A TOOL FOR REQUIREMENTS ENGINEERING PROCESS DEVELOPMENT

– A Knowledge Engineering Perspective

Li Jiang

School of Computer Science
The University of Adelaide, Australia
ljiang@cs.adelaide,edu.au

Armin Eberlein

Computer Engineering Department
American University of Sharjah, UAE
eberlein@ucalgary.ca

Abstract

*Using Software Engineering (SE) knowledge to support the software development process is challenging due to the complex structure of SE knowledge and the uncertain nature of large-scale software projects. However, already before developing software, establishing a development process suitable for the project at hand is another challenge. This paper presents a Requirements Engineering (RE) tool that contains a knowledge base to support RE process development and selection of RE techniques. The tool is built based on the **F**ramework for **R**equirements **E**ngineering p**R**ocess d**E**velopment (FRERE). The major merits of the tool over others is that the tool uses knowledge representation to manage the knowledge of the RE process and its technique, thus assisting development of the most suitable RE process for a software project.*

1 Introduction

The growing complexity of software projects poses a significant challenge for software industry. In the last three decades, researchers in Computer-Aided Software Engineering (CASE) have tried to find solutions to help system developers by providing integrated system development tools that support SE [1]. Numerous CASE tools have been developed which help to improve both quality and efficiency of software development. However, CASE tools helped improve software productivity far less than initially expected [2]. This is because most CASE tool development focuses only on design and programming issues and other aspects of the SE process are considered too difficult to automate and therefore less support is provided. Knowledge-based SE tools are one possible solution since these tools can make use of SE knowledge. Some of the tools developed in the last years provide help in the early stages of SE, such as requirements acquisition, where complete automation is very hard to achieve [3]. These knowledge-based tools have proven to be very useful in the decision-making processes of SE [4-5].

The application of RE techniques and best practices to software projects is still very limited. The likely main reason is that software engineers do not have enough guidance on which techniques to choose for a certain software project. A lot of researchers argue that selection of appropriate SE techniques for software projects is very important [6-7]. Moreover, Glass stresses that industry does not need new process models and techniques but guidelines on how and when to use existing models and techniques in projects of different domains [6]. In order to address this pressing issue, we have proposed in our earlier research a **F**ramework for **R**equirements **E**ngineering p**R**ocess d**E**velopment (FRERE) [8] which provides help for the development of RE process models and the selection of RE techniques for a software project.

The FRERE tool is an implementation of the FRERE framework. The major characteristics of the tool are: (1) The FRERE tool adopts a frame-based representation to store and later on retrieve detailed information about RE techniques. (2) The architecture of the FRERE tool was designed to make it highly scalable, expandable, and easy to integrate into other software engineering tools. This paper presents our research results and experiences on building and using the FRERE tool.

The motivation behind this research is the fundamental assumption that the use of a suitable RE process model that makes use of appropriate RE techniques contributes to the success of software projects and the overall quality of the software product [8-11].

The rest of the paper is organized as follows: Section 2 discusses the knowledge structure and components of the FRERE tool. The overall architecture of the tool and the key components are presented in Section 3. Finally, conclusions and future work are discussed in Section 4.

2 The knowledge structure of FRERE

RE process development and RE techniques selection is a decision making process. Holsapple and Whinston argue that knowledge elicitation, identification and classification are the foundation of effective decision support [12]. We, therefore, have identified and analyzed 26 RE process

31st Annual International Computer Software and Applications Conference(COMPSAC 2007)
0-7695-2870-8/07 $25.00 © 2007 IEEE

models and 46 techniques in our research and analyzed their characteristics to assist the development of customized RE processes and the selection of RE techniques [8]. These process models and techniques are well-defined and can be used in various domains. Table 1 shows the collection of RE techniques that we considered in our research. We are aware that the identified RE techniques are only a subset of currently available RE techniques and we plan to expand our current RE technique library in future.

