Requirements Eng (2008) 13:117–146
DOI 10.1007/s00766-007-0060-2

ORIGINAL ARTICLE

# A case study validation of a knowledge-based approach for the selection of requirements engineering techniques

**Li Jiang · Armin Eberlein · Behrouz H. Far**

Received: 9 November 2006 / Accepted: 22 October 2007 / Published online: 22 November 2007
© Springer-Verlag London Limited 2007

**Abstract** Requirements engineering (RE) is a critical phase in the software engineering process and plays a vital role in ensuring the overall quality of a software product. Recent research has shown that industry increasingly recognizes the importance of good RE practices and the use of appropriate RE techniques. However, due to the large number of RE techniques, requirements engineers find it challenging to select suitable techniques for a particular project. Unfortunately, technique selection based on personal experience has limitations with regards to the scope, effectiveness and suitability of the RE techniques for the project at hand. In this paper, a Knowledge-based Approach for the Selection of Requirements Engineering Techniques (KASRET) is proposed that helps during RE techniques selection. This approach has three major features. First, a library of requirements techniques was developed which includes detailed knowledge about RE techniques. Second, KASRET integrates advantages of different knowledge representation schemata and reasoning mechanisms. Thus, KASRET provides mechanisms for the management of knowledge about requirements techniques and support for RE process development. Third, as a major decision support mechanism, an objective function evaluates the overall ability and cost of RE techniques, which is helpful for the selection of RE techniques. This paper makes not only a contribution to RE but also to research and application of knowledge management and decision support in process development. A case study using an industrial project shows the support of KASRET for RE techniques selection.

**Keywords** Knowledge management · Reasoning · Requirements engineering · Techniques · Evaluation · Decision support

## 1 Introduction

Requirements engineering (RE) is a critical phase during software development and is a major contributor to software quality [1–5]. Industry increasingly recognizes the importance of using good RE processes and appropriate RE techniques when developing software systems [6, 7] to achieve high software quality. Glass et al. [8] stressed that we need a way to choose the most appropriate software development methodology for the task at hand. In addition, researchers emphasize the necessity of adopting proper requirements engineering techniques in order to elicit, model, document, verify and validate requirements so that a high quality specification can be derived [9–16]. Davis [17] states that knowing which technique to apply to a given problem is necessary for effective requirements analysis. In our research, numerous RE techniques have been identified and studied. Some techniques have similar

L. Jiang
School of Computer Science,
The University of Adelaide,
Adelaide, SA 5005, Australia
e-mail: jiang.li@adelaide.edu.au

A. Eberlein (✉)
Computer Engineering Department,
American University of Sharjah,
P.O. Box 26666, Sharjah, United Arab Emirates
e-mail: eberlein@ucalgary.ca

B. H. Far
Department of Electrical and Computer Engineering,
University of Calgary, 2500 University Drive,
N.W. Calgary, AB, Canada T2N 1N4
e-mail: far@ucalgary.ca



المنارة للاستشارات
www.manaraa.com

functionality but different complexity, while others are functionally complementary [18]. This variety makes the selection of a suitable combination of RE techniques for a specific project a challenging task. Moreover, a review of literature and practices in industry has shown that most companies select RE techniques in an ad hoc manner [13, 15, 18]. There is a big gap between the availability of RE techniques and their application in practice. This is likely due to the following reasons:

- RE practitioners are not aware of various RE techniques.
- No comprehensive guidance is available for the selection of RE techniques for a specific project.
- There is no overall support for the selection of RE techniques for all stages of the RE process.

In order to provide a solution for this problem, a new knowledge-based approach, called KASRET, was developed in our research to provide a mechanism to facilitate the selection of RE techniques. The objective of this research is to develop a knowledge-based decision-support system for the selection of RE techniques for a specific software project. KASRET helps select RE techniques for the overall RE process rather than only one stage of the process. This is one of the distinguishing features of our research. We use the RE process model proposed by Kotonya and Sommerville [1], which includes the following four stages: requirements elicitation, requirements analysis & negotiation, requirements documentation, and requirements validation. Requirements management is involved in all four stages of the RE process. Based on this model, RE techniques are classified into five categories in this research: elicitation, analysis & negotiation, documentation, verification & validation, as well as requirements management. Requirements management tools are essential for the success of software projects [19]; therefore, they are considered compulsory components to be included in all recommendations. However, the selection of a particular requirements management tool is not yet included in this paper and is subject to further research.

In summary, the major advantages of the proposed approach are summarized as follows: First, a RE techniques library was developed which includes extensive knowledge about RE techniques. This library is part of the overall Requirements Engineering Process Knowledge Base (REPKB) developed in our research. Second, different knowledge representation schemata and reasoning mechanisms are used in the library. A Case-Based Reasoning (CBR) mechanism is used to store and retrieve similar cases of previous projects. A frame-based reasoning mechanism is used to store and retrieve knowledge about a variety of RE techniques and to provide the overall guidelines of KASRET, consequently providing more support for the selection of techniques. Third, different mechanisms for RE technique analysis are provided based on RE knowledge in the library, such as technique clustering and an objective function. These mechanisms provide information for the analysis of RE techniques at a detailed level. This, in turn, facilitates techniques selection according to the characteristics of the new project. An industrial case study showed that the KASRET approach provides significant support for RE technique selection. This research is part of a larger project that works on a framework for RE process development [18, 20] and makes a contribution to requirements engineering as well as knowledge engineering and decision support in general.

It should be mentioned that one of the key assumptions of this research is that the usage of appropriate RE techniques leads to high quality requirements specifications [17, 21–23]. This is a fundamental assumption of the RE community.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 discusses knowledge types and their use for representing RE techniques. The reasoning mechanism built on the RE technique knowledge base is given in Sect. 4. The overall process of KASRET is presented in Sect. 5. An industrial case study is presented in Sect. 6. The major findings, final conclusion and future work are discussed in Sect. 7.

## 2 Related work

The most related research in the area of supporting RE techniques selection is a framework, called ACRE, proposed by Maiden and Rugg [21]. This framework offers help for the selection of 12 acquisition techniques which have been described in detail including: preconditions for its use and perceived strengths and weaknesses. ACRE also included six facets which help technique selection, namely purpose of requirements to be elicited, knowledge types required for using the techniques, ability for internal filtering of knowledge, observable phenomena required, acquisition context, and techniques interdependencies. Overall, ACRE provides guidance for techniques selection by using a question-driven approach. Another related approach is the high-level wish-list of seventy requirements for RE techniques proposed by Macaulay. The wish-list is developed based on the needs that support the RE process, human communication, knowledge development, documentation and management [12, 24]. Mapping needs of RE processes to Macaulay's wish-list for RE techniques does help during technique selection if enough well-understood techniques are available. Hickey and Davis [22,



المنارة للاستشارات
www.manaraa.com

25] developed a mathematical model of the requirements elicitation process. Besides showing the critical role of knowledge used in the process, this model also helps to improve the understanding of the elicitation process, and how elicitation techniques are selected. Recently, Tsumaki and Tamai [26] also proposed a framework with the aim to match RE techniques to project characteristics. In their framework, Tsumaki and Tamai attempt to characterize requirements elicitation techniques into two dimensions (elicitation operation types, and the target object types), and use the classification as the base for selection of appropriate techniques at the time of starting a project as well as at the time of recognizing a situation change in the project such as a change in the project nature or encountering an obstacle in defining a suitable set of requirements. Yet, the objectives of the framework have not been achieved as the granularity of the techniques and the characteristics of the project is very course. The framework does not provide sufficient information to help with RE technique selection.

