Proceedings of
the 28th International Business Information Management Association Conference

9-10 November 2016
Seville, Spain

ISBN: 978-0-9860419-8-3

Vision 2020:

Innovation Management, Development Sustainability, and Competitive Economic Growth

Editor

Khalid S. Soliman

International Business Information Management Association (IBIMA)

RoaML for Data Integration

Strîmbei Cătălin
linus@uaic.ro

Dospinescu Octavian
doctav@uaic.ro

Străinu Roxana-Marina
poparoxi@yahoo.com

Nistor Alexandra
alexandra.anichitoaei@yahoo.com

Alexandru Ioan Cuza University, Iasi, Romania
Faculty of Economics and Business Administration
Dept. of Business Information Systems

Abstract

In response to SoaML initiative to build UML models for SOAP services, in a previous paper it was proposed a new modelling framework, RoaML, as a specific profile to model RESTful web services. In this paper we will describe an extension of RoaML specialized on integration of REST Data Services, taking into consideration that Web Data Integration using the Restful principles could became a significant stream of the Data Integration field.

Keywords: RoaML, SoaML, Data Integration, Databases, Service Oriented Architecture, RESTful Web Services, SOA, UML

1. Introduction – Context and Problem Relevance

In an scientific article entitled "Modeling Web Services with RoaML" (Strîmbei & Olaru, Modeling Web Services with RoaML, 2015) it was noted a resurrection of developer community interest for Web Data Services due to the large proliferation of RESTfull Web Services structural principles. In this context we discussed about modelling of service oriented architecture topic and we proposed a new UML profile, RoaML, targeting RESTfull services. The RoaML profile aims to be complementary to the existing SoaML initiative focused on SOAP flavor of web services.

The market specialists (Dospinescu & Florea, 2016) consider that the need for mobility, the need to communicate over long distances and access to desired information anytime, anywhere were factors that determined fulminant evolution of mobile technology. The specialized literature (Hurbean & Fotache, 2013) also confirms that the IT industry is revolving around the build-out and adoption of a new platform, characterized by mobility, cloud-based application and service delivery, and value-

generating overlays of social business and pervasive analytics. In this paper, we will approach the Data Integration topic in the context of Web Services. In as much as the integration is one of the most notable achievements of web services, Web *Data* Services (Saleh, Kulczycki, & Blake, 2009), (Dospinescu & Perca, 2013) seems to be a “natural” starting point for some data integration strategy based on web technologies. Therefore, our goal is to make some contributions related to a more specific research question: How to use Web (Data) Services as Data Integration approach (among other initiatives)?
In this matter, our proposal takes into consideration the assumption of using Web Data Services (WDS) as a common/logical denominator layer in a Data Integration Architecture (DIA).
Our interest for web-centric data integrations solutions comes from the issues revealed by our experience and investigations on existing Data Integration Tools (often labeled as “modern” ETL Tools) regarding:

- non-standardized operational/proceeding methods to define and integrate candidate data sources;
- some obvious limitations concerning data integration modelling;
- leaning to a more procedural or operational approach (for practical reasons) with missing or very difficult/cumbersome declarative approach.

2. Considerations on the Potential of Web Services for Data Integration – Literature Review

The Data Integration (DI) terminology includes many kinds of definitions. According to (Doan, Halevy, & Ives, 2012), the DI process integrates disparate data into a common/unified view of information. Also, the specialized literature (Giordano, 2011) consider that DI combine multiple and distinct data sources into a common/unified data store (data system) in a consistent manner. Some authors (Reeve, 2013) remark that the data integration consists in a consolidation of data “in motion”.

Considering Data Integration *challenges,* (Doan, Halevy, & Ives, 2012), (Lee, 2015) describe as most relevant: locating the data on the web, importing the data into a structured workspace and combining data from multiple sources.
Other authors (Chaudhuri, Dayal, & Narasayya, 2011), (Sherman, 2015) enumerate as the most important challenges regarding the process of data integration: queries, the number of sources, the heterogeneity of data, autonomy, ETL Platforms and Tools approaches.
According to (Reeve, 2013), there are four distinct *types* of data integration: (i) batch data integration, (ii) real-time data integration, (iii) big data integration and (iv) data virtualization.
The literature (Giordano, 2011) describes very analytical the layered/vertical Data Integration *Architectures*:

- Conceptual Data Integration Models;
- Logical Data Integration Models;
- Physical Data Integration Models;
- Implementation Platforms/Technologies;
- Development Platforms/Technologies.