Our requirements engineering process knowledge base (REPKB) includes knowledge about RE process models, techniques, relationships, and various rules that were developed throughout this research. More specifically, REPKB includes the following libraries:

- Process model library: contains detailed information about RE process models.
- Technique library: contains information about RE techniques.
- Software project case library: contains information about the RE process models and techniques used in previous projects.
- Major CORE [1] assessment model library: contains information about the major CORE assessment model [13]. A CORE is defined as a specific interest or objective of the RE process which needs to be addressed to ensure the development of a well-defined, high-quality requirements specification that is complete, concise, unambiguous and consistent. The major COREs model was developed in our earlier research and is an RE process assessment model.
- RECMM assessment model library: contains information about the RE Process Maturity Model (REPMM) [14]. This model proposed by Sommerville and Sawyer is adapted to the FRERE framework to enhance the overall assessment ability of the framework.
- Suitability assessment model library: contains information about RE technique suitability assessment with respect to the characteristics of a specific software project [15].
- Rule and guideline library: contains information about a collection of dissent and assent rules regarding the usage of RE process models and RE techniques in a software project [8].
- Relationship library: contains information about relationships between RE techniques, RE process models, and the characteristics of software projects.

This paper focuses on the essential knowledge structures of RE techniques as an example of the design of the REPKB and the FRERE tool. The overall structure of the RE technique library is illustrated in Fig. 1. The library is composed of entities and their relationships. All entity types are represented as frames with each frame being a data structure that contains knowledge about a particular object [16]. Frames are flexible and can represent simple or complex objects, entire situations, or a management problem. The syntax used for the definition of the frames in the research is given below:

<Frame> = (<Frame-Identifier>, <Frame-Definition>, "ENDFRAME");

Table 1 Summary of analyzed RE techniques

ID	Technique Name	Most Common Area of Application in the RE Process
1	Brain Storming and Idea Reduction	Elicitation
2	Designer as Apprentice	Elicitation
3	Document Mining (Observation)	Elicitation
4	Ethnography	Elicitation
5	Focus Group	Elicitation
6	Interview	Elicitation
7	Contextual Inquiry	Elicitation
8	Laddering	Elicitation
9	Viewpoint-Based Elicitation	Elicitation (later stage)
10	Exploratory Prototypes (Throw-Away Prototype)	Elicitation, Analysis and Negotiation, Verification and Validation
11	Evolutionary Prototypes	Elicitation, Analysis and Negotiation, Verification and Validation
12	Viewpoint-Based Analysis	Analysis and Negotiation
13	Repertory Grids	Requirements Elicitation
14	Scenario Approach	Requirements Elicitation, Requirements Analysis and Negotiation, Documentation, Verification and Validation
15	Joint Application Design(JAD)	Elicitation
16	Soft Systems Methodology (SSM)	Elicitation
17	Goal-Oriented Analysis	Analysis and Negotiation
18	Viewpoint-Based Documentation	Documentation
19	Future Workshop	Elicitation
20	Representation Modeling	Analysis and Negotiation, Elicitation
21	Functional Decomposition	Analysis and Negotiation
22	Decision Tables	Analysis and Negotiation, Documentation, Verification
23	State Machine	Analysis and Negotiation, Documentation, Verification
24	State Charts (also known as State Diagrams)	Requirements Modeling, Documentation, Verification
25	Petri-nets	Analysis and Negotiation, Documentation, Verification
26	Structured Analysis (SA)	Analysis and Negotiation, Documentation, Verification
27	Real Time Structured Analysis	Analysis and Negotiation, Documentation, Verification
28	Object-Oriented Analysis (OOA)	Analysis and Negotiation, Documentation, Verification
29	Problem Frame Oriented Analysis	Analysis and Negotiation, Documentation, Verification
30	Goal-Oriented Verification and Validation	Verification and Validation
31	ERD (Entity Relationship Diagram)	Documentation
32	Analytic Hierarchy Process (AHP)	Requirements Prioritization
33	Card Sorting	Requirements Prioritization
34	SQFD (Software QFD)	Analysis and Negotiation and Elicitation
35	Fault Tree Analysis	Analysis and Negotiation and Elicitation
36	Structured Natural Language Specification	Requirements Documentation
37	Viewpoint-Based Verification and Validation	Verification and Validation
38	Unified Modeling Language (UML)	Documentation, Analysis and Negotiation, Verification
39	Z	Documentation, Analysis, Verification
40	LOTOS	Documentation, Analysis, Validation
41	Spec Description Language (SDL)	Documentation, Analysis, Validation
42	XP (Extreme Programming)	Elicitation, Analysis and Negotiation, Documentation, Validation
43	Formal Requirements Inspection	Requirements Verification and Validation
44	Requirements Testing	Requirements Verification and Validation
45	Requirements Checklists	Requirements Verification and Validation
46	Utility Test	Requirements Verification and Validation

[1] CORE is an abbreviation for Concern of Requirements Engineering.

