CHAPTER 5

mROSE: A RECOMMENDATION SYSTEM FOR MoDSE

Model Driven approaches have become a new software trend in software development process. MDE needs a new paradigm for software evolution which is known as MoDSE [1]. Many CASE tools have been evolved due to wide usage of model driven approaches. Tools are needed at different stages of model driven evolution. So, here the question arises “how do you choose the right tool?”. The contradictory experiences with MDA and UML tools appear puzzling and difficult to interpret. Tools do much work in model driven approaches [7]. So, it is very much essential to choose tools carefully. Hence, this chapter presents an approach for proposed recommendation system which helps users to choose right tool for right concern by generating automated recommendations.

5.1 NEED FOR A RECOMMENDATION SYSTEM IN MoDSE

Basically, MDA tool is a tool used to develop, interpret, compare, align measure, verify, transform, etc. models or meta models. In MDA approach we have essentially two kinds of models: initial models are created manually by human agents while derived models are created automatically by programs. For example, an analyst may create a UML initial model from its observation of some loose business situation while a Java model may be automatically derived from this UML model by a Model transformation operation which can be done with the help of automated tools. These tools perform more than one of the desired functions. For example, some creation tools may also have transformation and test capabilities. There are other tools that are specifically for creation, especially for graphical presentation, and only for transformation, etc. There is an increasing need for more disciplined techniques and engineering tools to support a wide range of model evolution
activities, including model-driven software evolution, model differencing, model comparison, model refactoring, model consistency, model versioning and merging, and (co-)evolution of models.

Research presented here proposed a framework which is discussed in chapter 4, from the evaluation it is observed that tool assessment and/or comparison is manual. To assess the tools under this platform user need to have minimum knowledge about tools. If the number of tools increases assessment becomes tedious. MDA and UML tools are suitable for the MoDSE concerns. To automate the tool assessment, comparison and selection of right tool for right concern in MoDSE, this chapter presents a recommendation system named as ‘mROSE’. It is used for tool selection which is conceptualized as a form of stakeholder interests and/or concerns. Such a perspective allows users to anticipate, explain, and evaluate different experiences and consequences following introduction and intention of tools. Recommendation system is a software application that aims to support users in their decision-making while interacting with large information spaces [57]. They recommend items of interest to users based on preferences they have expressed, either explicitly or implicitly. Ever-expanding volume and increasing complexity of information has therefore made such systems essential tools for users in a variety of information seeking activities.

Recommendation system helps to overcome information overload problem by exposing users to most interesting items, and by offering novelty, surprise, and relevance. However, there has been no systematic formulation of MDA tools for stakeholder concerns in MoDSE. Stakeholder concerns in MoDSE are identified as model mapping, model merging, model integration, model transformation, model consistency etc., [1, 13, 19, 26, 60, 65, 76, 94]. To gain knowledge about tools, stakeholders definitely explore existing tools in the literature. Much of the literature on these tools has intended to focus on discussion forums, panels, comparison strategies and frameworks. Discussion forum and/or panel are where users can share their ideas and answering the questions of audience [40, 47, 56, 62, 73, 87, 111, 112, 113, 114]. Framework determines the comparison strategies for features of tools of same category under a uniform platform [2, 14, 15, 23, 49, 105]. Thus, the proposed system
is intended to provide recommendations by bringing stakeholder concerns and tools together. Thus, the proposed system provides recommendations for tool selection in the context of Model-Driven Software Evolution. It is different from the above mentioned forums and communities.

5.2 AN APPROACH FOR PROPOSED RECOMMENDATION SYSTEM

Many stakeholders spend substantial effort in finding out the appropriate tool among huge number to perform the activities in MoDSE. Major goal of this chapter is to bring stakeholder, tools and concerns together to select a tool which is capable to achieve their concerns. The following subsequent sections demonstrate approach used in the proposed System. Recommendation System for Software Engineering (RSSE) is defined in as “a software application that provides information items estimated to be valuable for a software engineering task in a given context” [57]. mROSE is a proposed recommendation system which generates useful tool recommendations for users to understand evolution of models in Model-Driven Software Evolution.

To date, most Recommendation Systems for software engineering (RSSE) have focused on recommendations related to software development artifacts, particularly source code. RSSEs typically recommend code-to look at, change, or reuse. However, recommendations could address many other aspects of software development such as quality measures, tools, project management, and people could support an ever-widening array of software engineering tasks [57]. Recommendation system might use activity logs to deduce questions developers ask, and then coach them automatically on appropriate, possibly unfamiliar tools or features to answer those questions more efficiently [21]. Recommendation engine is required to generate recommendations. So, the following sub section describes the algorithms used for constructing recommendation engine.

