A Genetic Algorithm Based Source Code Mining Approach for Language Migration

S. Geetha
Research Scholar,
Madurai Kamaraj University,
Madurai, India.
geeta_baskar@yahoo.in

Dr. K. Iyakutti
Professor, Department of Physics and Nanotechnology,
S R M University,
Chennai, India.

***Abstract*--**The rapidly changing business environment of the day dictates the need for many software migration projects. Such migration projects bring with them certain benefits and challenges. Given the nature of such migration projects and their increasing importance, many principles have been proposed for such projects. This paper is an attempt to exploit the power of the advancements in the Computer Science discipline to improve the efficiency of migration projects. In particular, we address the problem of migration of source code from one language to another. The soundness of the proposed technique is validated by empirical results obtained by applying the technique to some projects.

Keywords: *Language Migration, Data Mining, Genetic Algorithms*

*****

1. Introduction

The rapidly changing business environments coupled with technological changes often necessitate migration of software projects from one language, technology or platform to another. Such migration projects bring with them a host of new challenges and opportunities. Various studies indicate that many organizations incur a substantial expenditure on such migration ventures. It seems reasonable to contemplate on increasing the efficiency and reducing the costs incurred on such migration projects.

The present paper presents an attempt of exploiting the power of data mining and genetic algorithms in aiding migration of source code from one language to another. The paper takes a specific instance of the problem viz. conversion of Java code to C Sharp although the principles outlined are amicable for application in the context of a wide variety of programming languages.

The rest of the paper is organized as follows: Section 2 reviews the related literature in the area; sections 3 and 4 introduce the general ideas of data mining and genetic algorithms respectively. The proposed approach is outlined in section 5 and the results of an empirical study with the proposed approach are presented in Section 6. Section 7 concludes the work and suggests directions for planned future work in the area.

2. Related work

Many researches have sought to explore the challenges and opportunities presented by migration projects. The work of Zhong et.al. [1] presents an attempt to mine API mapping relations which has the potential of greatly reducing the defects introduced by migration tools like Java2CSharp [2]. [4] represents an attempt to migrate legacy COBOL code to Java/CSharp. Lin presents the problems, methods and strategies for migration to relational systems [5]. The work also lists the benefits and risks of such a migration. Vitthal et.al.present a powerful tool that enables affordable, high-performance data migration in a wide range of storage requirements [6]. Oracle describes the process of creation of a migration project plan [7]. Most of the works present the guidelines for ensuring success in migration projects. Relatively little work has been reported on application of advancements in Computer Science to deliver results in the area.

3. Data Mining

Data mining refers to the non-trivial process of extraction of novel, potentially interesting and ultimately understandable patterns from large volumes of data [3]. The basic principle underlying data mining is that past data serve as vital knowledge base from which it is possible to extract knowledge that can be utilized in solving complex hitherto unsolved problems. Many data mining techniques exist including:

Association rule mining – the process of deducing associations between data items that can be of great aid in ascertaining cause-effect relationships

Classification – given a set of data items for which the class is known, this process is an attempt to deduce the class of

data items for which it is unknown. This can be of immense utility in prediction

Clustering – the process of grouping data items into clusters such that items within a cluster have a high degree of similarity

4. Genetic Algorithms

Genetic algorithms belong to the class of evolutionary computation that attempts to mimic the natural process of evolution in uncovering solutions to complex problems. The basic idea behind genetic algorithms is as follows: solutions are represented by chromosomes which contain genes representing various attributes of the solution. The process starts with an initial random population of chromosomes each representing a potential solution. The fitness of each chromosome in solving the problem is computed using some criteria and 2 chromosomes are chosen according to their fitnesses to be parents. The selected chromosomes are crossed-over to create a new offspring that inherits attributes from both the parents. The offspring may be mutated i.e. changed at random. The fitness of the newly created individual is calculated and the process of crossing over and mutation are repeated until some terminating criterion is met.