Also, we have detailed descriptions (Giordano, 2011) for functional/horizontal Data Integration *Architectures:*

- Extract/Publish Data sources;
- Transformations/Conceptual integrations;
- Load Integrated Data in Data Marts or Data Warehouses.

In this context, our work hypothesis refers to the usefulness of RoaML for Data Integration field. Our investigation takes into consideration the potential impact of RESTFull Web Services for DI architectural layers as conceptual, logical and physical models.

3. RoaML for Data Integration – Proposed Solution and Design

3.1 Existing Roa-ML profiles

SOAML (as SOA Modelling Language) represents an initiative aimed to define an UML extension (or profile) with a metamodel that extends UML to support the some essential modelling requirements for SOA. In (Strîmbei & Olaru, RoaML: AN INNOVATIVE APPROACH ON MODELING WEB SERVICES, 2015) there was proposed a specific approach named “RoaML - Rest Oriented Architecture Modelling Language” as a “step brother (or sister)” of the already established SoaML proposal, but focusing on RESTFull Web Services whose modelling do not fit well in SoaML context. RoaML Proposal define an UML Profile based on three metamodels (figure 1): the *Domain* metamodel and the Rest metamodel are joined into the *Rest Domain* metamodel in order to get a more business-oriented perspective while preserving the simplicity of REST architecture.

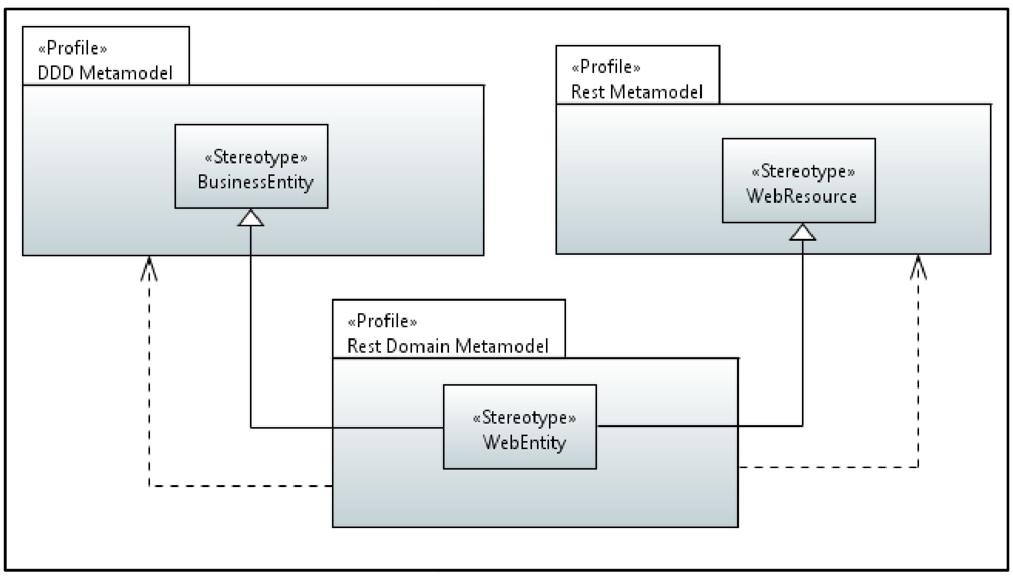

Figure 1: RoaML Metamodels and their relationships

Being a REST-focused proposal, RoaML tries to formalize RESTFull principles and concepts by using some meaningful UML stereotypes (figure 2) (Strîmbei & Olaru, Modeling Web Services with

RoaML, 2015). The "fundamental" concept outlined by RoaML is the *WebResource*, alongside of *RestService*, but the most comprehensive modelling buildings should be based on the principle of "resource" located or provided by some kind of "service".

