WEB APPLICATIONS DEVELOPMENT USING A COMBINATION OF JAVA AND GROOVY PROGRAMMING LANGUAGES

Katarina Mitrović*, Danijela Milošević*, Nenad Stefanović*
*Faculty of Technical Sciences Čačak, University of Kragujevac, Serbia
mitrović.kaca@yahoo.com
danijela.milosevic@ftn.kg.ac.rs
nenads@kg.ac.rs

Abstract - The area of this research is Web application development in Grails framework using Java and Groovy programming languages. This paper briefly describes the basic features of Grails framework, as well as the Groovy programming language, and the development of Web application using these technologies. Also, this paper presents comparative analysis of developing different functionalities in Java and Groovy programming languages. The aim of this work is to study similarities and differences of Java and Groovy programming languages, as well as to analyse the advantages and disadvantages of applying a combination of these languages in the development of Web applications.

1. INTRODUCTION

Ever since the programming languages were designed, the focus in the application development was on a single language. However, programmers who used only one programming language were not able to maintain high quality in each part of application, and very often they could not implement certain features, partially or completely. Over time, a lot of programming languages specialized for different purposes have been developed. It has come to the conclusion that the most effective software development could be achieved only by combining several programming languages. This trend was called polyglot programming.

This work is devoted to the analysis of a polyglot programming, in which are used Java and Groovy as a key programming languages. An overview of existing results and the literature pertaining to the subject of this paper is provided in the next chapter. The Groovy programming language and Grails framework are briefly described in the third chapter, while the development of Web applications using the above mentioned technologies is analysed in the fourth chapter.

2. BACKGROUND RESEARCH

Nowadays, the development of Web applications without combining at least two different programming languages is being difficult to imagine. Learning multiple programming languages is a demanding process, but when the necessary programming languages are mastered, developers are able not only to meet the requirements of customers, but also to speed up the process of creating applications, and at the same time to make code more concise and understandable. Having basic knowledge of more than one language can never be a con, but being expert on just one will limit you to find work for you.[1]

In multi-language programming paying attention to the level of interoperability of systems is necessary. In some cases, programming languages can be integrated smoothly and easily, but there are cases where the integration can be very difficult to achieve and where is such a combination of programming languages being unprofitable to apply in a single application. This is being determined by the programming languages themselves, necessary application functionalities, framework usage etc.

The example of polyglot programming, in which interoperability between systems takes place in a simple and effective way, is development of applications by using a combination of Java and Groovy programming languages, to whom this work is dedicated.

Development of Groovy programming language was started in 2003, with the goal of integrating dynamic functionality into Java programming language. The idea was initialized by programmer James Strachan, and the first version was realized in 2004. Groovy is an agile dynamic language for Java platform, with many features that are inspired by languages like Python, Ruby and Smalltalk, making them available to Java developers using Java syntax.[2]

Most of the functionalities offered by the Java programming language are included in Groovy, but Groovy is a lot more concise, because it takes less code for implementing them. XML parsing and SQL data manipulating is facilitated with Groovy, and a solution for scripting is provided by Groovy. Also, features that are not enabled in Java are provided by Groovy, such as writing code without semicolon, writing code without brackets, def data type for dynamic type assigning, maps, operator overloading, Star operator, Elvis operator, Spaceship operator, support for regular expressions, polymorphic iterator, incorporation of expressions in String data types, safe navigation operator, modularity, type checking, markup languages support, etc.[3]

In a very short time Groovy programming language became worldwide, and some of the largest companies embedded Groovy in their business, such as Netflix, Cisco, IBM, LinkedIn, Master Card, Oracle, Sony, UBS, etc.[3]
Grails framework represents an open source Web framework based on Groovy and Java programming languages. Grails is a powerful Web framework for the Java platform, whose goal is to duplicate the productivity of programmers. This is achieved with the “convention over configuration” approach, smart quality standards and APIs.

