

# Design of Rivalize and Software Development to Convert RDB to RDF

Syed Umar, P. Gayathri, Ch. Anil, N. Priya and R. Srikanth

**Abstract** In the storage of databases, the relational databases are used utmost for storing data in very large amount which should be with good and high integrated security to safe the data semantics. So, the Semantic Web services facilitate a good solution which allows the data to be used safely and restage with various applications. But the limitation is to transform relational data to Resource Description Infrastructure/Framework for the business applications which will mostly depend on the relational model and RDF. So in this paper, we are concerning on to develop a new methodology R2R which is a RDB–RDF with integration approach contains architecture, algorithms, and mapping which will enable the RDF-based RW access to enrich the various characteristics of RDB. So the R2R ingression technology helps to align the RDM and RDF to handle the blank nodes, a clear description is shown in below topics.

**Keywords** RDF · RDM · Ingression methodology · PoC

## 1 Introduction

Software engineering field is constantly changing. New paradigms, programming languages and tools to obtain a huge wave suddenly and then they are as fast as they were unconscious. A little time to sacrifice, because it is important for the success

---

S. Umar (✉) · Ch. Anil · N. Priya  
Department of Computer Science Engineering, MLR Institute of Technology,  
Hyderabad, India  
e-mail: umar332@gmail.com

P. Gayathri  
Department of Information Technology, Gokaraju Rangaraju Institute of Engineering  
and Technology, Hyderabad, India

R. Srikanth  
Department of Information Technology, Institute of Aeronautical Engineering,  
Hyderabad, India

of the new instruments, in both industry and research environment. Unlike short-term means, a body of knowledge provided by the companies usually still proceeds slowly and sometimes decades record for the largest database (RDB). This will inevitably lead to a conflict that must work as another generation of applications in the data. Old systems are reliable to easily adjust the views of their database applications or online transactions, and only when needed to change. Older systems are often important to the sun and should be too. They value for many organizations, but the material, maintenance, and further development are particularly difficult and costly.

This paper describes how RDB–RDF access [1] promotes the current transitional system, the Semantic Web application. Access offers a top semantics of relational database that exists. This allows RDF-based read and write relational databases. Based on the concept image that fills the gap between RDF and the relational model, the intermediary translates the Semantic Web applications, SQL flights. This enables the applications based on relational RDF and works with the same data and makes better use of the benefits of proven technologies such as database application's flexibility, security, and trade. The contribution of this thesis is to analyze how they are used successfully in access after the analysis of the development environment based on Eclipse software animation Rivalize [2] SOFAS [3] validated software platform to analyze cases and service-oriented distributed rations. We present a case where the focus of RDF and RDB is unsatisfactory and vision access.

## 2 Related Survey: Rivalize and SOFAS

### 2.1 *Rivalize*

In this case, a description of the two platforms to work together with the diagnostic program provides access information. The first option, Rivalize animated mannequin, considered legacy systems, although the bank scalable way of the past, which is deployed to analyze the development of software systems. Rivalize animation is based on the concept of historical license database (RHDB) [4]. It is used as a set of Eclipse plug-ins and combines data from multiple software repositories, such as version control, issue tracking, mailing list, and the combination of different databases, but listen and touch a variety of development components and software analysis. Animation is typical for old system's Rivalize: Even if the platform is still in active use, it is difficult to adapt to changing needs and the latest designs. Closely associated with the Eclipse makes it difficult to adapt and reuse tools and animation Rivalize algorithms in a new environment, such as within a service-oriented. The RHDB technology is also based on a traditional knowledge database. Therefore, it is difficult to exchange information with other sources of external data RHDB for communication with the local market, but not in general, and our

members do not have a special event of the center, which can be dereferenced online.

