Pair recommendation for pair programming

Mohanraj N.  
17epmc528@skcet.ac.in  
Sri Krishna College of Engineering and Technology,  
Coimbatore, Tamil Nadu

Akshaya S.  
akshayasampath96@gmail.com  
Sri Krishna College of Engineering and Technology,  
Coimbatore, Tamil Nadu

Ranjith K.  
17epmc538@skcet.ac.in  
Sri Krishna College of Engineering and Technology,  
Coimbatore, Tamil Nadu

Palanikumar A.  
17epmc533@skcet.ac.in  
Sri Krishna College of Engineering and Technology,  
Coimbatore, Tamil Nadu

ABSTRACT

Pair programming is a software development technique in which two programmers work together to solve a problem. When two programmers are combined together to solve a particular problem, the efficiency of the result is increased rather than the individual programmers solving the same problem. The formation of efficient pairs is very much important to increase productivity. In this project, we are proposing a methodology for the formation of efficient pairs which would increase the productivity and enhance the quality of code. Our method comprises a Graph database model in which all the programmers, their skill sets and other required attributes are created as nodes and relationships among them are established. All the Attributes are given separate weights according to the requirement. Generally, people with the same characteristics would mingle easily and can work together effectively. By this way we are finding the programmers with similar skill sets and attributes so that they can form an effective pair. Using Cosine Similarity measure, we are finding out the most similar programmers.

Experimental analysis was carried out involving the first year MCA students to check the efficiency of pairing using this method. Their details were collected and similarity was calculated among them in which the most similar students were combined as pairs. They were asked to learn a new language and solve a common problem within the given amount of time. Almost all the pairs have found the solution and coded it except few. And the feedback received from them showed that their partners were very much compatible in all aspects.

Keywords — Java, Neo4j

1. INTRODUCTION

The Pair programming technique is widely used in many software companies. The possible problems available in pairing up the programmers for pair programming and the technologies used for implementing our methodology are discussed in this chapter.

2. OVERVIEW

Pair programming is an agile software development technique where two programmers work side-by-side at one computer collaborating on the same design, algorithm, code or test. One, the driver, writes code while the other, the observer or navigator reviews each line of code as it is typed in. The two programmers switch these roles frequently. Many types of research reveal that through pair programming, software products can be produced in less time with higher quality. Also, it has been practised in the industry with great success for years.

In Pair programming, the formation of pairs is very much crucial as it determines the productivity and the quality of code. Generally, programmers are asked for a list of two to three names of potential partners and the partners are assigned based on these preferences. Sometimes random pairs are also formed. Formation of such preferred or random pairs will not be effective always. So In order to make effective pair programming, efficient pairs must be formed. We propose a method for finding efficient pairs by combining programmers with the most similar characteristics.

Our method comprises of a Graph database model A which all the programmers, their skill sets a. other required attributes are created as nodes and relationships among them are established. All the Attributes are given separate weights according to the requirement. Generally, people with the same characteristics would mingle easily and can work together effectively. In this way, we are finding the programmers with similar skill sets and attributes so that they can form an effective pair. Using cosine Similarity measure, are finding out the most similar programmers. Thus this method would produce the most effective pairs which could obviously Increase productivity.
3. SCOPE OF THE PROBLEM
The top problems of pair programming highlighted in the research paperwork "Pair programming: What's in it for me", Empirical Software Engineering and Management Conference, 2008 are as follows:

- Cost efficiency.
- Scheduling.
- Personality Clash.
- Disagreements.
- Skill differences.
- Hard to find a partner.
- Personal style differences.
- Distractions.
- Misanthropy.
- Bad Communication.

From the above list, except the cost efficiency and the scheduling factor, all the other problems arise due to the incompatible pairs. So the Manual hardships and the other problems in the partner selection of pair programming technique could be solved by automating the pair formation. Our method provides a way to find out the most similar programmers from which efficient pairs can be formed. This avoids the unwanted conflicts between the programmers and increases the productivity.

4. TOOLS AND TECHNOLOGIES
The tools and technologies used in this project are discussed below.

4.1 Google Forms
Google Forms is a free tool from Google that allows creating forms, surveys, quizzes, and such. The only requirement for creating and managing the forms is to have a Gmail account. It has the flexibility to share the forms with others and allows them to complete the forms online. It also has the feature to restrict the respondents to respond only once. After receiving the required number of responses, they can be downloaded as a CSV file and can be viewed using Spreadsheet.