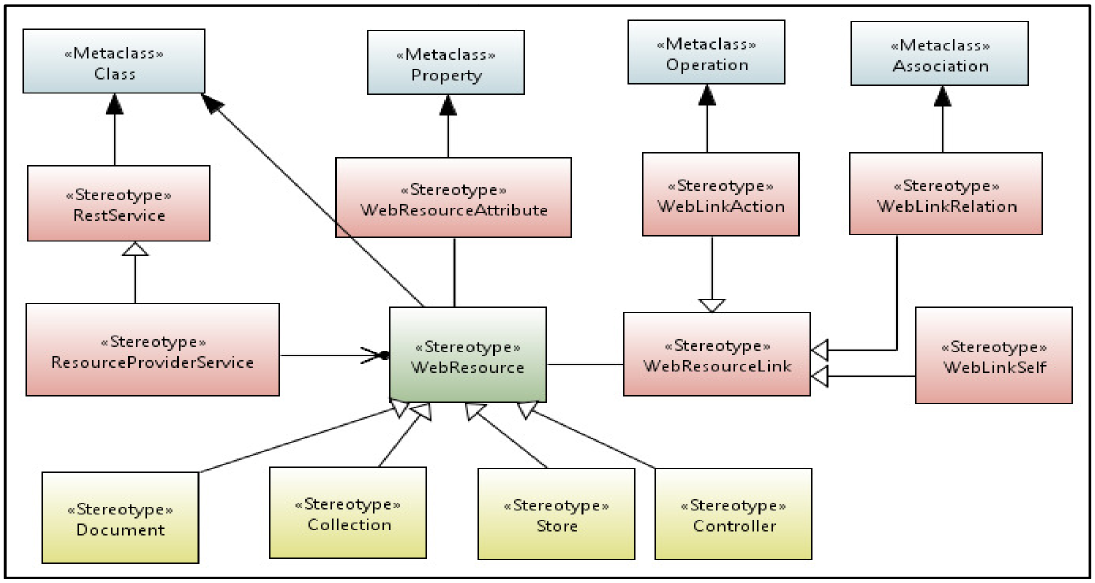

Figure 2: Core REST Metamodel stereotypes and their relationships

The first major construction built on RoaML basic Metamodel is a business-oriented framework named *Rest Domain Metamodel* whose main objective is to prepare *domain* models ready to be easily embodied into REST architectures.

3.2 RoaML4DI: Abstract Layer Profile

RoaML's Rest Domain Metamodel proposes the concept of *Web Entity* in order to merge the domain entity pattern (Evans, 2003) with the web *resource* principle of RESTFull architectures.

This way, the *WebEntity* stereotype took the central spot of our view and all its surrounding properties comes from *domain* context but prepared as web-accessible (figure 3): *WebEntityUID* (specialized from *Rest Metamodel: WebLinkSelf*), *WebEntityOperation* (specialized from *Rest Metamodel: WebLinkAction*), *WebEntityRelation* (specialized from *Rest Metamodel: WebLinkRelation*). This way, a REST System modeler could describe a REST application using the generic stereotypes of *Rest Metamodel*, or could adapt an already designed *domain model* using the more business specialized framework *Rest Domain Metamodel.*

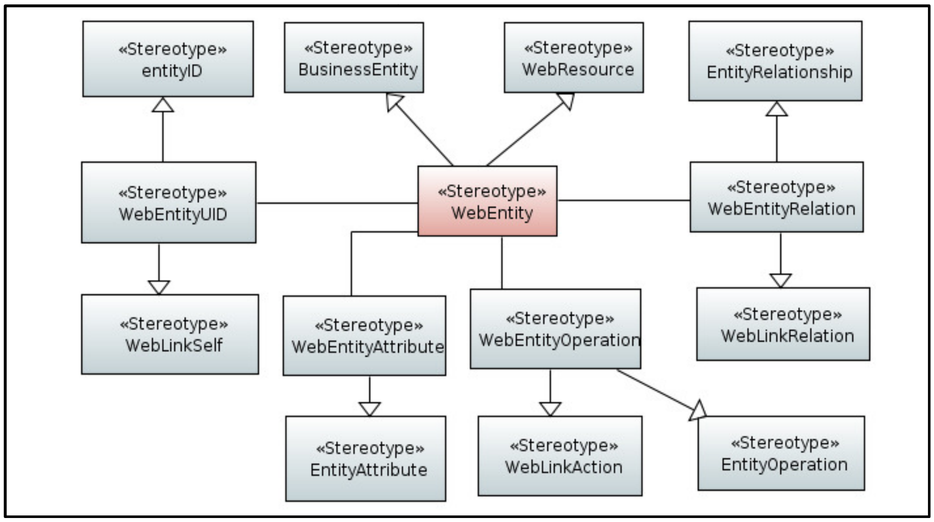

Figure 3: Rest Domain Metamodel of RoaML

We think that the proposed design of Rest Domain Metamodel sets the premises to model the business *domain* as integrated and unified view of data very useful in the field of data integration. To achieve this (next) level of abstracting we added some new metadata resources stereotypes (meta-resources) as described in the following paragraphs.