## 2.2 *SOFAS*

Evolvyzer allows us to integrate and analyze various aspects of software development and success. It recognizes, however, that the great potential that is easily accessible and modular learning is not limited to the platforms and languages and requires no special tools to install and configure. On this basis, this place is what led to the “Evaluation Software as a service” [5]. Get easy access to a wide range of analytical tools and Web services that use a number of suppliers to implement this concept easier and flexible platform bed (Software Analysis Services) [3]. SOFAS-based RESTful architecture [6] enables the determination which is based on the principle of a simple but powerful analysis software and the use of state resources; the transfer of the Web architecture consists of three main parts: malformation SERVITI software analysis, analysis of broker software, and software analysis and encyclopedias. The service exposes the functionality and data from Web services interface standard is silent. Broker analysis software features the interface between the provider and the services and users. [7] This directory contains all of the analytical services that are related to the specific study of the taxonomy of software.

## 3 Case Studies of Bridge Data Analysis Software

Stimulation of the incident, we will introduce the case when it comes to communication Rivalize tween animation and use banks. They must be removed from the case in two directions, that is, the bank Rivalize animation and vice versa. First Rivalize animation data for hundreds of software and systems, introduced during the last year [8]. Banks should be able to access the platform to get without having to re-enter. To do this, read RDF-based animation Rivalize access to the database. Importers Rivalize animation equipment second raw data and version control systems like CVS and SVN core history [8]. The data is important because Rivalize animation can be existing tools used to build it. However, it requires write access to the database Rivalize RDF-based animation. Banks finally implemented and extensible software metrics to calculate the data. Again, the RDF data model, but the appropriate ratio to achieve animation Rivalize database schema. KA by subscription rights RHDB necessary information, the animation Rivalize indicators [9]. Efforts must use reflecting bridge and Bank animation Rivalize RDB mapping RDF data access read and write relative. Because the approach is currently limited to read-only services issues, we have developed without access to read and write. The rest of this section, we present the ontology model is relational data model

animation Rivalize banks and many entries using the conceptual difference between two data models. We will continue to the issues raised in this discussion [10].

### ***3.1 Data Analysis Software System for Animation Rivalize***

Animation program consists of several parts Details Rivalize, covering many aspects of the software engineering field. In this study, we will focus on the elements of the historical source, one of the most important audits. Review prepared a special version of the file. Personally, it means that software developers to edit the file, and must, version control system. The latest news of the change (i.e., the commit message from the client), as well as additional information such as the number of lines involved. Release is an important step in the life cycle of a software system [11]. It is often referred to the name and the code of the latest versions of files and data to the picture. New or experimental features and bug fixes often in branch. If the code is stable and is connected to the housing. To be a significant source model must meet state assessment system. The release is usually only guaranteed, so we can improve the initial part of the code and model analysis based initialize [12]. Redesign class if it meets the highest levels of Java, C #, etc. Class members, including the properties and methods of the numbers. Classes can, attributes and methods summarized in the machine.

Relations between organizations such as the origin of the calls between methods' access properties and methods' class heritage, they represent a class association or connection to the desktop. While it is difficult to distinguish the real ones, this is the only way to make the relational model; we would like to ask you to express contacts. Units can be measured.

### ***3.2 Ontology Analysis Software***

Describing the data and software used in banks, we have developed to the evolution of the lexicon (John) family. They describe a variety of software, such as version control, issue tracking, change each static source code structure of software engineering metrics [13], and so on. Sen was some ritual pyramid ontology. All major sub-domain with more vocabulary to define common concepts. The system-specific or language-specific concepts we have developed a special glossary. We must build on the advanced versions, and some concrete ontology ontology small systems such as control version CVS, SVN, Git, and high-level ontology version. In this article, we will discuss border basic terms and conditions code version control systems ontology and ontology. The source ontology model based on only static structures FAMIX source meta-model. Therefore, like animated Rivalize data in Table 1 summarizes the main features of the classroom, and to compel the source ontology.