The whole Grails concept lies on Java virtual machine. Several software frameworks and technologies are included in Grails, such as Java EE, Spring, Hibernate and Sitemesh, and their functionalities are integrated in Grails, beside the Groovy programming language. The speed of scripting languages and the stability and security of the Java programming language are combined in Grails framework. A number of important technologies are provided to Grails by Java EE, such as agile software development, Web services and testing units. Grails is based on the MVC architecture, which is built through the abstraction of Spring’s MVC, while ORM which is used in Grails is an abstraction built over Hibernate.[4]

The existing results and the literature pertaining to the subject of this paper are listed below:

- Adam L. Davis, Modern Java: Java 7 and Polyglot Programming on the JVM, CreateSpace Independent Publishing Platform, USA, 2014
- Dean Vampler, Tony Clark, Guest Editors’ Introduction: Programming Multiparadigm, IEEE Computer Society, 2010
- Juhana Harmanen, Polyglot Programming in Web Development, Faculty of Computing and Electrical Engineering, 2013

The development of applications using combination of Groovy and Java programming languages in Grails framework is described in the following section, through examples and comparative analysis. Also, the advantages and disadvantages of this type of programming are listed in the next chapter.

3. THE EXAMPLE OF WEB APPLICATION DEVELOPMENT USING COMBINATION OF GROOVY AND JAVA PROGRAMMING LANGUAGES

For the purposes of this paper, explorative application, which implements Sales module, was developed. The application consists of activities regarding order processing. The preview of the explorative application is shown in Figure 1. Functionality is order entry functionality is demonstrated.

The class diagram for the purchase order, with its entities, relations and other elements, is shown in the Figure 2.

In explorative application development is used following:

• Grails 2.5.0. framework (https://grails.org/download.html).
• Groovy 2.4.3. programming language (http://www.groovy-lang.org/download.html).

Several directories with a particular purpose are constituting Grails project, such as Configuration, Domain Classes, Controllers, Services, Views and Layout, Libraries etc. Groovy and Java programming languages can be combined in Controllers and Domain Classes directories, so the maximum benefit from both languages can be obtained.

When writing code, the errors are not invoked if the code was written either in Java or Groovy programming languages. Each line of code can be written in any of these two programming languages, the semicolon at the end of the code line is optional, one method may consist in both Java and Groovy commands etc. There has been a great flexibility when writing code and different combinations of functionalities of these two programming languages can be used.

The main question that arises after the conclusion above is how is the program compiling the class in which are mixed two different programming languages without any problems?

The answer to this question lies in following: first of all, Java objects and classes are perceived by Groovy programming language as its own; second, the Groovy code is converted into Java byte code after compiling of Groovy classes, which is different from any other language that supports the JVM.[9] Since the Java programming language is compiled into Java byte code, it can be easily concluded that integration between Groovy and Java programming languages is achieved on the highest level.

4. RESULT ANALYSIS

4.1. Implementation of Groovy programming languages

One of the functionalities of Groovy programming language which is very useful and which was used multiple times in demonstration application is map. The tag "[:]
" is used for defining map and any number of members can be contained within a map. Each map member consists of a key and the assigned value. The default key type is String.

In the following example from the demonstration application, map is used for the presentation of the items of the document, whose members are attributes of item:

In the following example from the demonstration application, map is used for transferring message about operation execution successfullness:

```groovy
def message = [:]
message = [ispis:"Uspešno ste obrisali proizvod!",warning:"alert alert-dismissible alert-success"]
```

For database queries, simplified Groovy commands are used. A query for displaying codes and names of all the employees of customers from the database is shown in following example:

```groovy
def db = new Sql (dataSource)
def result = db.rows("SELECT SifraRadnikaKupca, ImePrezime FROM RadnikKupca ORDER BY SifraRadnikaKupca")
```

For view displaying and passing values into them, Groovy keyword render is used. In the following example, render command is used for displaying view “unos” and forwarding message values and list “listaJM” in that view:

```groovy
render(view: "unos", model: [message:message, listaJM:listaJM])
```

Passing through the members of the collection is simplified and improved in Groovy programming language by using each command. The following code shows how each member of the collection listaJMSQL is placed in the list listaJM, i.e. how the list of units of measure is converted to a list of text values:

```groovy
listaJMSQL.each { row ->
    JedinicaMere jm = new JedinicaMere(row.SifraJM, row.NazivJM)
    listaJM << jm.toString()
}
```

A comparative analysis of Groovy and Java functionalities can be made after researching the features and performances of Groovy programming language on the examples.