3.3 RoaML and Data Integration Architecture

As we discussed previously, the basic RoaML Rest Domain metamodel could be used to describe generic entities that could be specialized into:

- business entities to be accessed from initial data sources to be integrated;
- business entities that could result from the integration process, e.g. as dimensions or facts for OLAP services.

Our intention is to extend RoaML profile to be able to cover in a more complete manner the layered Data Integration architecture (vertical perspective) as of:

- *Abstract layer* with domain entities to describe source and target data structures;
- *Operational layer* with all web services involved by specific Data Integration processing;
- *Physical Layer* with the low-level infrastructure services or components that implement the services from the operational layer.

And yet another kind of components could be essential and necessary to this generic architecture: Meta-data services.

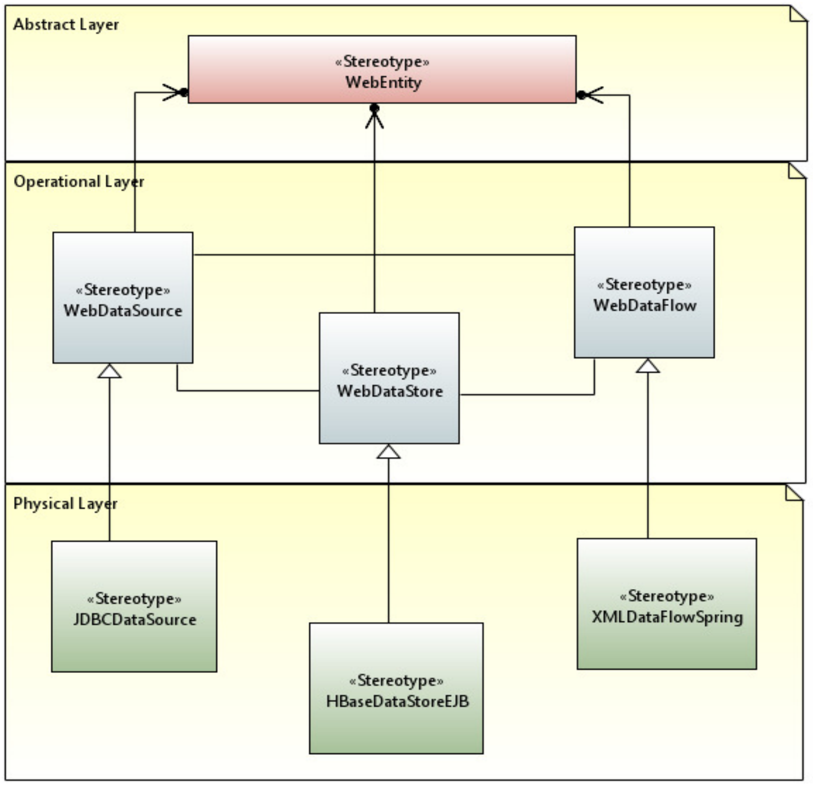

Figure 4: RoaML for Layered Data Integration Architecture

3.4 RoaML4DI: Operational Layer Profile

The *Operational layer* will contain some specific stereotypes for the web services that will form the functional (horizontal) DI Architecture:

- *DataSource REST Service* stereotype for the web services with the role of exposing and accessing (publish) original data entities;
- *DataFlow REST Service* stereotype for the web services with the role of loading and transforming entities from their original sources;
- *DataStore REST Service* stereotype for the web services with the role of persisting integrated business entities coming from DataFlows.