There are other approaches that are relevant; however, they only partially address the issues of techniques selection. Kotonya and Sommerville [1] proposed eight high level properties of RE techniques that help to differentiate between requirements techniques. Bickerton and Siddiqi proposed a framework for the classification of RE techniques [27]. This framework is built on social assumptions made about organizations and focuses on the nature of society and the expected role of requirements engineers. A decision aid for requirements engineers is discussed in the form of a table that classifies a representative sample of RE techniques. In order to deal with the uncertainties in determining requirements for information systems, Davis [28] proposed a five-step approach which includes four strategies and a set of RE techniques that can be used. The four strategies for getting requirements are: (1) asking, (2) deriving from an existing system, (3) synthesis from characteristics of the utilizing system, and (4) discovering from experimentation with an evolving information system application. Linking RE technique to the four strategies helps to select RE techniques. Browne and Ramesh [29] also proposed a way for the selection of requirements elicitation techniques built on the human cognitive model. In [30], after discussing several techniques which can be used in the RE process, and various requirements presentation styles in detail, Lausen then briefly explains an idea for RE technique selection using a matrix that contains RE techniques and objectives that needed to be addressed in an RE process.

Method engineering, on the other hand, provides approaches that help the development or adaptation of existing methodologies to the problem domain. It relies on extensive experiences of method engineers who can build a new methodology based on a collection of existing methods [31–33]. It targets the development of methodologies for large information system development.

As can be seen from the above summaries, there appears to be a lack of research into the selection of RE techniques for the whole RE process. Existing research related to the classification and identification of the characteristics of RE techniques is limited. Typically, previous research only looked at RE technique selection for individual parts of the RE process (such as the requirements elicitation phase), but the characteristics of RE techniques and software projects are not explored in depth. The result is that only limited help can be provided for the selection of RE techniques for a software project. Moreover, we have not found any related work that provides support for RE technique selection using knowledge-based decision support which is a key feature of our research.

## 3 RE techniques library

Building a well-structured knowledge-library is the first step to achieve effective support of RE techniques selection. As success factors to ensure better usage of the state of the art technology in practice, Reifer [34] suggested that rules associated with using the technology shall be documented, and guidelines for use shall be available, and the body of knowledge related to the technique shall be codified and available in a functional form. Thus, firstly, we analyze the 46 RE techniques (see Table 1) that were identified in our previous research [18] in great level of detail, and classify all the knowledge related to these RE techniques into different categories to facilitate effective knowledge management. As a result of the research, information about these RE techniques, their attributes, weaknesses and strengths were identified and stored in the RE Techniques Knowledge Library (RETKL), to be used during RE technique selection. Currently, RETKL is part of the prototype of the KASRET tool. RETKL currently includes 26 process models and 46 techniques. In the following subsection, the detailed structure of the RE technique library, the major components of the library, and the knowledge presentation mechanism used in the library will be presented. The components in the library serve as essential building blocks that are used in the KASRET methodology which will be presented in Sect. 5.

### 3.1 The overall structure of RETKL

The overall structure of RETKL is shown in Fig. 1 and is composed of entities and their relationships. All entities are represented as frames with each frame being a data structure that contains knowledge about a particular object [35].



المنارة للاستشارات
www.manaraa.com

**Table 1** RE techniques considered during this research

| ID | Technique name | Most common area of application in the RE process |
|---|---|---|
| 1 | Brain Storming and Idea Reduction | Requirements Elicitation |
| 2 | Designer as Apprentice | Requirements Elicitation |
| 3 | Document Mining | Requirements Elicitation |
| 4 | Ethnography | Requirements Elicitation |
| 5 | Focus Group | Requirements Elicitation |
| 6 | Interview | Requirements Elicitation |
| 7 | Contextual Inquiry | Requirements Elicitation |
| 8 | Laddering | Requirements Elicitation |
| 9 | Viewpoint-Based Elicitation | Requirements Elicitation (later stage) |
| 10 | Exploratory Prototypes (Throw-Away Prototype) | Requirements Elicitation, Requirements Analysis & Negotiation, Requirements Verification & Validation |
| 11 | Evolutionary Prototypes | Requirements Elicitation, Requirements Analysis & Negotiation, Requirements Verification & Validation |
| 12 | Viewpoint-Based Approach | Requirements Analysis & Negotiation |
| 13 | Repertory Grids | Requirements Elicitation |
| 14 | Scenario-Based Approach | Requirements Elicitation (later stage), Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 15 | Joint Application Design (JAD) | Requirements Elicitation |
| 16 | Soft Systems Methodology (SSM) | Requirements Elicitation |
| 17 | Goal-Oriented Analysis | Requirements Elicitation, Requirements Analysis & Negotiation, Requirements Documentation |
| 18 | Viewpoint-Based Documentation | Requirements Documentation |
| 19 | Future Workshop | Requirements Elicitation |
| 20 | Representation Modeling | Requirements Elicitation, Requirements Analysis & Negotiation |
| 21 | Functional Decomposition | Requirements Analysis & Negotiation |
| 22 | Decision Tables | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 23 | State Machine | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 24 | State Charts (also known as State Diagrams) | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 25 | Petri-nets | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 26 | Structured Analysis (SA) | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 27 | Real Time Structured Analysis | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 28 | Object-Oriented Analysis (OOA) | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |



المنارة للاستشارات
www.manaraa.com

**Table 1** continued

| ID | Technique name | Most common area of application in the RE process |
|---|---|---|
| 29 | Problem Frame Oriented Analysis | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 30 | Goal-Oriented Verification and Validation | Requirements Verification & Validation |
| 31 | Entity Relationship Diagram (ERD) | Requirements Documentation |
| 32 | Analytic Hierarchy Process (AHP) | Requirements Analysis & Negotiation |
| 33 | Card Sorting | Requirements Analysis & Negotiation |
| 34 | Software Quality Function Deployment (SQFD) | Requirements Analysis & Negotiation, Requirements Elicitation |
| 35 | Fault Tree Analysis | Requirements Analysis & Negotiation, Requirements Elicitation |
| 36 | Structured Natural Language Specification | Requirements Documentation |
| 37 | Viewpoint-Based Verification and Validation | Requirements Verification & Validation |
| 38 | Unified Modeling Language (UML) | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 39 | *Z* | Requirements Analysis, Requirements Documentation, Requirements Verification & Validation |
| 40 | Specification and Description Language (LOTOS) | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 41 | Specification and Description Language (SDL) | Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 42 | Extreme Programming (XP) | Requirements Elicitation, Requirements Analysis & Negotiation, Requirements Documentation, Requirements Verification & Validation |
| 43 | Formal Requirements Inspection | Requirements Verification & Validation |
| 44 | Requirements Testing | Requirements Verification & Validation |
| 45 | Requirements Checklists | Requirements Verification & Validation |
| 46 | Utility Test | Requirements Verification & Validation |