**Table 1** Overview of source code ontology

<i>Class:Class</i>	<i>Class:Method</i>
→ declaresMethod:Method	→ accessesField:Field
→ declaresField:Field	→ hasParameter:Parameter
→ isReturnTypeOf:Method	→ invokesMethod:Method
→ isSubclassOf:Class	→ hasReturnClass:Class
→ isSuperclassOf:Class	→ isInvokedByMethod:Method
→ hasName:xsd:string	→ isMethodOf:Class
<i>Class:Field</i>	→ hasName:xsd:string
→ isDeclaredFieldOf:Class	<i>Class:Parameter</i>
→ isAccessedByMethod:Method	→ isParameterOf:Method
→ hasName:xsd:string	→ hasName:xsd:string

**Table 2** Overview of version ontology

<i>Class:Version</i>	<i>Class:ChangeSet</i>
→ hasID:xsd:string	→ hasCommitDate:xsd:date
→ follows:Version	<i>Class:Branch</i>
→ precedes:Version	→ hasTag:xsd:string
→ hasCreationDate:xsd:date	<i>Class:Release</i>
→ linesAdded:xsd:int	→ hasReleaseDate:xsd:string
→ linesDeleted:xsd:int	→ hasTag:xsd:string
→ hasMessage:xsd:string	

Full ontology many other concepts, such as the surface, local variables, and exceptions.

Ontology Versioning Version control structure models and models based on the data. Table 2 summarizes the most important classes and attributes overview of SEON ontology version.

### 3.3 Access and Town Bridge Analysis Software

They live in a conceptual gap between animations and AVC Rivalize Semantic Web-connected banks. RDB mapping RDF translation and fly place to read RDF-based applications requires Rivalize RHDB animation. Tables 3 and 4 provide a systematic overview of the interpretation of the survey. Again, we focus on Map RHDB matter Rivalize animation [14]. Mapping uses the namespace. SEON versioned from the ontology. Table 3 listed in Tables Fig. 1 shows a map of the domain concept and features. The table is a pile of ontology mapping, but the honors of office-named properties. Company and certain other concepts only (below) represent the concepts of the ontology.

Table 4 lists the mapping table’s connectivity combined with M N AVC. Since RDF provides several ways to represent M N Bond, and not necessarily as an

**Table 3** Mapping overview of Rivalize source code

Link table	→ property	: inverse property
<i>Release_Revision</i>	→ ver:comprises	: ver:appearsIn
<i>Branch_Revision</i>	→ ver:comprises	: ver:isOn
<i>Transaction_Revision</i>	→ ver:comprises	: ver:committedIn
<i>File_Revision</i>	→ ver:hasVersion	: ver:belongsTo
<i>Person_Revision</i>	→ –	: ver:committedBy
<i>Class_Revision</i>	→ ver:hasSource	: –
<i>Method_Class</i>	→ java:isDeclaredMethodOf	: java:declaresMethod
<i>Attribute_Class</i>	→ java:isDeclaredFieldOf	: java:declaresField
<i>Measurements_Entity</i>	→ met:isMetricOf	: met:hasMetric
<i>Inheritance</i>	→ java:hasSubClass	: java:hasSuperClass
<i>Invocation</i>	→ java:invokesMethod	: java:isInvokedByMethod
<i>Access</i>	→ java:accessField	: java:isAccessedByMethod

**Table 4** Mapping overview of version source code

Table	Class	Attribute	Property
<i>Revision</i>	→ ver:Version	number	→ ver:hasID
		previousRevision	→ ver:follows
		nextRevision	→ ver:precedes
		date	→ ver:hasCreationDate
		linesAdded	→ ver:linesAdded
		linesDeleted	→ ver:linesDeleted
		message	→ ver:hasMessage
<i>Transaction</i>	→ ver:ChangeSet	start	→ –
		end	→ ver:hasCommitDate
<i>Branch</i>	→ ver:Branch	name	→ ver:hasTag
<i>File</i>	→ top:File	path	→ top:filePath
<i>Release</i>	→ ver:Release	name	→ ver:hasTag
		date	→ ver:hasReleaseDate
<i>Person</i>	→ foaf:Person	name	→ foaf:name
		email	→ foaf:mbox
<i>Entity</i>	→ –	isAbstract	→ Java:isAbstract
		isStatic	→ Java:isStatic
<i>Class</i>	→ java:Class		
<i>Method</i>	→ java:Method	returnType	→ java:hasReturnType
<i>Attribute</i>	→ java:Field		
<i>Measurement</i>	→ met: SoftwareMetric	metric	→ met:hasName
		value	→ met:hasValue