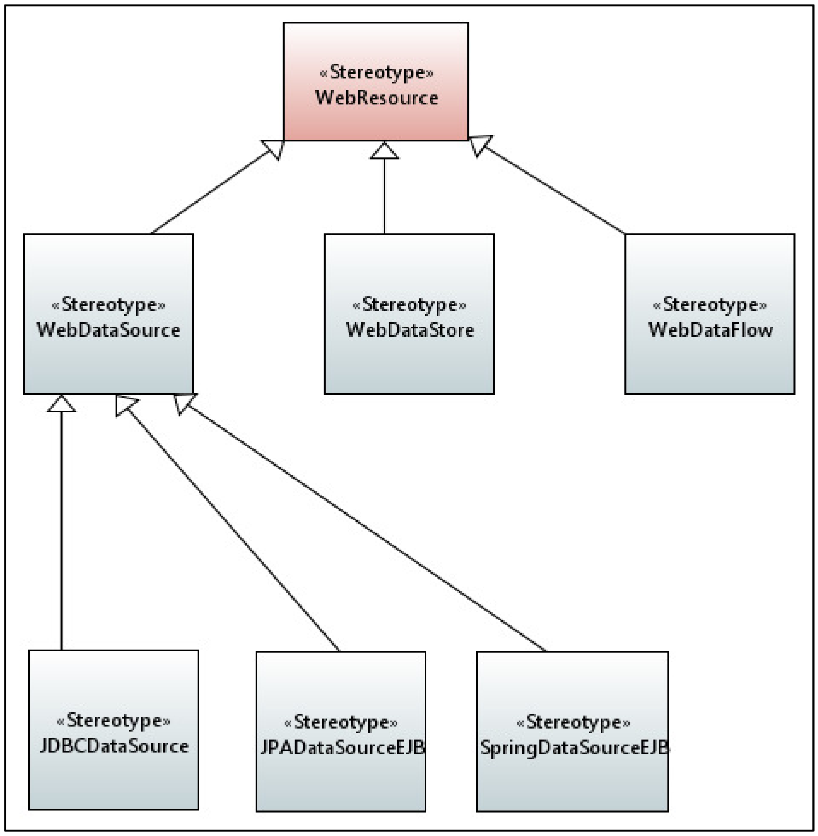

Figure 5: RoaML Operational Profile for Data Integration

The WebDataSource services will act as data providers, or data adapters or data wrappers for the original data sources that could be implemented using infrastructure and middleware technologies, e.g:

- JDBCDataSources to access SQL-relational databases;
- JPADataSourceEJB to access SQL and NoSQL databases and even structured (or non-structured) flat files using CVS, XLS, XML or JSON formats.

The restful services classified as *WebDataFlows* will produce and maintain (e.g, update as cross check data and change data capture) the new integrated business domain containing the new integrated web entities modelled by the same Domain Rest framework of RoaML.

The *WebDataStores* could represents web services to lead data flows' output into DataMart, DataWarehouse or BigData specific persistent data collections.

This way, our final goal is to create a common modelling context for all components of some Data Integration Architecture that could be implemented by a wide range of technologies but orchestrated using solely Web Data Services means.

3. Business Domain Modelling with RoaML-DI: a practical case study – experimental evaluation

Our intentions aim to use our experimental stereotypes grouped in RoaML DI profile to model the complexity of a real data integration business process as the one in the figure 6.