4.2. Comparative analysis of Groovy and Java programming languages

More flexible, shorter and readable code can be achieved by using following Groovy features:[9]

- Groovy is a dynamic language, which means that data types do not have to be assigned while writing the code; they can be assigned automatically during compilation. Groovy does not perform the standard call of methods. The call is sent as a message to an object, which can respond to it or not. In dynamic code writing, the data type is not important, but only a response message from the object.[8]
- The default access modifier is public.
- Instead of System.out.println, it is sufficient to write just println.
The semicolon at the end of code line is not mandatory.

Due to the dynamic nature of this language, the methods do not need to have a precisely defined return value. It is sufficient to use the universal data type def.

The last line of code in the method is considered as return value of the method, so the usage of return command is optional.

The upgraded version of the String data type – GString – is another useful innovation which Groovy brings. In this data type, concatenation is not performed with double quotation marks, but with literal “${text}”.

Properties are used instead of getters, setters and private access modifier of the attribute.

The following paragraphs will be devoted to a comparative presentation of certain functionalities in Groovy and Java programming languages with examples from demonstration applications.

Working with lists

Groovy

```groovy
listaRKSQl.each { row ->
  ...
}
```

Java

```java
for(RadnikKupca rk : listaRKSQl){
  ...
}
```

Groovy programming language is using the keywords each and row, while Java performs the same functionality with for loop.

Definition of method

Groovy

```groovy
def proveriString(def s) {
  ...
}
```

Java

```java
public String proveriString(String s) {
  ...
}
```

Due to the dynamic writing of Groovy programming language and default public access modifiers, Groovy method definition is shorter than in the Java language. Declaring of return value data type of methods and data types of parameters is not necessary.

The return value of method

Groovy

```groovy
def getSifraJM() {
  sifraJM;
}
```

Java

```java
public int getSifraJM() {
  return sifraJM;
}
```

The last code line in the method in Groovy programming language is automatically regarded as the methods return value, while in Java language is necessary to declare the data type of the return value and to specify the return keyword.

Working with database

Groovy

```groovy
def vratiListuRadnikaKupca() {
  def db = new Sql (dataSource)
  def result = db.rows("SELECT SifraRadnikaKupca, ImePrezime FROM RadnikKupca ORDER BY SifraRadnikaKupca")
  result
}
```

Java

```java
public ResultSet vratiListuRadnikaKupca() throws ClassNotFoundException, SQLException {
  Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
  String url = "jdbc:sqlserver://localhost:1433;databaseName=SOA_Baza;user=Dell-PC\User;password=;integratedSecurity=true;"
  Connection konekcija = DriverManager.getConnection(url);
  String upit = "SELECT SifraRadnikaKupca, ImePrezime FROM RadnikKupca ORDER BY SifraRadnikaKupca";
  Statementssql = konekcija.createStatement();
  ResultSet rs = sssql.executeQuery(upit);
  return rs;
}
```

In Groovy programming language, connection to the database is defined only once in the DataSource configuration file, so connection with database is established with a single line of code. The execution of queries in Groovy programming language is also automated by using rows keyword, so the obtaining of query results is performed with one line of code. When comparing the amount of code written in Groovy and Java language, it can be concluded that the innovations implemented in Groovy programming language influenced to significantly reduce the amount of code needed for connecting to database and executing SQL queries.