Frames are flexible and can represent simple objects, complex objects, entire situations, or a management problem as a single entity. The syntax of the frames used in this paper is defined as follows:

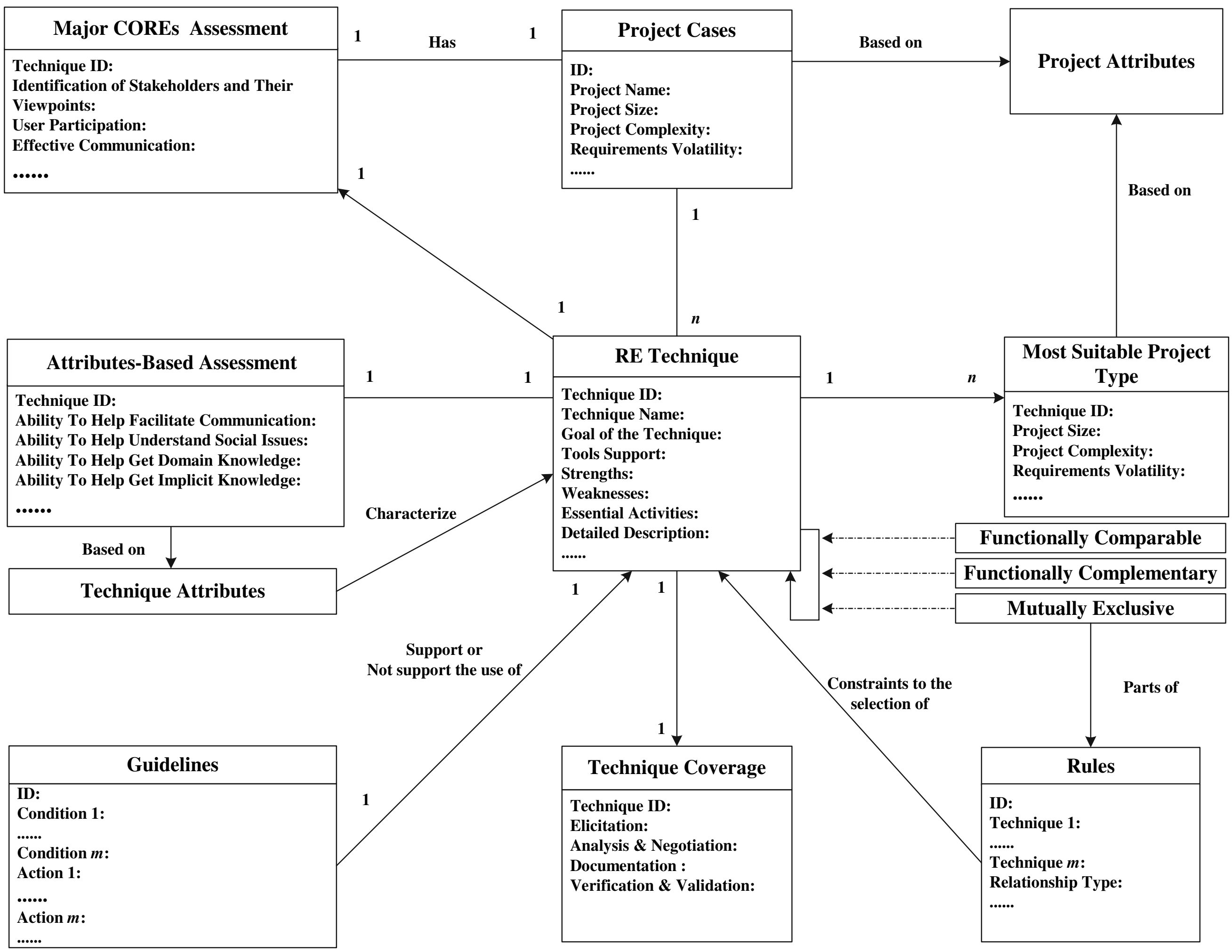
```
<Frame > = (<Frame-Identifier > , <Frame-Definition > , ''ENDFRAME'');
<Frame-Definition > = {(<Slot-name > '':'' <Facet > )}+;
<Facet > = <Value > |{(<Slot-name > '':'' <Facet > )};
<Value > = <string > |<integer > |<real > |(<Value > {'':'' <Value > });
<Frame-Identifier > = (<FRAME > , '':'', <Frame-name > )
```

Here, $\{a\}^+$ means “a” repeats 1 or more times.

There are different types of relationships between entities in RETKL. Some of the more complex ones are described in the following:



المنارة للاستشارات
www.manaraa.com




Notes:
1. Only slot names in each frame are shown in the figure for simplicity.
2. "......" indicates that there are still other slots in this frame.
3. The lines with arrow indicate the relationship between entities (frames).
4. In the attribute-based assessment frame, each slot is an attribute of a technique.
5. In the Major COREs Assessment frame, each slot is a major CORE.
6. *m* represents the *mth* item in the frame which indicates that there will be *m* items in the frame.
7."- - - - ▸" indicates that the techniques have three relationships: functionally comparable, functionally complementary, and mutually exclusive relationship
8. The numbers shown at the two ends of the line in the figure indicate the relationship between the two entities, such as 1 to 1, or 1 to *n* relationship.

**Fig. 1** Structure of RE techniques knowledge library

- The "has" relationship. For example, in Fig. 1 each Project Case has a major COREs[1] assessment of the RE techniques that were used in the project. The assessment information indicates which major RE concerns were addressed by the software processes used in the various project cases stored in the RETKL. The "has" relationship provides specific links between a project case and the major COREs addressed by the techniques used in the project which helps during RE technique selection for a new software project.
- The "functionally comparable", "functionally complementary" and "mutually exclusive" relationships represent possible relationships between two techniques. These relationships will be described in Sect. 3.2.2.3 (see Table 3).
- "Supportive" and "Not Supportive" relationships indicate that the guidelines either support or do not support the use of a particular technique.

As shown in Fig. 1, knowledge about RE techniques is organized into five categories in order to facilitate the selection of RE techniques: basic knowledge about RE techniques, advanced knowledge about RE techniques, guidelines for the use of RE techniques, information about

[1] CORE stands for Concerns of Requirements Engineering Process. The major COREs model was developed in our earlier research [6, 18] and is an RE process assessment model which provides information about the overall quality of the developed RE process. It can also be used to assess the overall capability of RE techniques. More details on this model will be given in Sect. 3.2.2.2.

Springer
www.manaraa.com

the RE techniques that were used in project cases, and relationships between RE techniques.

The industrial case study presented in Sect. 6 used the RETKL and suggests that the structure of the knowledge base is an effective means for the management of RE technique knowledge.