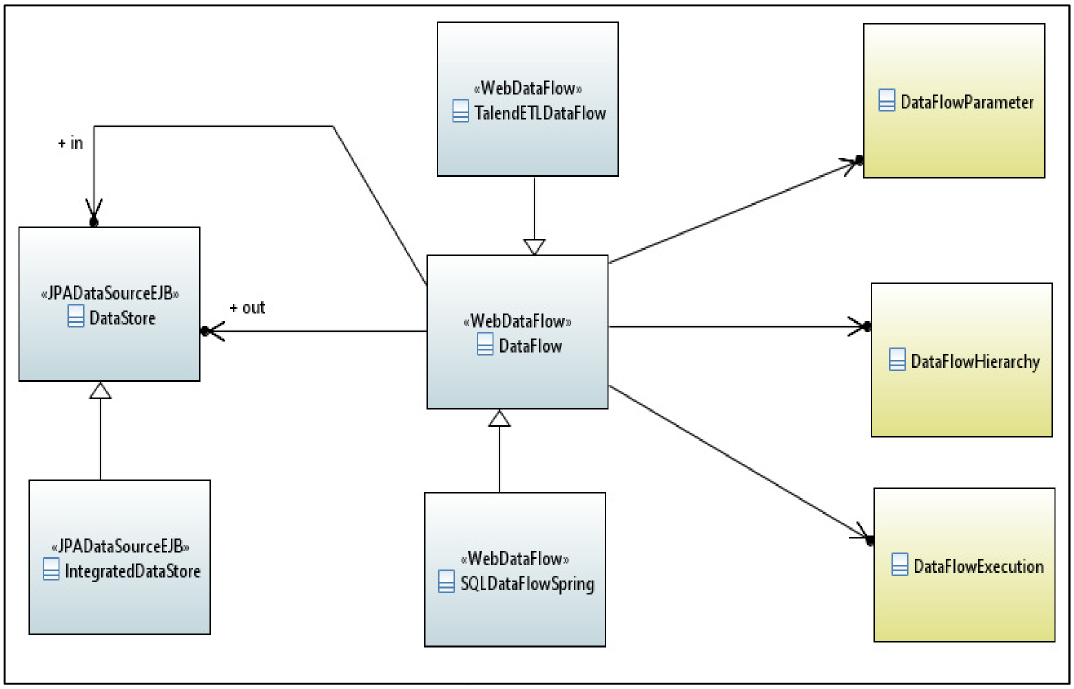

Figure 6: A specific data model built using RoaML for DI framework

Here we assumed a data integration context based on JPA-wrapped original data sources (as diverse SQL database systems, e.g. Oracle-based or PostgreSQL-based, or NoSQL database systems, e.g. Cassandra-based or MongoDB-based) processed by specific Data Flow services implemented by Java Spring Components (consuming SQL data) or implemented by some more generic and procedural processing tasks of Talend ETL Platform.These Data Flows expose their integration (or orchestration) endpoints by using the same REST conventional data structures (described as WebEntitied through Domain Rest RoaML profile) in order to be assembled in more complex hierarchical and executable structures.

5. RoaML4DI: Future concerns

A more specialized set of stereotypes need to be added into this starting point for a new methodology of Data Integration modelling based on web services approach. This need comes from an extensive list of concerns we think that will challenge (if they are not already challenging) the DI field, as:

- Large Scale Structured Databases (e.g. public sources: Wikipedia, Google Scholar, Researchgate etc.);
- Integrated Data Visualisation;
- Data Integration Democracy: user powered data flows;
- Social Media Data Integration;
- BigData Structural Models (NoSQL Data Models);
- BigData Large Repositories (clustering, parallel processing and caching);
- Microservices-based Architectures;
- Cloud-based integrated data repositories;
- Linked Data Integration (Semantic Web, RDFs, SparQL);
- Web Semantic and Data Ontologies.

In order to achieve a more practical value of such modelling framework, our future concern consists in going on with further investigations:

- to include some valuable Business Process Modelling concepts to extend the expressiveness of WebDataFlows;
- to build a truly MDA top-down framework with CIM level based on existing RoaML profile (of course including Data Integration extension) and with PIM level tested on some Java-based mature platforms (JEE and Spring) but containing extension point to existing ETL/DI open source platforms, such as Talend, JasperSoft or MuleSoft and JBoss Data Virtualization, and OCL transforming rules formalized using tools such as Eclipse-based PapyrusUML plus transformation plugins as Acceleo, ATL and QVTO.

Acknowledgments

„This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-II-RU-TE-2014-4-0748”.

References

Chaudhuri, S., Dayal, U., & Narasayya, V. (2011). An overview of business intelligence technology. *Communications of ACM, 54*(8), 88-98.

Doan, A., Halevy, A., & Ives, Z. (2012). *Principles of Data Integration.* Morgan Kaufmann.

Dospinescu, N., & Florea, D. (2016). Smartphone Brands Design and Buying Decision. *Ecoforum, 5*, 139-150.

Dospinescu, O., & Perca, M. (2013). Web Services in Mobile Applications. *Informatica Economica*, 17-26. doi:10.12948/issn14531305/17.2.2013.02

Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software.* Boston: Addison Wesley.
Giordano, A. (2011). *Data Integration Blueprint and Modeling: Techniques for a Scalable and Sustainable Architecture.* IBM Press Pearson.
Hurbean, L., & Fotache, D. (2013). Mobile Technology: Binding Social and Cloud into a New Enterprise. *Informatica Economica, 17*(2), 73-83. doi:10.12948/issn14531305/17.2.2013.06
Lee, K. (2015). Survey: Data integration challenges costing healthcare agencies billions. *Search Data Management*. Retrieved from http://searchhealthit.techtarget.com/blog/Health-IT-Pulse/Survey-Data-integration-challenges-costing-healthcare-agencies-billions
Reeve, A. (2013). *Managing Data in Motion Data Integration Best Practice Techniques and Technologies.* Elsevier Inc., Morgan Kaufmann.
Saleh, I., Kulczycki, G., & Blake, B. (2009). Demystifying Data-Centric Web Services. *Web-Scale Workflow*, 86-90.
Sherman, R. (2015). Data integration platforms take users beyond ETL software. *Search Data Management*. Retrieved from http://searchdatamanagement.techtarget.com/feature/Data-integration-platforms-take-users-beyond-ETL-software
Strîmbei, C., & Olaru, G. (2015). Modeling Web Services with RoaML. *Economy Informatics, 15*(1), 26-34.
Strîmbei, C., & Olaru, G. (2015). RoaML: AN INNOVATIVE APPROACH ON MODELING WEB SERVICES. *Proceedings of the IE 2015 International Conference.* Bucharest: ASE Bucuresti.