5.2.1 Algorithms for Recommendations Generation
In this section to generate recommendations an existing two model based recommendation algorithms are used [29, 66, 99]. The similarity relation between the different tools is captured by first algorithm by building a model, where as the second algorithm uses above model to derive recommendations for an active user. In first algorithm (i.e. Algorithm 5.1) the model \( M \) is constructed which is used for generating recommendations. Matrix \( T \) is the \( n \times m \) matrix which is the input to this algorithm, where \( n \) denotes the distinct number of users and \( m \) denotes the distinct tools in dataset. The similarity relation between tools is computed as pair wise tool-to-tool similarity. Parameter \( p \) specifies the number of tool-to-tool similarities that will be stored for each tool. The output of the Algorithm 5.1 is the model \( M \), which is represented by an \( m \times m \) matrix. In model \( M \), the \( j^{th} \) column stores the \( p \) most similar tools to tools \( j \). In particular, if \( M_{ij} > 0 \), then the \( i^{th} \) tool is among the \( p \) most similar tools of \( j \).

BEGIN ALGORITHM /* To Compute Tool Similarities */

1. READ THE MATRIX ‘T’ AND SIMILARITIES ‘p’.

2. INITIALIZE THE MOST SIMILAR TOOL ‘j’ TO 1

3. INITAILIZE THE REGULAR TOOL ‘i’ TO 1

4. IF ‘i’ NOT EQUAL TO ‘j’

   THEN \( M = \text{SIM} (j^{th} \text{ tool in } T, i^{th} \text{ tool in } T) \)

   ELSE \( M = 0 \)
5. REPEAT STEP 4 UNTIL $i > m$ /* ‘m’ is number of tools */

6. INITIALIZE THE REGULAR TOOL ‘i’ TO 1

7. IF ‘i’ NOT EQUAL TO AMONG LARGEST VALUES OF ‘p’

   THEN $M = 0$

8. REPEAT STEP 7 UNTIL $i > m$ /* ‘m’ is number of tools */

9. REPEAT THE STEP 2 TO STEP 8 UNTIL $j > m$

10. PRODUCE ‘M’

END OF ALGORITHM

Algorithm 5.1: To Compute Tool Similarities

The precomputed model $M$ is (Algorithm 5.1) treated as input to the Algorithm 5.2. The tools that have already been selected by the active user is denoted as $U$ which is an $m \times 1$ vector, and $N$ is the number of tools to be recommended. If the active user has selected the $i^{th}$ tool then the information in vector $U$ is set as $Ui = 1$ and zero otherwise. $X$ is an $m \times 1$ vector which stores the nonzero entries correspond to the tools to be recommended ($N$). The output of the algorithm is the number of tools recommended to active user which is denoted as $N$. The number of recommendations to be generated is depends on the number of similar tools $p$ which is used to build model $M$ and $U$.

BEGIN ALGORITHM /* To Generate Recommendations */
1. READ THE MODEL ‘M’

2. ACTIVE USER SELECTED TOOLS ‘U’, WHERE U is mx1 VECTOR

   /* ‘m’ is number of tools */

3. THE NUMBER OF TOOLS TO RECOMMEND ‘N’

4. MULTIPLY ‘M’ WITH ‘U’ AND KEEP IT INTO ‘X’

   WHERE ‘X’ IS NON ZERO ENTRIES OF RECOMMENDED TOOLS

   \[ X: = M*U \]

5. INITAILIZE THE SIMILAR TOOL ‘j’ TO 1

6. IF ‘U’ NOT EQUAL TO ‘0’ or SEARCH FOR NON ZERO ENTRIES IN ‘U’

   THEN \[ X: = 0 \]

7. REPEAT STEP 6 UNTIL \( j > m \)  /* ‘m’ is number of tools */

8. INITAILIZE THE REGULAR TOOL ‘j’ TO 1

9. IF ‘X’ NOT EQUAL TO AMONG ‘N’ LARGEST VALUES IN ‘X’

   THEN \[ X: = 0 \]

10. REPEAT STEP 9 UNTIL \( j > m \)  /* ‘m’ is number of tools */

11. PRODUCE ‘X’.

END OF ALGORITHM
Algorithm 5.2: To Generate Recommendations

The output of the algorithm is vector $X$ which is computed as follows. First, multiply $M$ with $U$ to compute vector $X$ (step 4). Union of the similar concerns for each tool that has selected by the active user and the sum of these similarities corresponds to the nonzero entries of $X$. Second, verify the number of tools that have already been selected by the active user is not equal to zero or nonzero entries of $U$, then $X$ is set to zero (Step 6). Finally, $X$ sets to zero by verifying all the entries of $X$ if it is not equal to $N$ largest values of $X$ (Step 7). Reasonably small values of parameters such as $M$ ($10 \leq M \leq 30$), $X$ ($1 \leq X \leq 10$), $U$ ($1 \leq U \leq 5$), and $N$ ($1 \leq N \leq 10$) are considered for evaluation purpose which leads to good results. Evaluation is also done for even larger values like 50 tools selected by an active user in which comparison for similarities and generating recommendations is similar to the above mentioned reasonable values.