## 3.2 RE technique knowledge and its representation

RE techniques selection is a decision making process. Holsapple and Whinston [36] argue that knowledge elicitation, identification and classification are the foundation of effective decision support. Therefore, we have identified a set of RE techniques and documented the knowledge about these techniques and experiences of using them in industry. The set of RE techniques presented in Table 1 (see [18, 37] for a detailed list of references for each technique) are representative techniques for the different phases of the RE process and their selection was based on a set of criteria, such as maturity[2] of a technique, industrial awareness and experience with a technique, coverage of RE process, etc. [18]. Furthermore, we also analyzed and captured characteristics of the identified RE techniques. To ensure effective support for selecting techniques for the entire RE process, the techniques identified in the research cover all phases of the RE process, from requirements elicitation to requirements verification and validation (see Table 1). It is worth mentioning that the identified RE techniques are only a subset of all currently available RE techniques and we plan to further expand our current library in the future. Each technique might cover more than one phase of the RE process as shown in Table 1. Additionally, the granularity of the techniques presented here is not the same which reflects the reality of RE techniques. For example, Brain Storming is a simple technique compared to JAD which has a very high level of complexity.

Once the RE techniques knowledge is elicited and documented, using adequate ways and to represent it to effectively process and retrieve it becomes an important issue. In the following subsections, we show how the knowledge about RE techniques is represented in this research.

### *3.2.1 Representation of basic knowledge*

The knowledge structure of RE techniques is very complex, but frames provide an effective mechanism to organize RE knowledge and support techniques selection. Basic knowledge about RE techniques can be represented in the following three different types of frames (see also Fig. 1):

- RE Techniques Frame: Each technique is represented by one frame which includes basic information of the RE technique, such as its name, goals, strengths, weaknesses. Such a frame provides general help for the selection of RE techniques.
- Techniques Coverage Frame: This frame indicates what parts of the RE process are covered by a certain technique. For instance, if an RE technique covers elicitation, analysis, documentation and management but not verification, then the coverage of the technique is 4/5. The knowledge stored in these frames can be used to select a technique for a specific part of the RE process.
- Most Suitable Project Type Frame: This 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 [18]. 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 using the three frames discussed above.

### *3.2.2 Representation of advanced knowledge*

*3.2.2.1 Guidelines and rules* The guidelines for the use of RE techniques are derived from books, research papers and practitioners with real-life experience of using the techniques. These guidelines are consistent with information provided in the Most Suitable Project Type frame, yet have a different focus. The information provided in a Most Suitable Project Type frame shows the project types for which a particular RE technique is suitable. A guideline describes the certainty with which a technique is or is not recommended based on the given project attributes.

There are two guidelines for each technique: "Assent Guidelines" and "Dissent Guidelines". An Assent Guideline of a technique states in what situations the technique will work well. A Dissent Guideline states in what situations the technique will not be helpful. Two examples of guidelines are shown in Fig. 3.

Moreover, the following two types of rules were defined and stored in the Rule frames:

[2] A mature technique refers to a technique that is well-defined and has systematic steps or a well-defined collection of notations, is well-organized and documented, and has been used in industrial projects.



المنارة للاستشارات
www.manaraa.com

**Fig. 2** Examples of frame definitions

**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. Will be more 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, food and drinks.
(6) Schedule and conduct the group meeting, record the meeting using tape recorder and/or a systematic recording form.
(7) Prepare data and analyze results.
(8) Generate report.

**Detail 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.

**(a) RE Technique Frame**

**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

**(b) Most Suitable Project Type Frame**

**Focus Group Coverage**

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

**(c) Technique Coverage Frame**

**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.

**Fig. 3** Examples of guidelines

Assent Guideline

**ID: 4**
**Guideline Type:** Supportive
**Logical Operation:** AND
**Condition 1:** Stakeholder Heterogeneity Of ?X IS High
**Condition 2:** Degree Of The Importance Of Usability Of ?X IS High
**Condition 3:** Importance OF The Eliciting Implicit Knowledge Of ?X IS High
**Condition 4:** Customer Availability OF ?X IS High
**Technique 1:** Ethnography
**Action :** RECOMMEND **TO** ?X
……

Dissent Guidelines

**ID: 56**
**Guideline Type:** Not Supportive
**Logical Operation:** OR
**Condition 1:** Time Constraint OF ?X IS High
**Condition 2:** Customer Availability OF ?X IS Low
**Condition 3:** Cost Constraints OF ?X IS High
**Technique 1:** Ethnography
**Actions:** NOT RECOMMENDED **TO** ?X
……

Notes: ?X indicates a specific software project

- "Cost Reduction" rules. This rule checks for various ways to reduce cost. For instance, using more than one type of formality (e.g., mathematical notation) in a software project will likely waste resources. We therefore defined the following "Cost Reduction Rule": "If the number of formal methods used in the technique combination is ≥2, then present the "Cost alert message".
- "User-Defined Rules". These rules express constraints on technique selection based on the characteristics of the project. The issues related to the existence of certain cultural and technical constraints in the software organization for technique application have been discussed in [18, 38].

It is worth mentioning that guidelines[3] and rules are different as rules are compulsory while guidelines are not.

*3.2.2.2 Techniques assessment information* Advanced RE technique knowledge includes assessment information of each technique based on two types of assessments:

1. Major COREs assessment: Major COREs (COncern of RE process) is a model for the evaluation of an RE process [6]. Each major concern is a specific interest or objective of the RE process which needs to be addressed. The major COREs model can also be used for the evaluation of RE techniques. For example, we can examine how many major COREs a RE technique addresses. The higher the number of major COREs addressed by a technique the more the technique will contribute to a high-quality requirements specification. In this research, all the identified RE techniques were assessed using the major COREs model. This type of knowledge is represented using Major COREs frames. An example of the major COREs Assessment of the technique "Interview" is shown in Fig. 4. Due to page limitations, only partial results of the assessment are presented. The COREs shown in the table are related to the elicitation aspects of an RE technique, therefore, the assessment can be used to evaluate the effectiveness of RE elicitation techniques.
2. Attribute-based assessment: Based on the analysis of 46 RE techniques, we proposed 31 attributes of RE techniques in our previous research [18]. These attributes are derived based on existing research such

[3] *A guideline is a* statement or an instruction about the best techniques to be chosen or best actions to be taken under certain conditions. Compared to rules, guidelines are not compulsory.



www.manaraa.com

**Attributes-Based Assessment for Interview Technique**

**Instance of:** Attribute-based Assessment
**Technique ID:** 2
**Ability to help facilitate communication:** 0.8
**Ability to help understand social issues:** 0.8
**Ability to help get domain knowledge:** 0.6
**Ability to help get implicit knowledge:** 0.2
**Ability to help identify stakeholders:** 1.0
**Ability to help identify non-functional requirements:** 1.0
**Ability to help identify viewpoints:** 0.8
**......**

