DECISION SUPPORT FOR REQUIREMENTS ENGINEERING PROCESS DEVELOPMENT

Li Jiang Armin Eberlein
Department of Electrical & Computer Engineering
University of Calgary, Canada
e-mail: {jiangl; eberlein}@enel.ucalgary.ca

Abstract

This paper presents a methodology that provides decision support for the iterative development of requirements engineering (RE) processes, which is a significant step towards bridging the gap between RE theory and practice. The methodology presented is based on a multi-criteria decision making model [18] and an N-dimensional process development methodology [1] in order to provide support for requirements engineers to tailor/define RE process models and to select the most appropriate RE techniques for a specific project. Currently available RE models and techniques are stored in a RE Process Knowledge Base (REPKB) which is the basis for decision support. The general theory behind this methodology is described and an algorithm for the implementation of our decision support is discussed in the paper. Preliminary results suggest that the framework makes a useful contribution to requirements engineering by helping requirements engineers to develop better RE processes.

***Keywords*: Requirements engineering, process model, decision support, process development.**

1. INTRODUCTION

The importance of requirements engineering (RE) has been stressed numerous times in literatures. Some literature provides empirical evidence of the benefits of RE [2][3][4], while others report case studies [5][6] that indicate that improving the RE process will potentially lead to improvements in the productivity of large and medium-sized software organizations [7].

Most people agree that following a well-defined RE process and using appropriate RE techniques will have a positive contribution to software quality. Nevertheless, there is still a big gap between RE theory and practice.

Currently, there are numerous RE process models available [8] and each one uses several techniques that address different issues of system development. In our previous research, we identified about 26 RE process models and 56 techniques that can be used in RE processes for various domains.

We believe that the choice of RE models and techniques for a project is an important part of software development. Nevertheless, this aspect is often ignored in favor of personal preference or existing inertia of company practice [9]. Many researchers and practitioners emphasized the significance of choosing the right techniques and models for specific software development. Glass et al stressed that we need a way to choose the most appropriate software development methodology for the task at hand [10]. Davis states that knowing which technique to apply for a given specific problem is necessary for effective requirements analysis [11]. Cockburn further argues that one methodology per project is essential for the success of the project [12].

When choosing suitable models and techniques for modeling requirements of a particular project, characteristics of the proposed system, its context and constraints need to be considered [9][14][15]. Previous research in this area shows that the values of project attributes are relevant to the choice of RE techniques and models [9][10][13]. Furthermore, given certain project attributes, some RE process models are more suitable than others and certain techniques will provide more benefits than others.

The methodology proposed in this paper is a step towards bridging the gap between RE theory and practice by providing decision support for the requirements engineers to develop appropriate RE models and to select suitable techniques.

The paper is organized as follows: Section 2 discusses decision support models. The algorithm for the implementation of the methodology is discussion in section 3. Section 4 provides conclusions and outlines further research.

CCECE 2003–CCGEI2003, Montreal, May/mai 2003
0-7803-7781-8/03/$17.00© 2003 IEEE

2. DECISION SUPPORT MODELS

Two decision support models have been defined in our research: the first model is a decision support model for the selection of process models and techniques. This model helps requirements engineers to select the most appropriate RE models and techniques. It is the first step towards providing support for tailoring appropriate RE models and techniques. The selection could be an iterative process. The second model provides an evaluation mechanism for the RE process model and techniques that the requirements engineer tailored or selected. This provides information about how well the defined model and selected techniques are suited for the project at hand.

2.1 Decision Models

At a high level of abstraction, RE models and techniques can be divided into the following two parts:

(1) *Domain-oriented models and techniques* are best suited for specific application domains. Further, the problem domain can be subdivided into e.g., embedded domain, semi-embedded domain, communication domain, general information and web-applications domain. Different domains require different models and techniques.

(2) *Domain-independent models and techniques* can be applied to any type of domain or project but often need to be tailored to the project at hand.

The first decision support model is based on project and product attributes and helps the requirements engineer decide on the RE model and techniques that will be used in the project. The following gives brief descriptions of some of the project and product attributes used in our approch:

Project size defines the size of the project in terms of number of requirements. Possible values are: Very big, big, medium, small, very small.

Requirements volatility has the following five levels: Very high, high, middle, low and very low.

Project category has the following values: Communication, embedded, semi-embedded and dynamic. Some of these categories were taken from the COCOMO model.

