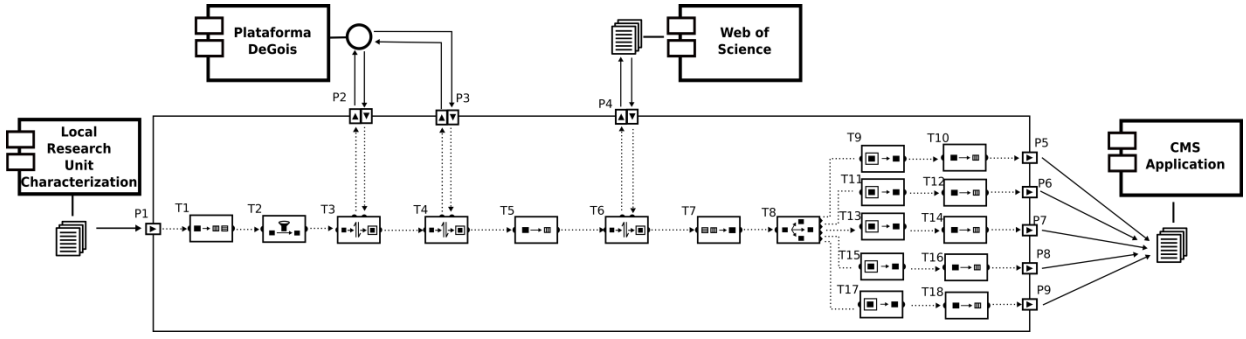

Figure 5. Science and research outcomes information management integration solution

The workflow begins at entry P1, which periodically loads “Researcher.xml”. The first task (task T1) splits the data obtained from P1 in parts to be further processed, each part corresponding to a researcher name. Second task (T2) filters out the researchers that do not hold the "Effective" status (Status != "Effective"). Then, remaining messages in the solution are replicated at task T3, one copy is used to build a "DeGois®" researcher IDquery to be forwarded to "PlataformaDeGois®" by Solicitor Port P2, the other copy remains at task T3 for being enriched with information returned from "PlataformaDeGois®". At this moment the integration solution has the researcher DeGoisId needed to fetch his/her CV in the following query to be sent to “PlataformaDeGois®”.

After getting the researcher DeGoisId, the solution will ask for the Curricula. Message coming from Task T3 is replicated at task T4; like before, one copy will be used to get the researchers CV in XML from "PlataformaDeGois®" (corresponding to his/her DeGoisId) through Solicitor Port P3; the other copy remains at task T4 and will be enriched with the result returned from port P3.

Task T5 does a message schema change for the message to hold new information coming from "Web of Science". Specifically, the XML attribute "Factor-of-Impact-JCR" is added to the XML element "Article-Detailing".

Task T6 retrieves information from "Web of Science®" related to JCR journal impact factor, to be associated, by the means of ISSNIds, to researchers bibliographic production, using attribute "Factor-of-Impact-JCR".

From now on, information about researchers does not need to be treated individually. Task T7 re-unifies the messages with information about each researcher into a single message, for research unit granularity processing. Task T8 replicates this message into five copies, that will be used to produce five different HTML output documents, containing research unit scale indicators in a per category basis (projects, papers, organized events, awards, advanced training).

Tasks T9, T11, T13, T15, and T17 (Slimmer tasks) perform message cleansing, preserving only information related to each of the specific research indicator to be calculated/processed.

Finally, tasks T10, T12, T14, T16, and T18 perform message transformation, more precisely, transformation of XML represented data into HTML documents, corresponding to the five different categories of research indicators. The output of these tasks (five HTML documents) is forwarded through ports P5-P9 to "CMS Application" in the form of CMS articles, and immediately made accessible by the ("Joomla®") CMS instance.

6. Conclusions and future work

This paper presented an innovative integration solution targeted for science and research outcomes information management. Research outcomes information management at research units, institutional and national levels were presented, as well as the overall research outcomes management ecosystems. Information producers, consumers, sources and platforms were addressed with focus on interoperability problems and information systems integration complexity.

EAI integration approach was selected as one the most suitable integration approaches to deal with integration problems in a systematic, cost effective, high-level abstraction fashion. Among state of the art EAI technologies, Guaraná was chosen due to its advantages with respect to some integration solutions quality attributes, with emphasis on independency.

A Guaraná based DSL solution was designed and explained in detail. The integration solution functional requirements were studied, analysed and the corresponding integration solution built accordingly.

7. References

Fowler, M. (2010). *Domain-Specific Languages.* Addison-Wesley.

Frantz, R. Z. (2012). *Enterprise Application Integration - An Easy-to-Maintain Model-Driven Engineering Approach.* Sevilha: The Distributed Group ETSI Informática.

Frantz, R. Z. (n.d.). *Home Page.* Retrieved 04 10, 2015, from TDG Research Group: http://www.tdg-seville.info/rzfrantz/sdk-architecture-1-4-0

Frantz, R. Z., & Corchuelo, R. (2012). A Software Development Kit to Implement Integration Solutions. *SAC 2012.* Riva del Garda, Italy.

Frantz, R. Z., Corchuelo, R., Roos-Frantz, F., & Sawicki, S. *Enterprise Application Integration - Modelling Enterprise Application Integration Solutions.*

Frantz, R. Z., Reina Quintero, A. M., & Corchuelo, R. (2011). A domain-specific language to designate enterprise application integration solutions. *International Journal of Cooperative Information Systems* , 143-176.

Frantz, R. Z., Sawicki, S., Roos-Frantz, F., Yevseyeva, I., & Emmerich, M. (2015). On Using Markov Decision Process to Model Integration Solutions for Disparate Resources in Software Ecosystems. Barcelona: ICEIS - International Conference on Enterprise Information Systems.

Hohpe, G., & Woolf, B. (2003). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.* Addison-Wesley.
Messerschmitt, D., & Szyperski, C. A. (2003). *Software Ecosystem: Understanding an Indispensable Technology and Industry.* MIT Press.
Plataforma Degóis. (n.d.). (FCT - Fundação para a Ciência e a Tecnologia) Retrieved 04 16, 2015, from Plataforma DeGóis - Plataforma Nacional de Ciência e Tecnologia: http://degois.pt/index.jsp?id=1
Schmidt, D. C. (2006). Guest Editor's Introduction: Model-Driven Engineering. *IEEE Computer , 39* (2), pp. 25-31.