**(a)**

**Major COREs Assessment for Interview Technique**

**Instance of:** Major COREs Assessment
**Object of assessment:** Technique
**ID:** 2
**Identification of stakeholders and their viewpoints :** 1.0
**User participation:** 0.8
**Identification of the goals, expectation, scope and context of the project or system : 0.8**
**Effective communication:** 0.8
**Elicitation of functional requirements:** 1.0
**Elicitation of non-functional requirements and system constraints:** 0.8
**Considerations of social, organizational, and political issue :** 0.8
**Recording requirements sources and rationale:** 0.8
**Assessment of system feasibility:** 0.6
**......**

**(b)**

Notes:
1. The attributes in each frame are shown in bold. The attribute values are shown in normal font.
2. In the attribute-based assessment frame (a), except for the first two slots, each slot includes an attribute of a technique and the assessment result of the technique with respect to the attribute. The assessment result adopts the range of [0, 0.2, 0.4, 0.6, 0.8, 1.0]. The numerical value of 0 indicates that the technique has no ability to support the attribute, 1.0 indicates very strong ability.
3. In the Major COREs Assessment frame, (b) except for the first three slots, each slot includes a major CORE and the assessment result of the techniques with respect to the major CORE. The assessment result adopts the range of [0, 0.2, 0.4, 0.6, 0.8, 1.0]. The numerical value 0 indicates that the assessed technique does not address the concern at all; while 1.0 indicates the assessed technique fully addresses the concern.

**Fig. 4** Attributes-based assessment and major COREs assessment for the interview technique

as in [1, 12, 24, 25] as well as our own research in RE techniques. These attributes and their categories are shown in Table 2. The first column lists the categories of the attributes which correspond to the four stages of the RE process. The third column lists the actual attribute name. Each attribute is defined with a list of criteria [18] to ensure its measurability. An ordinal measurement scale is used for all the attributes, i.e. the attribute values are set as none (or "not relevant"), very low, low, medium, high and very high. A total of 46 RE techniques were assessed by three RE researchers using the proposed attributes. Sample data from the overall dataset is shown in columns 4 to 9 of Table 2 which contains the normalized results of the assessment of six RE techniques. For example, the technique Interview, the "Ability to help facilitate communication" is assessed as "very high". The normalized value for "very high" is 1. Thus, the entry for that column is 1.

As can be seen in Table 2, the attributes in the schema provide a means to measure different aspects of an RE technique. At a high level, these aspects can be divided into two parts:

- Attributes that describe the ability of a technique (attributes 1 to 28): The higher the value of an attribute, the more suitable is the technique for addressing the attribute.
- Attributes that describe economic factors (attribute 29 to 31): The higher the value of these attributes, the higher the cost of using this technique.

This classification is essential for the evaluation of RE techniques. Details of the usage of the schema will be provided in Sect. 5.

More information on the derivation of the technique attributes and the evaluation of techniques can be found in [18]. The technique assessment information is stored in an Attributes-Based Assessment frame which helps determine the overall ability and cost of each technique. An example of an Attributes-Based Assessment frame is presented in Fig. 4.

*3.2.2.3 Relationships between RE techniques* Three different kinds of relationships between RE techniques were identified in our research. These relationships represent another type of knowledge that can also be used to support techniques selection. A discussion of these relationships is given in the following:

- Functionally comparable relationship and functionally complementary relationship: In order to analyze the techniques in detail, a data set of attributes of RE techniques was derived from surveys and experts, and then analyzed using a clustering method. This analysis has shown that some RE techniques are functionally comparable and some are functionally complementary to each other. The formal definitions of these two concepts are given in Table 3. Knowing about these relationships is very helpful during the selection of RE techniques. For example, if two techniques are functionally comparable and both are known by the project team, the less expensive technique will be a good choice for a project with tight budget and time constraints. The concept of functionally complementary



المنارة للاستشارات
www.manaraa.com

**Table 2** A proposed classification schema for RE techniques and their assessment

| Categories | No. | Attributes of the techniques | Interview | Exploratory prototypes (throw-away prototype) | JAD | Functional decomposition | State charts (also known as state diagrams) | AHP |
|---|---|---|---|---|---|---|---|---|
| Elicitation | 1 | Ability to help facilitate communication | 1 | 0.8 | 1 | 0 | 0 | 0.6 |
| | 2 | Ability to help understand social issues | 0.6 | 0.2 | 1 | 0 | 0 | 0 |
| | 3 | Ability to help get domain knowledge | 0.6 | 0.4 | 0.6 | 0 | 0 | 0 |
| | 4 | Ability to help get implicit knowledge | 0.2 | 0.2 | 0.2 | 0 | 0 | 0 |
| | 5 | Ability to help identify stakeholders | 1 | 0 | 1 | 0 | 0 | 0 |
| | 6 | Ability to help identify non-functional requirements | 1 | 0 | 0.8 | 0 | 0 | 0 |
| | 7 | Ability to help identify viewpoints | 0.8 | 0 | 1 | 0 | 0 | 0 |
| Analysis and negotiation | 8 | Ability to help model and understand requirements (both general and domain specific requirements) | 0 | 1 | 0 | 0.8 | 1 | 0 |
| | 9 | Ability to help analyze and model requirements with the understandable notations | 0 | 0 | 0 | 1 | 0.8 | 0 |
| | 10 | Ability to help analyze non-functional requirements | 0 | 0 | 0 | 0.2 | 0 | 0 |
| | 11 | Ability to facilitate negotiation with customers | 0 | 0.8 | 0 | 0.4 | 0.4 | 0.6 |
| | 12 | Ability to help prioritize requirements | 0 | 0 | 0 | 0.2 | 0 | 1 |
| | 13 | Ability to help identify accessibility of the system | 0 | 0.8 | 0 | 0.6 | 0.6 | 0 |
| | 14 | Ability to help model interface requirements | 0 | 1 | 0 | 0.2 | 0.6 | 0 |
| | 15 | Ability to help identify reusable requirements and support requirements reuse | 0 | 0 | 0 | 0 | 0 | 0 |
| Documentation and notation | 16 | Ability to represent requirements (Expressibility) | 0 | 1 | 0 | 0.8 | 1 | 0 |
| | 17 | Ability to help verify requirements automatically by using the notation | 0 | 0 | 0 | 0.2 | 0.8 | 0 |
| | 18 | Completeness of the semantics of the notation | 0 | 0 | 0 | 0.6 | 0.6 | 0 |
| | 19 | Ability to help write unambiguous and precise requirements by using the notation | 0 | 0 | 0 | 0.6 | 0.8 | 0 |
| | 20 | Ability to help write complete requirements | 0 | 0 | 0 | 0.6 | 0.6 | 0 |
| | 21 | Ability to help management of requirements | 0 | 0 | 0 | 0.6 | 0 | 0 |
| | 22 | Modularity | 0 | 0 | 0 | 0.6 | 0 | 0 |
| | 23 | Implementability (Executability) | 0 | 0 | 0 | 0 | 0 | 0 |
| Verification and validation | 24 | Ability to help identify ambiguous requirements | 0 | 0 | 0 | 0 | 0.6 | 0 |
| | 25 | Ability to help identify interactions (inconsistency, conflict) | 0 | 0 | 0 | 0 | 0.2 | 0 |
| | 26 | Ability to help identify incomplete requirements | 0.2 | 0 | 0 | 0 | 0 | 0 |
| Other aspects | 27 | Ability to support COTS-based RE process | 0 | 0 | 0 | 0 | 0 | 1 |
| | 28 | Maturity of supporting tool | 0 | 0.8 | 0.4 | 0.8 | 0.6 | 1 |
| | 29 | Learning curve (Introduction cost) | 0.2 | 0.2 | 0.6 | 0.4 | 0.6 | 0.6 |
| | 30 | Application cost | 0.4 | 1 | 0.6 | 0.2 | 0.6 | 0.4 |
| | 31 | Complexity of techniques | 0.2 | 0.2 | 0.2 | 0.2 | 0.4 | 0.4 |