5.2.2 Computing Pair Wise Tool Similarity

In general user based recommendations systems for a particular user are computed by using three step approach [67] and the approach used in the mROSE is described as follows:

- In the first step, identifying the user interest (concerns selected), for each selected concern, identify the right tool.

- In the second step, compare the selected tools from the displayed list.

- In the third step, based on the comparison report, calculate the ratio of number of concerns selected by the user, total number of concerns, and ratio of number of
selected concerns, number of satisfied concerns by each tool. By using these numerals recommendations are generated.

In proposed recommendation system mROSE, existing ‘MapReduce’ technique is used to compute pair wise similarity between the tools [42]. It is based on the combinators of ‘map’ and ‘reduce’ functions. ‘map’ applied to collect data items such as tools and concerns and returning the results. ‘reduce’ function applied to collect only interested tools/concerns and returning the results. This technique is selected because of its efficiency in computing pair wise similarity for large collections. For recommendations where there is a need to find the similar features of tools to a tool which is interested at. So, there is a need to compute the similarity between pair of tools. But this kind of recommender systems has large data sets where it needs lot of computations, correlations and to deal with temporal aspect since the user interests changes with time. So, we need correlation calculation done periodically so that the results are up to date. For this reason ‘MapReduce’ technique is used to handle this scenario.

In proposed recommendation system mROSE, input is mapped by map function and output is returned by reduce function. User selecting either concern or tool is an input to mROSE, then map function maps selected concern/tool with all the tools in the data set then tools and their concerns are get shuffled. To eliminate unwanted concerns and/or tools reduce function is used where only user selected concerns and relevant tools are listed out. This is done by comparing similarities between the tools. The reduced set of tools comparison is not possible at a time so, pair wise tool similarity is computed. Similarities between the tools are identified as symmetric similarity. In mROSE symmetric similarity is treated as string similarity which is a concern like ‘model mapping’, ‘model to code transformation’ and tool names like ‘ArgoUML’, ‘Enterprise Architect’.
In mROSE pair wise tool similarity expressed as an inner product of concern weights. A tool $t$ is represented as a vector $W_t$ of concern weights $w_{c,t}$, which indicates the importance of each concern $c$ in the tool $t$ by considering symmetric similarity, measures defined as follows:

$$ Sim(t_i, t_j) = \sum_{c \in V} w_{c,t_i} \cdot w_{c,t_j} $$

(1)

Where $Sim(t_i, t_j)$ - similarity between tools $t_i$ and $t_j$ and $V$ is the concern set. In this type of similarity measure, a concern will contribute to the similarity between two tools only if it has non-zero weights in both. Therefore, $c \in V$ can be replaced with $c \in t_i \cap t_j$ in equation (1). If a concern appears in tools $x$, $y$, and $z$, it contributes only to the similarity scores between $(x, y)$, $(x, z)$ and $(y, z)$ and the list of tools that contain a particular concern is exactly retrieved. Thus, processing all retrieves the entire pair wise similarity matrix can compute by summing the concern contributions. Using ‘MapReduce’ technique in proposed recommendation system the following observations are made:

- Efficiently performs pair wise similarities between tools.
- Computation complexity is reduced.
- Allows the user to detect the similarities between the numbers of tools at a time.
- Response time for the user is made faster.

Using ‘MapReduce’ technique, similarity computations of number of tools at a time is implemented efficiently in proposed mROSE recommendation system. Recommendations are generated easily based on tool similarity summary.

‘MapReduce’ technique is used in mROSE as follows:

Step 1: Input: select concern

Step 2: ‘map’ selected concern with concerns of all the tools.

Step 3: Similarities are computed based on string matching.
Step 4: Output: For similar concerns ‘Reduce’ lists only the tools which have selected concern.

The above ‘MapReduce’ technique is used in mROSE in the following manner:

**Fig. 5.1 Map Function in mROSE**
5.3 SUMMARY

This chapter deals with proposed recommendation system which is named as mROSE. Proposed recommendation system is used to generate recommendations for selecting right tool for right concern in MoDSE. Detailed explanation about how recommendations are generated by using two algorithms is given. Three step approach in mROSE is described and which is used for selecting user interest, for selected concerns appropriate tools get selected and these selected tools will be compared and summary also generated, based on this recommendations will be provided. For tool comparison MapReduce technique is used which exists in the literature for computing pair wise ItemSimilarity. But this technique used here for computing pair wise similarity between various tools. mROSE recommendation system generates recommendations entirely depends on the tool comparison summary. Proposed mROS can compare any number of tools at a time and even the number of tools is huge. It is
also noticed that time taken to generate comparison summary and recommendations is very less.