Genetic Algorithms have been successfully applied in various domains () and have delivered promising results in the search for solutions for complex problems.

5. Proposed Approach

The main objective of the research is to exploit the power of data mining and genetic algorithms in aiding source code migration. The genetic algorithm is capable of learning complex mappings that exist between classes, methods and API usages between two languages. For example, the source code for outputting a string to the console screen in java is

System.out.println(“Welcome”);

The corresponding statement in C Sharp is

Console.WriteLine(“Welcome”);

We would want the system to learn the mapping between the Console class in C Sharp and the System.out PrinStream object in Java. Also to be learned is the mapping between the methods WriteLine of the Console class in C Sharp and the println method of the PrintSteam class in Java.

The task is obviously a non-trivial one as is illustrated by the simple example shown above. The automated translation of a source code with millions of such statements is a daunting task. Many existing tools for the task including Java2CSharp provide considerable help but still are not completely free of errors and defects [1].

The proposed solution for the problem is utilizing genetic algorithms for the task. First the system needs to be presented with projects that are available in both the languages. Such projects can be typically obtained as most organizations would have already carried out similar migration tasks in the past. Using these projects the system is trained to learn the mapping. The procedure is very similar to the one proposed by Zhong et.al. [1]

5.1 Pre-processing

To successfully apply GA to the task in hand, the programs are subject to a pre-processing step which is a simple parsing to identify the various tokens of the program such as types, method names, method parameters, object references and the like. IN the problem being addressed, since both the languages are object-oriented the difficulty is considerably reduced. In the case of applying the technique for programming languages with heterogeneous programming paradigms, the procedure is slightly more complicated. We talk more about this issue later in Section.

5.2 Solution Encoding

After the pre-processing step, the tokens are represented as genes. For instance, the System class of Java is represented as shown:

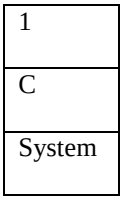

Fig 1 – Encoding of a Java Class

In the representation, 1 indicates that the language is Java, C indicates that he token being represented is a class. Followed by this identifier is the name of the class. On the other hand, the Console class of C Sharp is represented as

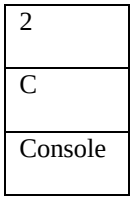

Fig 2 – Encoding of a C Sharp Class

ISSN: 2321-8169
2335 – 2338

Here 2 indicates that the language is C Sharp. After this pre-processing, solutions are represented as chromosomes each containing 2 genes for each potential mapping. For instance, the solution represented below indicates that there is a mapping between the BigInteger class in Java and Decimal in C Sharp, String class in Java and the String class in C Sharp, println method of Java and WriteLine method of C Sharp.

1	1	1
C	C	M
BigInteger	String	println
2	2	2
C	C	M
Decimal	String	WriteLine

Fig 3 – Solution represented as a chromosome

5.3 Fitness Evaluation

The fitness of each solution is obtained as follows: the mapping represented by the solution is applied to a subset of chosen projects. The Euclidean distance between the obtained translated version and the actual translated version available is calculated. Let d denote the distance. The fitness is evaluated as 1/d. The results obtained for each of the chosen subset of projects is averaged to get a measure of fitness

5.4 Selection

Roulette wheel procedure is used for selection of two potential parent chromosomes for cross-over

5.5 Cross over

Cross over is accomplished with a cross over probability of 10%.

5.6 Mutation

The offspring created by cross over is mutated with a mutation probability of 5%.

5.7 Elitism

The top 10 chromosomes (in terms of fitness) in each generation is passed over to the next generation as elitist chromosomes. This ensures that a good solution is not lost inadvertently as a result of cross-over and mutation.

5.8 Terminating Criterion

The procedure is stopped after 200 generations.