www.manaraa.com

**Table 3** Formal definition of the technique relationships

| | Functionally comparable techniques | Functionally complementary techniques | Mutually exclusive techniques |
|---|---|---|---|
| Definition | Two techniques $t$ and $t'$ are functionally comparable if and only if $t$ and $t'$ are in the same cluster and the differences of their attributes' values are within a specified range, i.e. $\lvert t(j) - t'(j)\rvert \le \varepsilon$ for all $j = 1, ..., 31$, where $\varepsilon$ is a project dependent value, normally $\varepsilon \le 0.4$. The set of functionally comparable techniques of $t$ can be written as a function $F(t)$,<br>F: $\mathbf{T} \to$ p ($\mathbf{T}$),<br>$F(t) = \{t' \mid t' \in \mathbf{T} \wedge \lvert t(j) - t'(j)\rvert \le \varepsilon\}$ for $j = 1, ..., 31$. | Two techniques $t$ and $t'$are functionally complementary if and only if $t$ and $t'$ are not in the same cluster and $\left\lvert\sum_{k=1}^{31}(t(k) - t'(k))\right\rvert \le \varepsilon$. Our research determined that $\varepsilon \le 0.8$ is a suitable value for $\varepsilon$. The set of functionally complementary techniques of technique $t$ can be written as a function $C(t)$, where,<br>$C$: $\mathbf{T} \to$ p ($\mathbf{T}$),<br>$C(t) = \{t' \mid t' \in \mathbf{T} \wedge \left\lvert\sum_{k=1}^{31}(t(k) - t't(k))\right\rvert \le \varepsilon\}$. | Two techniques $t$ and $t'$ are mutually exclusive if and only if the usage of technique $t$ violates the basic principles of technique $t'$.<br>The set of exclusive techniques of technique $t$ can be denoted as $E(t)$, where<br>$E$: $\mathbf{T} \to p$ ($\mathbf{T}$),<br>$E(t) = \{t' \mid t' \in \mathbf{T} \wedge t'$ and $t$ are mutually exclusive$\}$ |
| Semantics | The semantics of the functionally comparable techniques states that the techniques are functionally similar with regards to the attributes defined in our research. If two RE techniques are functionally comparable, the major functions of these two techniques are very similar. The condition "in the same cluster" ensures that the two techniques are functionally similar. | The semantics of the functionally complementary techniques states that the techniques are functionally not the same, but complementary to each other. If two RE techniques are complementary, the advantage of one technique is the weakness of the other. The condition "not in the same cluster" ensures the two techniques are not functionally similar, but complementary. | The semantics of the mutually exclusive techniques states that the constraints (policies) of the two techniques are in conflict. The usage of such techniques in the same RE process model would cause process inconsistencies. |
| Example | For example, $Z$ and SDL are *functionally comparable techniques* as both techniques can support formal requirements documentation and verification. This is also supported mathematically, as MAX$\lvert Z(j) -$ SDL$(j)\rvert = 0.4$ and $0.4 \le \varepsilon = 0.4$ for all $j = 1, ..., 31$ | For example, Ethnography (Observation) and Interview are complementary techniques, because Observation is good at eliciting implicit knowledge and the overall behavior of the system, but not good at identifying stakeholders, future requirements and the technical data of the system. Interviewing, on the other hand, is good at doing these. Therefore, these two techniques are functionally complementary | For example, the technique "XP requirements verification" and technique "$Z$ requirements verification" are mutually exclusive techniques since requirements verification using $Z$ violates the basic principles of XP |
| Property | Functionally comparable techniques satisfy the commutative law, but do not satisfy the transitive law. For example, if $t_i$ is a functionally comparable technique to $t_k$, then $t_k$ is also a functionally comparable technique to $t_i$; however, if $t_i$ is a functionally comparable technique to $t_j$, and if $t_j$ is a functionally comparable technique to $t_k$, then it is not necessarily true that $t_i$ is a functionally comparable technique to $t_k$. | Complementary techniques satisfy the commutative law, but not the transitive law. For example, if $t_i$ is a functionally complementary technique to $t_k$, then $t_k$ is also a functionally complementary technique to $t_i$; however, if $t_i$ is a functionally complementary technique to $t_j$, and if $t_j$ is a functionally complementary technique to $t_k$, then it is not necessarily true that $t_i$ is a functionally complementary technique to $t_k$ | Mutually exclusive techniques satisfy the commutative law, but not necessarily the transitive law. For example, if $t_i$ is a mutually exclusive technique to $t_k$, then $t_k$ is also a mutually exclusive technique to $t_i$; however, if $t_i$ is a mutually exclusive technique to $t_j$, and if $t_j$ is a mutually exclusive technique to $t_k$, then it is not necessarily true that $t_i$ is also a mutually exclusive technique to $t_k$ |

$\mathbf{T}$ is the set of all existing RE techniques $t$; $p$ ($\mathbf{T}$) denotes the power set of $\mathbf{T}$; $t$, $t'$ and $t_i$ denote individual RE techniques, currently, i = 1,..., 46; $A_1, A_2,..., A_n$ are the attributes of $t$, $A_i \in \mathbf{A}$, $i$ = 1, ..., 31; $\mathbf{A}$ represents the set of all currently identified attributes of RE techniques. The values of $\mathbf{A}$ are defined as real numbers in the range of [0, 0.2, 0.4, 0.6, 0.8, 1] or $\mathbf{A}$ = [0, 0.2,0.4, 0.6, 0.8, 1]; $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ are the values of the attributes for techniques $t_i$, i.e. $t_i = \langle a_{i,1}, a_{i,2}, \ldots, a_{i,m}\rangle$; $t_i(j) = a_{i,j}$; $i = 1, \ldots, 46$, $j = 1, \ldots, 31$. In this case $\mathbf{T}$ can be written as $\mathbf{T} = A_1 \times A_2 \times \cdots \times A_m$



www.manaraa.com

techniques, on the other hand, helps the developer find technique combinations that offer the maximum benefit for the elicitation, analysis, documentation, verification and validation of requirements.

- Mutually exclusive relationship: This type of relationship between RE techniques is used to examine the consistency of the recommended RE techniques. Techniques $t$ and $t'$ are mutually exclusive if technique $t$ violates some basic principles of technique $t'$ (Refer to Table 3 for the definition and an example of mutually exclusive techniques). This knowledge is captured in the Rule frame.