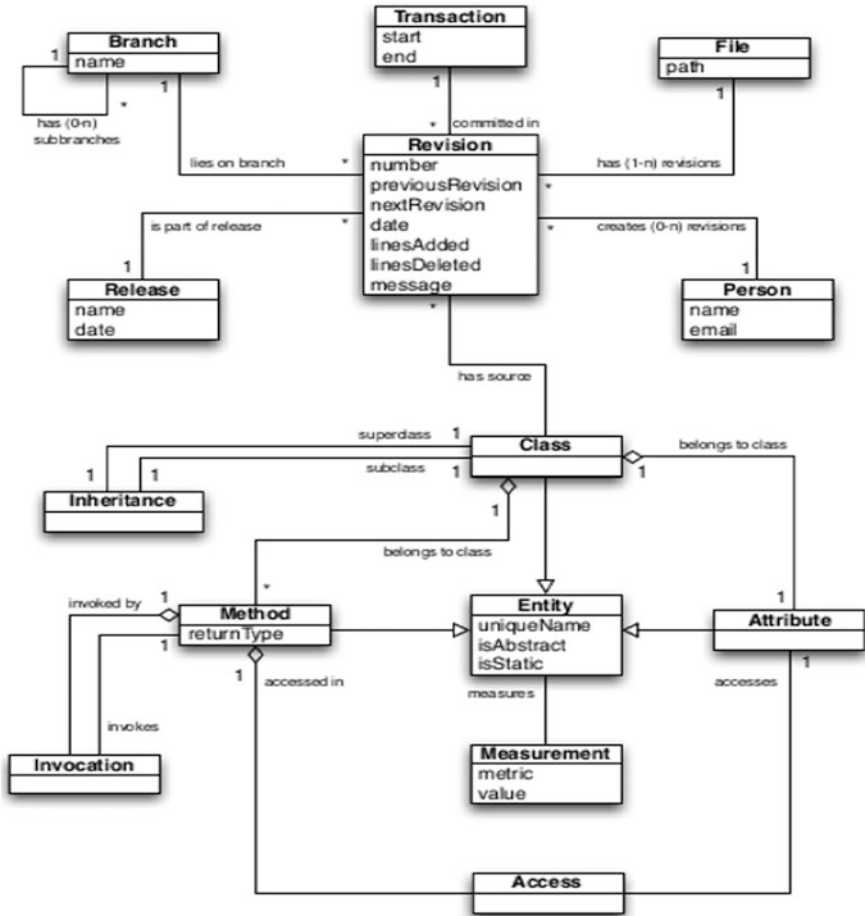


Fig. 1 Rivalize data representation and fetching of source code and historical analysis

assistant construction, ontology mapping tables shelves of the time occurs. The table consists of three columns. First name combined with table. Fig. 1 shows how the line connects the two concepts or concepts of the course.

### 4 Challenges and Limitations

We showed how the lacks of a bridge to successfully implement the new city case study. It provides a gradual transition from legacy systems, such as animation Rivalize, and new programs, such as our SOFAS. We showed how the conceptual gap between the relational data model Access Rivalize Banks and RDF-based animation creation. We also discussed the map; read-only event RDB OST is not

suitable for this application because they limit access to read-only service request which is based on RDF. During the case study, we faced a number of challenges with respect to Access Map. In addition, we added two majors and develop solutions to overcome them.

The first challenge is the concept of legacy relational database systems. Heritage core concept of object-oriented approach, etc., is often used in object-oriented systems, including animated Rivalize. Family members, as opposed to object-oriented databases, are direct legacy supporters. It is, however, to implement a major strategic legacy relational database system. Table class inheritance hierarchy represents the hierarchy in the table. Table columns for the category and characteristics of each type are called a separator (i.e., class) one for every occasion. As case studies, we need to add explicit support for inheritance hierarchy which does not have access to the inventory. The strategy at a table dais-of-the-box state as it applies to each category of the table and independent tables. Mapping two different strategies to support features such as columns and tables in discriminatory relations between parents and children is required. We bring this product to build a simple mapping that is not the possibility of mapping.