4.2 Neo4J
Neo4j is a world's leading open source Graph Database. It is completely developed by using Java Language by Neo Technology. Neo4j is:

- An open source Graph Database
- Schema-free
- No SQL

Graph Database also is known as Graph Database Management system or GDBMS which stores In the form of graph structures. It stores application’s data in terms of nodes, relationships and properties. Just like RDBMS stores data the form of .rows, columns' of Tables. GDBMS stores data in the form of 'graph,

Simply, we can say that Graph Databases are mainly useful to store more connected data. If we use ROOMS Databases to store more connected data. Then they do not provide proper performance for traversing a large amount of data. In these scenarios, the Graph Database improves the application performance very well.

Nowadays, Most of the Social Networking applications like Facebook, Google+, Linkedin, Twitter, Yammer etc. and Video hosting applications like Google YouTube, Flicker, Yahoo Video etc. are using more connected data.

CQL Stands for Cypher Query Language. Like Oracle Database has query language SQL, Neo4j has COL as a query language. It is a query language for Neo4j Graph Database. It is a declarative pattern-matching language and follows SQL like syntax. It supports many clauses like WHERE, ORDER BY etc., to write very complex queries M very easy manner.

4.3 Neo4J JAVA API
Neo4j provides JAVA API to perform all database operations in a programmatic way. It supports two kinds of API:

- Neo4j Native Java API
- Neo4j Cypher Java API

5. RELATED WORKS IN PAIR RECOMMENDATION
In pair programming, there is no specific efficient methodology or system for pair recommendation. Usually, in companies where pair programming is being practised, they collect two to three name lists of partners using which they assign pairs. But many research works were based on the compatibility between the partners. Some of the factors that influence pair recommendation are discussed in this chapter.
6. FACTORS INFLUENCING PAIR COMPATABILITY

The major factors to be considered in pair programming are product quality and production rate. The current research is focusing on the influence of these factors on pair compatibility. The impact of pair compatibility was studied by (Katira et al 2004), in which, they observed that the pairs with the same skill had a significant influence on pair compatibility. Further, they also studied the impact of personality types, self-esteem and technical competence.

In their improved work, they analyzed the impact of gender and ethnicity. Finally, they conclude that pairs with same skill are compatible when compared with other factors. (Laurie Williams et al 2003) extended this work and considered additional factors like time management, learning style and work ethic. They conclude that students prefer to pair with someone they perceive to be of similar technical competence. Their results show that pairing specific learning style, yields very compatible pairs, potentially because of their ability to complement each other’s expertise. They further conclude that pairing students with strongly dissimilar work ethics will more likely yield to incompatible pairs.

7. AN ASSOCIATION RULE-BASED APPROACH

ARM and Apriori algorithm to solve ARM problems was introduced by (Agarwal et al 1993). Association rules are used in many areas like market basket analysis, social networks, stock market etc. Here we use association rules to discover compatibility between pairs.

Pair compatibility is influenced by various parameters like skill level, technical competence, designation, experience, personality interests, time database in which management, learning style and self-esteem. An association rule is to be found is viewed as a set of tuples. In market basket analysis a tuple could be (bread, butter, jam) means that whenever a customer purchases bread he/she will purchase both butter and jam in the same transaction. In the pair compatibility context, each tuple represents a parameter and the list of programmers satisfying the parameter.

For example, the tuple {A, B, C} may be the list of programmers having the skill of developing programs using Java programming language. Each tuple in the database represents a parameter with the list of programmers satisfying the same. An association rule found on this dataset may be of the form {A}, {B,C} which means programmer A can be paired with programmer B as well as with programmer C.

Our proposed methodology also finds out the compatible pairs by considering the programmers as vectors and finding similarity among them using cosine similarity.

8. PROPOSED METHODOLOGY

The Effectiveness of the pair programming can be determined only by the efficiency of the pairs generated. The methodology for generating efficient pairs is discussed in this chapter.

8.1 General idea

We are proposing a graph database model to store the skill sets and the other characteristics of the programmers participating in pair programming using which similarities among the programmers is found using the cosine similarity measure. The programmers having the greatest similarity value are combined together to form an effective pair.

8.2 Data collection

The Programmers participating in pair programming should produce their profile having their complete details. The details may be questioned separately by the employer according to their requirement or it may be taken directly from their employee database. In our method, we have used Google Forms for data collection. The details collected should contain the skill sets and characteristics of the programmers. Some of the basic data that is compulsory to be collected are as follows:

- Name
- Gender
- Age Group
- Nativity
- Languages
- Programming Proficiency
- Years of Experience
- Domain Experience
- Projects

The above mentioned are crucial as it determines the pair compatibility eliminating the problems in pair programming such as Personality clash, Disagreements, Skill differences, Personal style differences, Misanthropy, Bad Communication.

8.3 Data processing

The collected data is stored in a graphical database for easy representation and retrieval. The graphical database has two types of nodes namely programmers and features. The edges are only between the programmers and their features nodes. Each programmer's features are retrieved from the database as a set of attributes. The similarity is calculated among all the programmers and the pairs are generated.
9. REFERENCES