### 3.2.3 Project cases

In our previous research we documented several project cases to determine what RE techniques are the most suitable for a certain type of project [18]. These cases were developed based on a structured survey completed by practitioners in industry as well as experts in academia. Some cases are derived directly from past projects. Some cases are derived from the predictive judgment aggregated from different experts. For example, if expert $A$ recommends techniques $T_i = \{t_1, t_2\}$ for project $Pr_i$ and expert $B$ recommends $T_j = \{t_1, t_3\}$ for the same project then the final recommendation would be either $T_i \cup T_j$ if $t_2$ and $t_3$ are not mutually exclusive, or $T_i \cap T_i$ if $t_2$ and $t_3$ are mutually exclusive. According to the argument made by Helmer and Rescher [39], in science, especially in imprecise science where no accepted measurements are available, the incorporation of expert opinions or judgment into the research subject area can be considered valid if a structured approach is adopted. Based on this assertion, we argue that the process used in this research for the derivation of the recommendation rules from experts would qualify as an acceptable method for decision support.

The project cases are stored in a project case library by using a frame representation. An example of project cases currently contained in our project case library is given in Fig. 5. These project cases can be retrieved and the RE techniques used in these projects are considered as the initial recommendation for a new project. To find appropriate cases, a case-based reasoning (CBR) mechanism is necessary as the knowledge could not be logically organized due to its complex and diverse nature. However, the recommended techniques may not be totally suitable for a new project. In this case, further refinement and adaptation of the initial recommendation are needed.

## 4 Reasoning mechanism

The knowledge structure of RE techniques is complex and multi-dimensional, which makes efficient knowledge

**Fig. 5** A frame-based representations for a software project case

**FRAME:** Web-Based Realtor
**Instance Of :** Project Case Frame
**Project Name:** Intelligent Web-based Realtor System
**Case ID:** 19

**Project Situation:**
**Major Attributes:**
**Project size:** Medium
**Project complexity:** Medium
**Requirements volatility:** Medium
**Organization-customer relationship:** DGP (stands for the software company developing a generic product for a specific market)
**Project category:** Semi-detached
**Degree of safety criticality:** High
**Quality standard**: High
**Product type:** New
**Time constraints:** Medium
**Cost constraints:** Medium
**Team size:** 12
**Acquaintance of the domain:** Medium
**Degree of RE knowledge:** Low
**Degree of knowledge about requirements:** Medium

**Other Attributes**
**Availability of skilled facilitator:** High
**Stakeholder heterogeneity:** Medium
**Degree of innovative of the Project:** Medium
**Customer availability:** High
**Degree of the importance of reusability:** Medium
**Degree of the importance of eliciting implicit knowledge:** Low
**Degree of outsourcing:** Very Low

**Recommended RE Techniques:**

| | | | |
|---|---|---|---|
| **Elicitation:** | 1. Focus Group | 2. QFD | |
| **Analysis & Negotiation:** | 1. Scenario-Based Approach | 2. Goal-Oriented Approach | 3. Structured Analysis |
| **Documentation:** | 1. UML | 2. Structured Natural Language Specification | |
| **Verification & Validation:** | 1. Formal Requirements Inspection | | |
| **Tool Support:** | DOORS | | |

**……**

**ENDFRAME**



المنارة للاستشارات
www.manaraa.com

management challenging. We therefore use three types of reasoning mechanisms in the KASRET approach: Case-Based Reasoning, Frame-Based Reasoning, and Relational Reasoning. These three mechanisms are discussed in the following subsections.

### 4.1 Case-based reasoning

Our Knowledge-based Approach for the Selection of Requirements Engineering Techniques (KASRET) uses case-based reasoning (CBR) to allow past experiences to be used when new projects are being developed. CBR is carried out after the attribute values of the new project are determined. The purpose of CBR is to look for a case in RETKL that has similar project attribute values as the new project. The similarity between the project attributes (see Fig. 5 for an example of the project attributes) of the existing cases and the attributes of the new project is calculated using the modified weighted Euclidean distance [40]. That case that has the highest similarity to the given project, denoted as $\gamma_i$, is taken as the result of CBR. The values of the decision attributes are the specific RE techniques that were used in the most similar project case $\gamma_i$ and will be recommended to the users as suitable techniques for the new project. Let's assume that:

- $R$ is the existing set of project cases contained in RETKL. The case set can be represented as $(X_k, D_m)$, with $X_k$ representing the condition attributes (i.e., techniques attributes), and $D_m$ representing the decision attributes (i.e., techniques used in the project). All existing cases have $k$ project attributes and $m$ decision attributes.
- $\gamma_i = (X_i, D_i)$ represents one of the existing cases in $R$; $i = 1, ..., n$, with $n$ being the total number of cases.
- $X_i = (x_{i,1}, x_{i,2}, ..., x_{i,k})$ denotes the values of the project attributes of the $i$th case in RETKL; $x_{i,j}$ represents the value of the $j$th project attribute for the $i$th case; $j = 1, ..., k$.
- $D_i = (d_{i,1}, d_{i,2}, ..., d_{i,m})$ denotes the values of the decision attributes of the $i$th case in RETKL. $d_{i,p}$ represents the value of the $p$th decision attribute for the $i$th case; $p = 1, ..., m$. A value of a decision attribute $d_{i,j}$ is a specific RE technique used for the project with the values of project attributes$(x_{i,1}, x_{i,2}, ..., x_{i,k})$ as the conditions.
- $Y = (y_1, y_2, ..., y_k)$ denotes the values of attributes of a new project; $y_j$ represents the value of the $j$th attribute for the given project; $j = 1, ..., k$.
- $W_j$ denotes the weight for each project attribute; $j = 1, ..., k$; $W_j = [1, ..., 5]$.

Then, the similarity function can be defined as:

$$\text{Similarity}(X_i, Y) = \sqrt{\frac{1}{\sum_{j=1}^{k} W_j * (F(x_{i,j}, y_j))^2}} \quad i = 1, \ldots, n;\ j = 1, \ldots, k \tag{1}$$

where

$$F(x_{i,j}, y_j) = \begin{cases} a_j - b_j & \text{If the values of } x_{i,j}, y_j \text{ are in ordinal scale and } x_{i,j}, y_j \text{ are mapped (normalized) to the numerical values } a_j, b_j \text{ within } [0,1] \text{ for all } j. \\ 0 & \text{If } x_{i,j}, y_j \text{ are in nominal scale and } x_{i,j} = y_j \\ 0.5 & \text{If } x_{i,j}, y_j \text{ are in nominal scale and } x_{i,j} \neq y_j \end{cases}$$

Based on the similarity calculation, the initial set of recommended techniques for the given project are:

$$T_{\text{IR}} = \{d_{i,1}, d_{i,2}, \ldots, d_{i,n} | \gamma_i \in R \wedge \text{Max}(\text{similarity}(X_i, Y)), \quad i = 1, \ldots, n, \} \tag{2}$$

$T_{\text{IR}}$ represents the initial set of recommended RE techniques, Max(similarity($X_i$, $Y$)) indicates the maximum value of the similarities calculated between all existing cases according to (1).