Degree of safety criticality can have the following values: very high, high, middle, low and very low.

Similarly, we define other attributes such as Product Type, Team Size, Project Complexity, Project Duration, Project Risk, Organization and Customer relationship[3], Uncertainty factors, Product Quality Standard, etc.. The various values of these attributes have provided good information for the selection of RE models and techniques. We constructed several decision tables within REPKB to support decision making. The major tables are "Models_Selection" and "Techniques_Selection", the general structure of which are shown in Tables 1 and 2:

Table 1 RE model selection decision table

Case No.	A_1	...	A_j	...	A_n	M_1	...	M_k
...	...	...	...	...	...	...	...	...
i	$V_{i,1}$		$V_{i,j}$	...	$V_{i,n}$	$V(M_1)$	...	$V(M_k)$
...	...	...	...	...	...	...	...	...

Table 2 RE techniques selection table

Case No.	A_1	...	A_j	...	A_n	T_1	...	T_m
...	...	...	...	...	...	...	...	...
i	$V_{i,1}$		$V_{i,j}$	...	$V_{i,n}$	$V(T_1)$	...	$V(T_m)$
...	...	...	...	...	...	...	...	...

$A_1,\ldots,A_j,\ldots A_n$ represent the condition attributes of the project in the decision table. $V_{i,1},\ldots,V_{i,j},\ldots,V_{i,n}$ represent a group of values for the attributes $A_1,\ldots,A_j,\ldots A_n$. $M_1,\ldots,M_k$ represent the recommended RE process model, $T_1,\ldots,T_m$ represent the RE recommended techniques. M_i and T_i are decision attributes. In the Models_Selection table, $V(M_1)$ means the first recommended RE model for the given values of $V_{i,1},\ldots,V_{i,j},\ldots,V_{i,n}$. $V(M_k)$ means the *k*th recommended model. Similarly, in the Techniques_Selection table, $V(T_1)$ means the first recommended RE technique for the given values of $V_{i,1},\ldots,V_{i,j},\ldots,V_{i,n}$, and $V(T_m)$ means the *m*th technique that is recommended. We organized the RE techniques into the following five categories in order to provide better decision support: Elicitation (E), Analysis & Negotiation (A), Documentation (D), Validation & Verification (V) and Tool Support (T).

Following the structure of the decision tables of Model_Selection and Technique_Selection, 50 cases have been defined based on surveys of project managers in industry and analysis from our previous research. We expect that more cases will be defined and stored in the REPKB as our research progresses.

Based on the established cases in the two tables, the decision support methodology uses Case-Based Reasoning [17] and multi-attribute decision-making theory [18]. The similarity calculation uses the Euclidean Distance and the Weighted Euclidean Distance to calculate the similarity between the user-defined context and the cases stored in the REPKB.

Once the requirements engineer has specified the attribute values of a certain project, the most similar cases in the REPKB will be identified and the most suitable process models and techniques for this case will be retrieved. Based on these cases the requirements engineers can make the final decision about the model s/he wants to tailor and the techniques s/he wants to select for the project at hand.

2.2 Evaluation Model

In order to allow the requirements engineer to evaluate the tailored RE process model, we developed rules and evaluation functions. In the following sections, we give a brief introduction to the rules and (due to space limitations) to one of the evaluation functions.

2.2.1 Rules.

Various factors need to be considered when developing RE process models. Of course, one of the major concerns is to develop high-quality requirements, but we also have to consider other issues such as time-to-market pressure, cost-benefit of the project, etc. Additionally, some RE process models and techniques have key features that make them more or less suitable for certain situations. All these characteristics serve as guidance for the usage of these models or techniques. The rules we defined are based on these principles. Two examples of rules that we defined are:

- *IF* RE_Model_Selected=XP **&** Technology_Set **INCLUDE** SDL, *THEN* Process_ Suitability_ Level = low
- *IF* RE_Model_Selected=**SREM** & Num_of_CMM_Activities>=2 & Technology_Set **INCLUDE** Formal method, *THEN* Process_suitability_ level=high

These rules reflect both the current practice of RE in industry and research results in academia. Even though these rules still need further refinement, we argue that our current set of rules show the feasibility of our approach to decision support for requirements engineering.

2.2.2 Process Matching Functions.

Process Matching Functions define the overall suitability of the models and techniques which the user developed and selected to the characteristics of the product under development and its project environment. They consist of two matching functions: Techniques Matching Function and Model Matching Function.