6. Results and Discussion

The proposed approach was applied to sets of 20, 35, and 50 projects. Each of the projects was available in 2 versions – java and the migrated C Sharp version. The percentages of correct mappings learned by the top 3 fit solutions with 50 projects are tabulated below. To demonstrate the superiority of the proposed approach, the results obtained using the approach proposed by Zhong et.al.[1] involving the construction of API Transformation Graphs (ATG's) are also shown.

Table 1 - Percentages of correct mappings learned by the top 3 solutions
81.23%
77.71%
73.22%

Table 2 – Percentage of correct mappings

Using Existing Approach (of Zhong et.al.)	Using GA (with 20 projects)	Using GA (with 35 projects)	Using GA (with 50 projects)
75.47%	76.62%	78.32%	81.23%

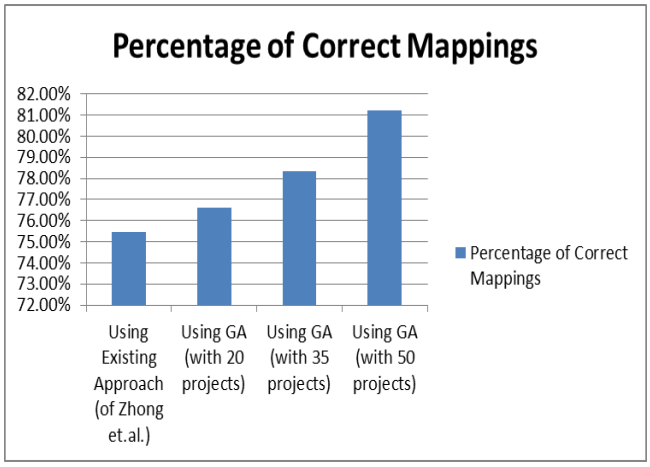

Fig 4 – Comparision of the percentage of correct mappings learned

Several observations can me made from the results. First, the performance of the pposed solution is sensitive to

the number of projects available for testing the effectivenesso f solutions. Second, the proposed approach outperforms the existing approach proposed by Zhong et.al. This is in line with expectations as one of the arenas in which GA's have been most successful is the learning of complex mappings and the problem fits in this domain.

7. Conclusions and Future Work

A GA based system was proposed for the problem of migration of source code from one language to another. Given the rapid technologial changes many more migration projects are likely to arise and hence the development of an effective system for such migration is crucial. The developed system addressed the problem of mapping from Java to C Sharp. Since both belong to the object oriented paradigm, the problem becomes slightly more simpler. More work needs to be done for migration between different paradigms (as from COBOL to C Sharp).

As a part of future work, the proposed system can be allpied to open source applications to improve the credibility of the stated results. Other advancements in Computer Science discipline like neural networks and natural computation can be applied as it is necesarry that no stone should be left unturned in effors to improve the efficiency of migration tasks.

8. References

[1] Zhong, Hao, Thummalapenta, Suresh, Xie, Tao, Zhang, Lu, Wang, Qing, Mining API mapping for language migration, Proceedings of the 32 nd ACM/IEEE International Conference on Software Engineering, Volume 1, 2010.

[2] http://j2cstranslator.wiki.sourceforge.net

[3] Han, Jiawei, Kamber, Micheline, Data Mining: Concepts and Techniques, Second Edition, 2006

[4] Software Mining, Retrieved from: http://www.softwaremining.com/services/ Legacy_Modernization_Toolkit.jsp,

[5] Lin, Chang-Yang, Migrating to Relational Systems: Problems, Methods, and Strategies, Contemporary Management Research, Vol. 4, No. 4, December 2008.

[6] Vitthal, Shinde Anita, Baban, Thite Vaishali, Warade, Roshni, Chaudhari, Krupali, Data Migration System in Heterogeneous Database, International Journal of Engineering, Science and Innovative Technology, Vol.2 , Issue 2, Mar, 2013.

[7] Preparing a Migration Plan, Retrieved from: http://docs.oracle.com/html/B15857_01/prepare.htm