<Frame-Definition> = {(<Slot-name> ":" <Facet>)}$^+$;
<Facet> = <Value> | { (<Slot-name> ":" <Facet>) } ;
<Value> = <string> | <integer> | <real> | (<Value> {","
<Value>});
<Frame-Identifier>=(<FRAME>, ":", <Frame-name>)
<Frame-name>=<string>;
<Slot-name> = <string>;
Here, {x}$^+$ means "x" repeats 1 or more times.

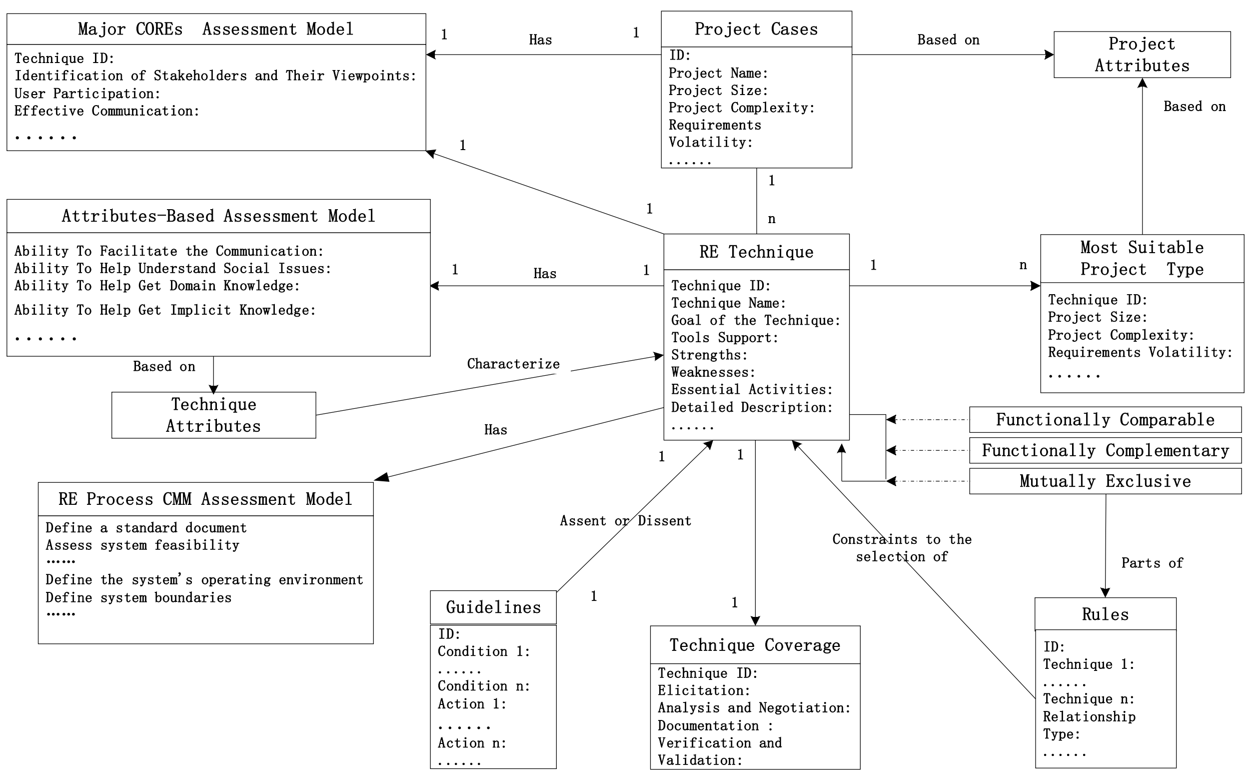
Different types of relationships between entities are contained in the REPKB. Some of the more complex ones are described in the following:

- "has" relationship. Each RE technique entity "has" information about their assessment results using major COREs and the Attribute-Based assessment model [8]. This information reveals the capability of each RE technique according to attributes defined in the two models.
- "Assent" and "Dissent" relationships indicate that the guidelines either support or discourage the use of a particular technique.
- The relationships between RE techniques such as "functionally comparable", "functionally complementary" and "mutually exclusive" relationships help determine the usefulness of combining several RE techniques [8].

As shown in Fig. 1, knowledge about RE techniques is organized into five categories to facilitate their selection which includes basic knowledge about RE techniques, assessment information about each technique with three assessment models (RECMM, COREs, and Suitability Model [15]), guidelines for the use of RE techniques, project cases for using the techniques, and relationships between techniques.