Table 3. Model matching table

A_i	M_1	...	M_k
$V_{1,i}$	$m_{1,1}$	...	$m_{1,k}$
...	...	...	...
$V_{i,i}$	$m_{i,1}$	...	$m_{1,k}$
...	...	...	...
$V_{k,i}$	$m_{k,1}$	...	$m_{1,k}$

Table 4. Techniques matching table

A_i	T_1	...	T_n
$V_{1,i}$	$m_{1,1}$	...	$m_{1,k}$
...	...	...	...
$V_{i,i}$	$m_{i,1}$	...	$m_{i,k}$
...	...	...	...
$V_{k,i}$	$m_{k,1}$	...	$m_{k,n}$

For each attribute value, we give an evaluation value against each technique and each model as illustrated in Tables 3 and 4. These tables are part of REPKB. Based on this evaluation, the formal definition for the matching function technique t can be represented as:

F_t: $A_i \rightarrow M_t$, $\quad M_t \in [0..1]$

where:

A_i represents attribute i; its *j*th value is $V_{j,i}$

T: represents technique set, $t \in T$ is one of techniques

M_t: represents the match of technique t with respect to A_i

The overall matching value of the selected RE techniques used within a model can be represented as:

$$\beta = \sum_X average\ (\sum_t X.F_t(A_i))$$

$X \in \{E,A,D,V,T\}$. Here X is the category into which the RE models have been divided (see section 2.1).

The definition of a matching function for RE process models is similar to the definition of the technique matching function.

3. ALGORITHMS FOR THE IMPLEMENTATION OF THE METHODOLOGY

We developed two algorithms for the implementation of the methodology that we proposed in the earlier section. The first algorithm helps users tailor existing RE process models and select most appropriate techniques. The second algorithm supports users during the development of RE models. In this paper, we only describe the first algorithm. The second algorithm was published in [1].

The algorithm for tailoring existing process models is as follows:

Step 1: *Define the context of the project.* In this step, the user needs to define all the attributes of the project and product; the values of these attributes define both the criteria of the selection of RE process models and the techniques.

Step 2: *Similarity reasoning for identifying the most appropriate RE process model and RE techniques.* Based on the user's definition of the project and product attributes, case-based reasoning searches for suitable models and techniques already stored in the REPKB. The most similar cases will be identified and recommended to the user.

Step 3: *First selection.* The user will make a first decision based on the recommendations provided by step 2, i.e. s/he needs to select one process model and some techniques and then tailor the process model and techniques in step 4.

Step 4: *Tailoring of activities and techniques selected.* The requirements engineer tailors the selected process model and techniques according to RE process knowledge and tailoring rules in REPKB. The building blocks of the tailoring framework are illustrated in Fig. 1:

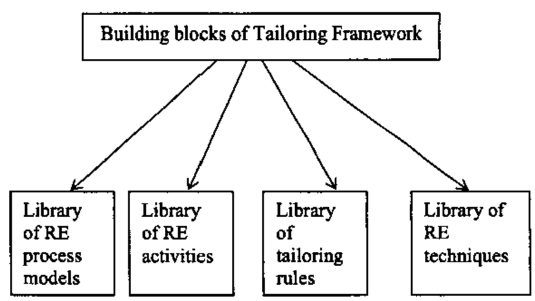

Figure 1. Overview of Building block for tailoring

In order to help the tailoring process and to manage the RE model, an RE model template is defined in the methodology. The template includes the following attributes: Activity_Number, Activity_Name, Actors, Category, Objectives, Pre-Activities, Post-Activities, Artifacts, CMMI_level, Entry_Criteria, Time_Consuming_Factors, Subactivities, Detail_description, Applicable_Domain, Candidate_Techniques, Exit_Criteria.

The significance of the template is summarized briefly in the following:

Firstly, the template provides a unified model for the definition and management of RE process models stored in REPKB. Secondly, it provides a mechanism for identification of the relationships among activities, such as subactivities, superactivities. Thirdly, dynamic relationships, such as pre-acitivities, post-activities can be identified according to the user's definition. Fourthly, the key elements of RE processes and related RE issues can be identified from this template, such as roles, activities, objectives of RE. All this information provide further help for requirements engineers to decide which activity and techniques are best for the context defined during step 1.