Concatenation of Strings

Groovy

```groovy
def toString (){
  "${sifraJM } - ${nazivJM }
}
```

Java

```java
String toString (){
  return sifraJM + " - " + nazivJM;
}
```
In Groovy programming language the attributes can be directly called within the string (ie. Gstrings) by combining the characters - “g{}”, while writing attributes outside the quotes is required in Java programming language.

Get and set methods (properties)

Groovy

def nazivProizvoda

Java

private String nazivProizvoda;
public String getNazivProizvoda() {
    return nazivProizvoda;
}

public void setNazivProizvoda(String nazivProizvoda) {
    this.nazivProizvoda = nazivProizvoda;
}

In Groovy programming language, instead of the standard Java getters and setters, properties are used. They represent something between methods and attributes and they are implicitly called through their get and set methods.

The advantages and disadvantages of applying a combination of Groovy and Java programming languages in developing applications is listed in the next chapter.

4.3. Advantages and disadvantages of combining Groovy and Java programming languages

The basic features and benefits of Groovy programming language are following [1][2]:

- Flat learning curve (Groovy is easy to learn for Java-programmers);
- Support for domain specific languages (flexible syntax, advanced integration and customization mechanisms for integrating readable business rules in application);
- Compact syntax (concise, readable and expressive syntax);
- Support for dynamically writing;
- Powerful processing of primitives (basic interface, or the part of code that is upgraded in order to create more sophisticated interface or code);
- Facilitating development of Web applications;
- Support for a unit testing;
- Perfect integration with Java (Groovy is transparently and smoothly connected with Java programming language and any library);
- Dynamical and rich ecosystem (development of Web applications, concurrent, asynchronous, parallel execution, test frameworks, additional tools, code analysis, support for development of user interface – these are just a few of features that Groovy covers);
- Powerful functionality (meta-programming, functional programming, static compiling and many others);
- Support for scripting.

Some of the most important advantages, which are achieved using a combination of Groovy and Java programming languages in Grails frameworks for Web applications development, are:

- A concise, simple and clean code.
- Increased productivity of programmers.
- The ability to adapt quickly to change.
- High performance and scalability.
- Cost reduction, time savings and improved software quality.
- Modularity and reusability.
- Very active and up-to-date communities.
- Maximum use of the benefits of the frameworks over which the Grails framework is made.
- The ease of setting up and starting the work.
- Focus on domain classes.
- Minimized need for restarting the application upon changes.
- Built-in testing.
- No configuration.
- Built-in support for REST.
- Easy integration.
- Large number of available plug-ins for Grails.

Some of the negative effects which are common when using a combination of Groovy and Java programming languages in Grails framework for Web applications development are:

- Learning multiple programming languages.
- If the domain class object is placed in the session, there might be a separation of the object from the session.
- Some parts of the implementation "leak" from time to time.
- Declaring all variables with def is unsustainable and illegible.
- Since the number of lines of code is reduced to a minimum, it can not be effectively applied to a large number of patterns.
- GORM is problematic in dealing with multi-threaded applications.
- It does not support other ORM technologies.

5. CONCLUSION

This research has analysed one type of polyglot programming, i.e. programming using a combination of Java and Groovy programming languages in Grails framework. It has been shown that the polyglot programming technique results in a very flexible, high quality, fast and concise software. Polyglot programming is a programming by using more than one programming language, which means that each part of the application is developed with programming language that provides the needed functionality in the best possible way. Grails framework offers the "foundation" for the development of applications, and Java and Groovy programming
languages can be used and combined, depending on the functionality that needs to be implemented.

Once the programmer learns the way how Java and Groovy programming languages work together in Grails framework, and get to know the structure and functioning of the Grails frameworks, he is able to create a Web application in such a way to gain a maximum benefit, and bypass deficiencies of single-language programming.

With single-language programming, Web applications can be created, but with using Java and Groovy programming languages application creation is easier, both in temporal and functional sense. Web application development time is shorter, implementation of functionality is simpler, each programming request can be resolved in at least one way, code is concise, clear and easy to maintain when these programming languages are being used, which leads to development of high quality applications and satisfaction of both programmers and users.

REFERENCES