To accommodate the complexity of the knowledge structure of RE process models and techniques, frames are used as a mechanism to organize RE process models and technique knowledge, and to effectively support RE process development. Based on further analysis of the RE process models and techniques, we determined that the knowledge about RE process models and techniques requires a multi-dimensional schema for efficient storage and retrieval [8] For example, the knowledge about RE techniques can be represented in the following three different types of frame (see Fig. 1):

- RE Techniques frame: This type of frames includes basic information of an RE technique, such as its name, goals, strengths, weaknesses. Such a frame provides general help for the selection of RE techniques. Each technique is represented by one frame.
- Techniques Coverage frame: This type of frame



Notes:
1. Only slot names in each frame are shown.
2. "......" indicates that there are still other slots in this frame.
3. The "⟶" indicates the relationship among each entity (frame).
4. In attribute-based assessment frame, each slot is an attribute of a technique.
5. In Major COREs Assessment frame, each slot is a major CORE.
6. "⇢" indicates that some techniques have three relationships: functionally comparable, functionally complementary, and mutuallly exclusive relationship

Figure 1 Structure of RE techniques knowledge library

Focus Group Coverage

Instance of: Technique Coverage
Technique ID: 5
Elicitation: Y
Analysis and Negotiation: N
Documentation: N
Verification and Validation: N
Management: N
Tool Support: Y

(a)

Notes:
1. The attributes in each frame are shown in bold. The attribute values are shown in normal font.
2. In the coverage frame, "Y" indicates that the technique covers the stage; "N" indicates that the technique does not cover the stage.
3. For the frame Most Suitable Project Type: "Any" indicates this attribute could have any value within the valid range.

Focus Group

Instance of: RE Technique
Technique ID: 5
Technique Name: Focus Group
Tools Support: No specific tool available for this technique
Strengths: Various viewpoints from different stakeholders can be elicited. Focus on a particular problem is possible.
Weaknesses: Workshops tie up stakeholders, often for several days at a time. Most effective for medium-sized systems. A trained facilitator is needed.
Essential Activities:
(1). Formulate the meeting focus.
(2). Identify and train facilitators.
(3). Generate guidance for the "Focus Group" meeting.
(4). Decide who will participate in the meetings and inform participants.
(5). Make arrangements for the meeting, such as equipment, refreshments.
(6). Schedule and conduct the group meeting, record the meeting using a tape recorder and/or a systematic recording form.
(7). Prepare data and analyze results.
(8). Generate report.
Detailed Description: A Focus Group is a structured group interview process, conducted for the purpose of obtaining detailed information about a particular topic, product, or issue.

(b)

Focus Group Most Suitable Project Type

Instance of: Most Suitable Project Type
Technique ID: 5
Project Size: Small to Medium
Project Complexity: Low to Medium
Requirements Volatility: Low to Medium
Organization Customer Relationship: Any
Project Category: Any
Degree of Safety Criticality: Any
Quality Standard: Medium to Very High
Product Type: Any
Time Constraints: Very Low to Low
Cost Constraints: Very Low to Low
Team Size: Small to Medium
Acquaintance with Domain: Very Low to Medium
Knowledge of RE: Any
Degree of Knowledge of Requirements: Very Low to Medium

(c)

Figure 2 Examples of frame definitions for the RE technique "Focus Group"