Step 5: *Evaluate the tailored RE model according to the rules and evaluation functions.* The tailored RE model and the selected techniques are evaluated based on the evaluation models and rules discussed in section 2.

Step 6: *Check the evaluation result.* If the calculation result is lower than a predefined threshold or too many rules have been violated, the model customization has likely to be reconsidered. If the user is not satisfied with the evaluation results, then s/he has to go back to either step 2, 3 or 4.

Step 7: *Customize the model and techniques.* In this step, the user will further adjust the sequence of activities and define more details for each activity if necessary. The methodology provides guidance about suitable requirements metrics.

4. CONCLUSIONS AND FUTURE RESEARCH

Process development is time-consuming and expensive and few project teams can afford to develop their own models from scratch [16]. For this reason, support during process development is necessary to enact effective software processes for projects. This is also true for RE process development. The methodology proposed in this paper provides a possible solution for RE process development by combining decision support mechanisms with evaluation mechanisms.

A case study has shown that this framework provides good support for RE process development and techniques selection. The benefit gained from this methodology is not only that it can help requirements engineers to develop suitable RE process models, but this methodology can also provide education opportunities for requirements engineers to enrich their RE knowledge through REPKB.

Our further research will provide more evaluation functions for RE model evaluation as well as an implementation of the approach.

References

[1] A. Eberlein and L. Jiang, Towards a RE Process Model, 7th International Conf. on OO Information Systems, 2001, 281-290.

[2] B. W. Boehm, Software Engineering Economics, Prentice Hall, 1981

[3] L. A. Macaulay, Requirements Engineering, Applied Computing, Springer, 1996.

[4] I. Sommerville and P. Sawyer, Requirements Engineering: A Good Practice Guide. Chichester: John Wiley and Sons.

[5] D. Herlea, C.M. Jonker, J. Treur, and, N.J.E. Wijngaards: A Case Study in RE: a Personal Internet Agent, *Technical Report,* Vrije Universiteit Amsterdam, Dept. of Artificial Intelligence.

[6] T. Thanasankit and B. Corbitt, Towards Understanding Managing RE - A Case Study of a Thai Software House, Proc. 10th Australasian Conference on Information Systems, 1999

[7] K.E. Emam and A. Birk, Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability, IEEE Trans. on Software Engineering, Vol 26, NO. 6, Jun. 2000.

[8] L. Jiang and A. Eberlein, RE: A Review and A Proposal, Proceedings of the Third ASERC Workshop on Quantitative and Software Engineering, Feb. 17-18, 2003, Banff, Alberta, Canada.

[9] Elizabeth Haywood and Philip Dart, Analyzing projects to decide how to model the requirements, Proceedings of The Fourth Australian Conference on RE, Sep. 1999, pp149-159, CSIRO-Macquarie University JRCASE

[10] R. L. Glass and Iris Vessey, Contemporary Application-Domain Taxomonies, IEEE Software Pages 63-76, 1995

[11] Davis M. Alan, Software Requirements, Objects, Functions and States Prentice Hal, 1993.

[12] Alistair Cockburn, A Methodology Per Project, Technical Report, Humans and Technology, TR 99.04, Oct.1999.

[13] S. Pfeiffer & A. Eberlein, RE for Dynamic Markets, Proceedings of the 6th IASTED International Conference, Cambridge, MA, November 4-6, 2002, pp. 100-105

[14] G.R. White and H.Shoaee, The NLC software requirements methodology, 8th International conference on Accelerator&Large Experimental Physics Control Systems, 2001, San Jose, Cal.

[15] U. Nikula and J. Sajaniemi, BaSyRE: A Lightweight Combination of Proven RE Techniques. International Workshop on Time-Constrained RE. pp.69-78.Germany, Sep. 9, 2002

[16] A. Agnar and E. Plaza, Case-Based Reasoning:Foundational Issues, Methodological Variations, and System Approaches, AICom - AI Communications, IOS Press, Vol. 7: 1, 39-59. 1994

[17] W. S. Humphrey and M. I. Kellner, Software Process Modeling: Principles of Entity Process Models, Technical Report, CMU/SEI-89-TR-002, ESD-89-TR-002, Feb. 1989

[18] C. Zopounidis and M. Doumpos, PREFDIS: A multicriteria decision support system for sorting decision problems,Computers and Operations Research, vol. 27, No. 7-8, pp 779-797, 2000.