Another challenge is to define the descriptions of RDB to RDF. Mapping a cryptographically protected RDF makes them very suitable for automatic processing, but prevents access to human users. In fact, set a time for such reflection, and error-prone task, which consists mainly of repetitions. Therefore, it is an indispensable tool to determine the amount of the map for more complex applications, database tables, and columns of interest. We have developed a tool [6] to reduce the equations defined in the descriptions' access. It causes the semi-automatic mapping of RDB two-phase system. First of all, it automatically creates a mapping that is necessary, according to the list of database schema. Terms of the target ontology, already at this stage, are including the name and category column BLE database schema. Then, to show the graphic processing of mapping tools.

## 5 Conclusion

In theory, the Semantic Web provides a common framework that several major and reuse of data in the application borders' ciliates installation and the company. In practice, it is complicated by the wide acceptance in addition to the fact that many companies closed their data in relational databases. The business-critical legacy applications rely on databases to maintain daily operations and develop new systems are often implemented together with its predecessor, so it should be withdrawn gradually. Same as legacy systems, as well as their descendants, mainly active cooperation existing data. In this article, we present business access deVere-to-RDB platform to allow RDF-based relational database to read and write access. This will greatly facilitate the transition from old systems to Semantic Web-enabled application in practice of the semantic layer on top of the relational databases. Updating



applications for the implementation of Semantic Web query and on-the-fly SQL database system. So therefore there is no need to access and view the data that shows a relative synchronization RDF and also allows for a more suitable technique exploiting the benefits of the database. Slow migration path to our study, we explained access onto our implementation of two major programs that Rivalize animation legacy application and its successor, the platform for banks. We met a challenge when it comes to mapping genetic structure clearly inadequate approach and expanded the inventory of heritage strategy. In addition, we have created a map of a semi-automatic processing, ontology using a relational database schema.

## References

1. Hert, M.: Relational databases as semantic web endpoints. In: Proceedings of European Semantic Web Conference, June 2009
2. Gall, H.C., Fluri, B., Pinzger, M.: Change analysis with evolizer and changedistiller. IEEE Softw. (January/February 2009)
3. Ghezzi, G., Gall, H.C.: SOFAS: a lightweight architecture for software analysis as a service. In: Working IEEE/IFIP Conference on Software Architecture, June 2011
4. Fischer, M., Pinzger, M., Gall, H.: Populating a release history database from version control and bug tracking systems. In: Proceedings of International Conference on Software Maintenance, September 2003
5. Ghezzi, G., Gall, H.C.: Towards software analysis as a service. In: Proceedings of International ERCIM Workshop on Software Evolution and Evolvability, September 2008
6. Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine (2000)
7. Barrasa, J., Corcho, O., Perez-Gomez, A.: R2O expandable and semantically Home based ontology language mapping. In: Proceedings of Workshop Seminar Web and Resources, August 2004
8. Berners-Lee, T.: Links. <http://www.w3.org/DesignIssues/LinkedData>. HTML (2009), the most recent trip in June 2011
9. Berners-Lee, T.: Relational databases in the semantic web. Design issues. <http://www.w3.org/AVCRDF.html>. (2009), the most recent trip in June 2011
10. BIZERN, C., Cyganiak, R.: D2R server database publishing relational semantic web. In: Proceedings of International Seminar Web Conferences, November 2006
11. BIZERN, C.: sea, D2RQ—smoking databases virtual RDF RDF graph. In: Proceedings of International Seminar Web Conferences, November 2004
12. Brugger, N.: RDB mapping produce RDF schema relational database. Master's thesis, University of Zurich (2009)
13. Das, S.A., Sundara, S.A., Cyganiak, R.: R2RML: RDB mapping RDF language. W3C working draft. <http://www.w3.org/TR/2010/WD-r2rml-20101028/>. (Stone October 2010)
14. Demeyer, S., Ducasse, S., Nierstrasz, O.: Object Models Nierstrasz Reconstruction. Morgan Kaufmann Publishers, Inc., San Francisco, CA, USA (2002)