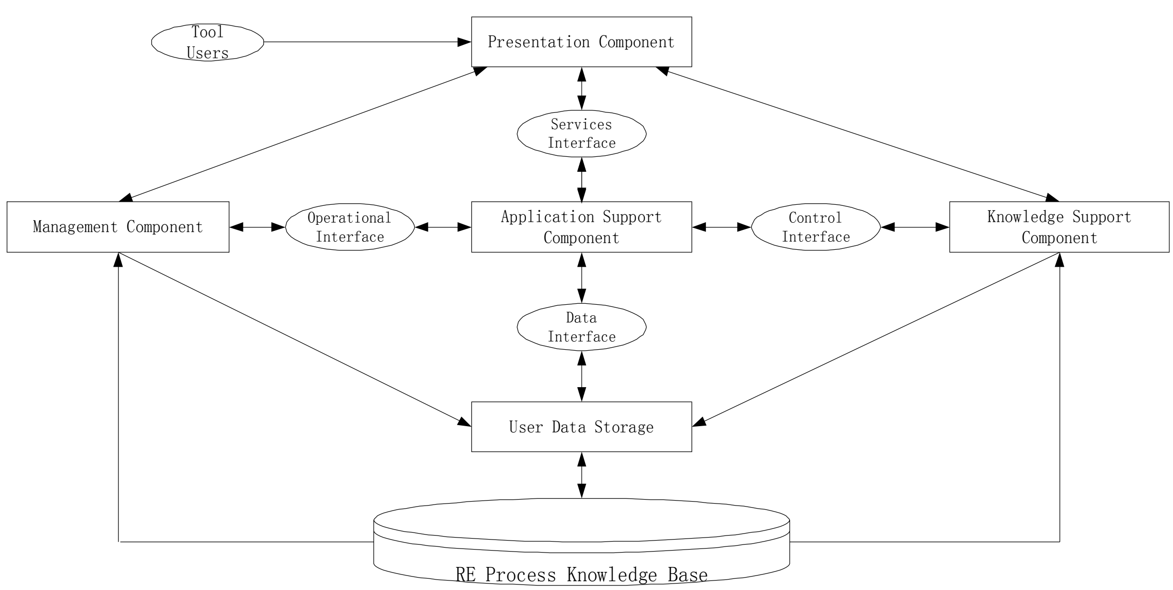
includes the information about the parts of the RE process that are covered by a certain technique.

- Most Suitable Project Type frame: This type of frame stores information on the suitability of RE techniques for certain project types. The project types are characterized by project attributes, such as project size, project complexity, and requirements volatility [8]. Even though most techniques can be used in any type of software project, some techniques are likely more suitable for certain types of projects than others. The knowledge represented in these frames provides high-level guidance for the selection of RE techniques.

The RE technique "Focus Group" is described in Fig. 2((a), (b), (c)) using the three frames discussed above.

Frame-based and rule-based reasoning mechanisms help retrieve and process knowledge contained in REPKB. More detailed information about the reasoning mechanisms can be found in [8].

An industrial case study which used our REPKB suggests that the structure of the knowledge base is suitable for the management of RE techniques and process model knowledge[8].



Legend: (1) " ←→ " indicates the flow of data and control in both directions (2) " → " indicates the flow of data and control in one direction

Figure 3 The Overall Structure of the Tool FRERE

0-7695-2870-8/07 $25.00 © 2007 IEEE

3. Overall Architecture of the FRERE Tool

FRERE uses an extended version of the architecture of the Generalized Attribute-centred RE Process tool (GAREP) [17]. The overall structure of the tool is illustrated in Figure 3 and consists of six components:

1. Presentation component. This component takes care of the input and display of relevant information to the user (see Section 3.1).
2. Application support component. This part supports initializing, scheduling, and managing the tool. It has four interfaces that connect to the service interface, control interface, data interface, and operation interface.
3. Knowledge support component. This component is used to manage the REPKB and schedule the communication between application support and presentation support. It also manages operations related to user data storage, such as data completeness checks.
4. User data storage. This is a database that stores user information, project information, previously used RE process models, and related information.
5. RE Process Knowledge Base (REPKB). Information related to RE process knowledge such as RE process models, techniques, good practices, guidelines and rules are stored in the REPKB.
6. Management component. The system management component helps tool users to configure the tool, control its conditions, and manage user authorization and the contents of the REPKB.

Each component has its specific roles during RE process development. Some of the components are explained in more detail in the next two subsections.

3.1 Presentation Component

The presentation component of the FRERE tool supports the involvement of different users including requirements engineers, process engineers and trainees. It defines how the information is displayed to them.
The presentation component consists of the following eight types of user interfaces:

- Interfaces for providing methodological guidelines.
- Interfaces for project definition. Fig. 4 shows an interface screen shot of the project definition for the Port Scheduling System (PSS) - a software project in which the FRERE tool was used [8].
- Interfaces for activities definitions. Seven operations (Tailoring, Selecting, Adding, Deleting, Splitting, Merging, and Modifying [8]) applicable to the activities were defined. The *Adding* operation for the definition of an activity "*Identify the viewpoints by referencing to the viewpoint class hierarchy*" in the case of the PSS project is shown in Fig. 5.
- Interfaces for technique selection. This type of user interface allows the user to select the process model, building blocks, techniques, and tools based on the recommendations of the tool.
- Interfaces for process Assessment. This type of user interface allows the user to assess the newly developed

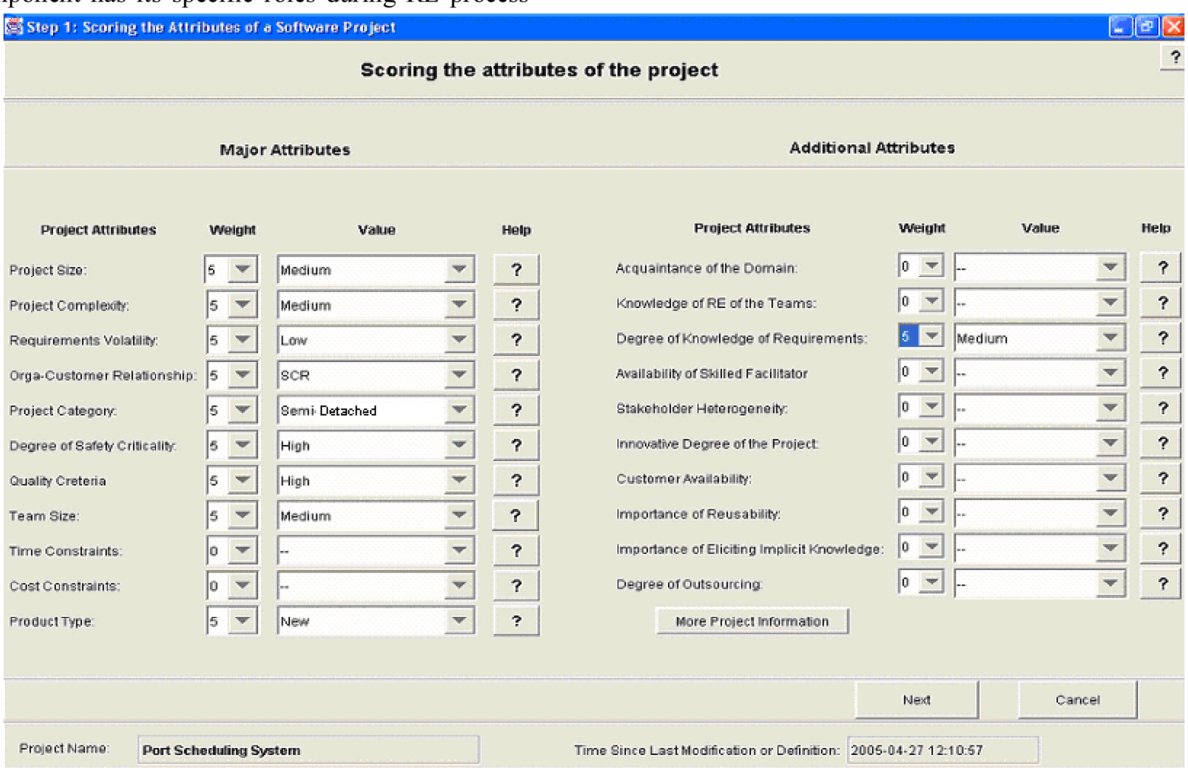


Figure 4 Project Definition User Interface

0-7695-2870-8/07 $25.00 © 2007 IEEE

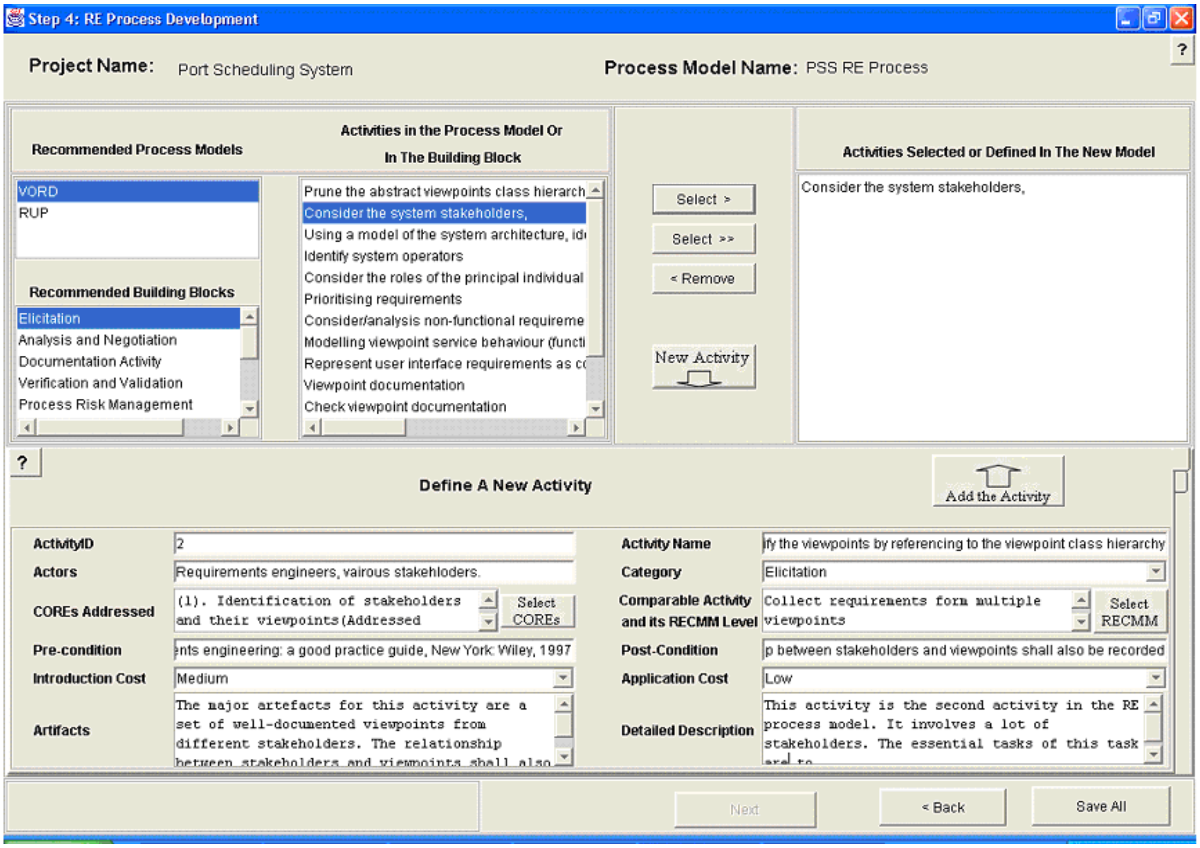


Figure 5 "Adding" Operation for an Activity Definition

RE process model based on predefined process assessment models.

- Interfaces for comparison of techniques and process models. In order to help the process developer, RE techniques and RE process models can be compared in detail using the tool.
- Interfaces for process model presentation. The overall process model can be presented in its entirety to the user through this type of interface.
- Interfaces for heuristic guidelines. During RE process development, information is provided by the tool to explain the meaning of terminology, attributes and procedures.

In order to perform its tasks, the representation component communicates with application support to submit the user requests and to collect the results.

3.2 Knowledge Support

Knowledge support is another big component of the tool. It provides fundamental support for RE process development with the help of the REPKB. REPKB provides the following functions:

- Rules management. Different kinds of rules were defined in the REPKB to help RE technique selection [8]. The tasks for rules management include the scheduling, controlling, and managing of rules.
- Guidelines management. Five kinds of guidelines for assisting RE process development are provided: general guidelines, building block guidelines, process model guidelines, techniques guidelines, and methodological guidelines. The tasks of guidelines management include scheduling, controlling, and managing guidelines.
- Case-based reasoning (CBR) management. One of the important functions of the tool is to facilitate the decision making process during RE process development. The tasks of case-based reasoning management are to schedule, control and manage the case-based reasoning process.
- Process assessment management. RE process assessment uses three models: major COREs assessment model [18], RECMM model [14] and Suitability Assessment model [15]. The tasks of process assessment management are to schedule, control and manage the evaluation models.
- Process models comparison management. The tasks of this function include controlling, and managing the comparison of RE process models.
- Techniques comparison management. This is a mechanism to control and manage the comparison of RE techniques.
- The knowledge support component in this architecture is designed to assist the representation, processing, and retrieval of RE process knowledge in such a way that

the tool users can make maximum use of the knowledge contained in REPKB. During the runtime of the tool, knowledge support cooperates with application support to analyze the information coming from the tool users. Based on the knowledge in REPKB, knowledge support checks the input of users and the information that is already in the REPKB. At the end of the analysis, the results are queried by application support to provide feedback to the tool user.

3.3 Data Storage

During RE process development, information about the project, recommendations based on the Case-Based Reasoning (CBR), decision support, newly provided information, etc. have to be stored and maintained over time. In order to manage this information efficiently, a user database is defined.

The user data storage provides all tool users with up-to-date information about software projects and the RE process models that were developed and used in the software projects.

4. Conclusion and future work

This paper presented the overall architecture of the FRERE tool and the structure of REPKB, in which the knowledge of RE process models and techniques is stored and managed. Several typical components and the functionality of each of the components are explained. Although each part has its own responsibilities, all the parts are orchestrated to support requirements engineers, incremental development, and reuse of RE process knowledge. At the same time, the tool can also provide methodological support for RE process development by offering decision support to requirements engineers for the selection of the most suitable techniques and practices for a software project.

The first prototype of the FRERE tool was implemented in Java and Access and was successfully used during the development of the RE process model and selection of techniques for the development of a Port-Scheduling System (PSS). The result of this case study strongly indicates that the FRERE tool has contributed to the success of the PSS project [8].

Currently, a new version of the FRERE tool is under development which will be implemented with Java and MySql. The migration of the REPKB from Access to MySql has already been done, refactoring of the application support and implementing new functionality for the application support will be the next task.

Our future work will focus on further improving the FRERE framework and including more RE process models and techniques into REPKB. The tool will be validated using an extensive case study.

References

1 Case AF, Computer-aided Software Engineering Technology for Improving Software Development Productivity, ACM SIGMIS, Vol. 17 , Issue 1 ,1985, pp: 35 – 43,.

2 Huff, CC, Elements of a realistic CASE tool adoption budget, Communications of the ACM, Vol. 35 , Issue 4, April 1992 , pp: 45-54

3 Sommerville I. Software engineering, Seven Edition, Addison-Wesley Publishers Limited, 2004

4 Rich, C.; Waters, R.C.;Knowledge intensive software engineering tools Knowledge and Data Engineering, IEEE Transactions on Vol. 4, Issue 5, Oct. 1992, pp(s):424 - 430

5 Ruhe G. SE Decision Support - Methodology and Applications. In: 'Innovations in Decision Support Systems' (ed. by Tonfoni and Jain), International Series on Advanced Intelligence, Vol. 3, 2003, pp 143-174.

6 Glass RL, Matching Methodology to Problem Domain, Communications Of The ACM, Vol. 47, No. 5, May 2004, pp. 19-21

7 Jones C Software Assessment, Benchmarks, and Best Practices. Addison-Wesley, 2000

8 Jiang L, A Framework For Requirements Engineering Process Development, PhD Thesis, University of Calgary, Sep. 2005

9 Emam KE, Birk A, Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability, IEEE Trans. on Software Engineering, Vol 26, NO. 6, Jun. 2000.

10 Damian, D.; Chisan, J An Empirical Study of the Complex Relationships between RE Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management, IEEE Transactions on SE, Vol. 32, Issue 7, July 2006, pp:433-453

11 Kotonya G. and Sommerville I. Requirements Engineering, Processes and Techniques, John Wiley & Sons Ltd, 1998

12 Holsapple CW and Whinston AB., Decision Support Systems: A Knowledge-Based Approach, West Publishing, Company, Minneapolis/St Paul, 2000, ISBN 0-314-06510-5

13 Jiang L., Eberlein A. and Far B. H. Evaluating the RE Process Using Major Concerns, Proceedings of the IASTED International Conference on Software Engineering, Feb. 16-19, 2004, Innsbruck, Austria

14 Sommerville I. and Sawyer P. Requirements Engineering: A Good Practice Guide, New York: Wiley, 1997

15 Jiang L, Eberlein A, A Case Study For Technique Suitability Assessment Based on Project Attributes, 14th Annual IEEE International Conference on the Engineering of Computer Based Systems, 2007 (ECBS), Tucson, AZ USA

16 Turban E., Aronson JE., Decision Support Systems and Intelligent System, Sixth Edition, Prentice Hall. Upper Saddle River, New Jersey, 2001

17 Zhang Q. Support for the Requirements Engineering Process, Master Thesis. Department of Electrical and Computer Engineering, University of Calgary, July, 2003

18 Jiang L, Eberlein A, Far BH. A Methodology for RE Process Development, ECBS 11th IEEE International Conference on the Engineering of Computer-Based Systems, Brno, Czech Republic, May, 2004

0-7695-2870-8/07 $25.00 © 2007 IEEE