In this similarity calculation, we assume that no two existing cases have exactly the same value of overall similarity with the new project, i.e. no $X_j$ exists in RETKL, such that Similarity($X_i$, $Y$) = Similarity($X_j$, $Y$). If $X_j$ exists, then additional attributes must be considered that help differentiate between the two cases or the developer has to choose one of them based on personal preference.

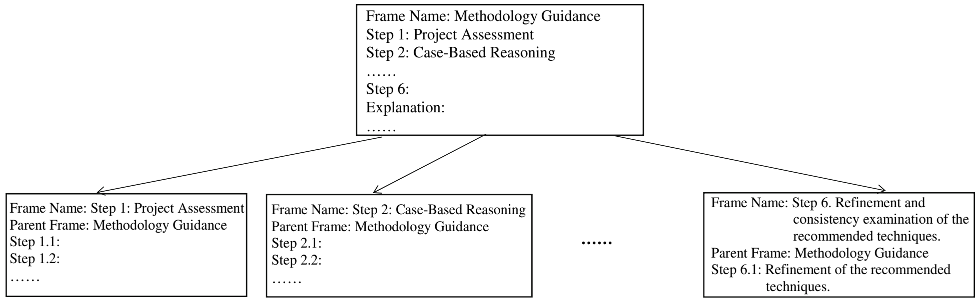
### 4.2 Frame-based reasoning

Frames are a very effective method for storing complex knowledge or models of knowledge. They allow generic features and the relationships between these features to be represented and processed [41]. Frames also allow several instances of features to be stored within the same knowledge base. As can be seen from the structure of the guidelines frame for the KASRET approach illustrated in Fig. 6, the inheritance structure allows process knowledge to be stored in different frames more effectively. It also allows the knowledge both from lower levels (child level) and higher levels (parent level) to be retrieved efficiently. Therefore, the knowledge stored in this type of frames can point out what step requirements engineers are currently working on, what the previous step was, and what the next



المنارة للاستشارات
www.manaraa.com

**Fig. 6** A structure of guidelines frame for the KASRET approach




step will be. These are the major reasons for using Frame-Based Reasoning (FBR) in our research.

The software literature defines numerous diverse attributes for software projects as well as for RE techniques. However, if the number of attributes in a frame is too big, knowledge retrieval becomes inefficient. Additionally, we found that it is necessary to define a cohesive set of typical attributes for software projects as well as for RE techniques. Therefore, during the design of RETKL, 21 project attributes and 31 technique attributes were defined [18], which allow the efficient storage, management and retrieval of knowledge.

In this research, FBR was implemented using a mechanism that is similar to the production rule inference method [42] with added operations on one or more slots that indicate the status of the current inference. FBR is initiated if the conditions in one or more slots are satisfied and/or a particular status in the inference process is reached. In summary, the support for RE technique selection provided by FBR can be categorized into the following types:

1. Support for retrieving general information of RE techniques. Frame-based reasoning can help RE techniques selection by retrieving the techniques knowledge from RETKL. The information can be very general, such as the activities of the technique, its strengths, weaknesses, guidelines for use, etc. For example, the aim of the query shown in Fig. 7 is to look for the identification numbers, names and the strengths of techniques that are most suitable for a software project with the following characteristics:

   - Project Complexity is equal to or below Medium
   - Requirements Volatility is equal to or greater than High
   - Time Constraint is equal to or greater than High
   - Team Size is equal to or less than Small

   The result of the frame-based reasoning for the example in Fig. 7 can be seen in Table 4.

2. Support for retrieving specific guidelines for selection of RE techniques: As mentioned in Sect. 3.2.2.1, guidelines knowledge for RE techniques selection was developed during this research. The guideline knowledge is represented in the Guideline frames. Built on the Guideline frames, a frame-based reasoning mechanism is provided. As an example of the procedures of RE techniques selection, the algorithm of the Frame-based reasoning for generating guidelines for selection of RE techniques is described in pseudo code in Fig. 8. The reasoning can support the following types of information retrieval:

   - Information regarding the suitability of a specific technique for a particular project (i.e., suitable or unsuitable). This type of information is related to two kinds of guidelines, “Assent Guidelines” and “Dissent Guidelines”. This helps requirements engineers to construct a techniques recommendation space which is required for the overall techniques selection process in KASRET. A techniques recommendation space, represented as $T_{RS}$, is the set of those candidate RE techniques that are considered suitable for the given project. However, this techniques recommendation space is still subject to further refinement in order to get a compact, consistent set of RE techniques that have lower cost and higher ability in the context of the given software project.
   - Information regarding mutually exclusive RE techniques (see Table 3). Information regarding mutually exclusive RE techniques can be used to eliminate inconsistencies in the selection of RE techniques.

3. Frame-based reasoning helps produce methodological guidelines for carrying out the tasks of the proposed

**Fig. 7** Query for the knowledge of RE techniques

```
SELECT ?Tech_ID, ?Tech_Name, ?Strength
FROM  Most_Suitable_Project_Type
WHERE  ProjectComplexity <=Medium AND RequirementsVolatility>=High AND TimeConstraint>=High AND TeamSize<=Small.
```


www.manaraa.com

**Table 4** Result of the frame-based reasoning based on the query illustrated in Fig. 7

| ID | Name of retrieved techniques | Advantages of the technique |
|---|---|---|
| 42 | Extreme Programming | Low management overhead, better communication between developers, more satisfied customers and shorter release cycles |
| 11 | Evolutionary Prototype | Good for identifying implicit knowledge, anomalous states, early feedback, etc |
| 10 | Exploratory Prototype (throw-away prototype) | Good for elicitation of user requirements that are hard to articulate. Good for identifying implicit knowledge, anomalous states, etc |
| 1 | Brainstorming and Idea Reduction | Simple, easy, good for eliciting functional requirements |
| 14 | Scenario-Based Approach | Easy integration with OO methods such as OOSE or UML. Positive feedback in trials. Used in later stages of elicitation when initial requirements are already available. Aimed at the elicitation of detailed functionality of the system |
| 28 | Object-Oriented Analysis (OOA) | Maintainability through simplified mapping to the real world. Easier verification by the user. Reusability of analysis artifacts which saves time and costs. Productivity gains through direct mapping of artefacts to features of OO Programming Languages |

**Fig. 8** An abstract description of the Guideline frame-based reasoning process with pseudo code

```
Load the definition of attributes of the given project: Aj = yj , j=1,…, k
   FOR each guideline ri(C, t) in the Guideline Frames
      IF C of the guideline ri match exactly the definitions of project attributes Aj = yj;
      THEN
         IF the Guideline Type of ri is a Assent Guideline
         THEN
              Begin
                 Present the guideline ri by recommending the technique t ;
              End
         ELSE
              IF the Guideline Type of ri is a Dissent Guideline AND t ∈ TRS
                 Present the guideline rj by NOT recommending the usage of the technique t ;
                 Remove the technique t from the candidate set of techniques;
              ENDIF
         ENDIF
      ENDIF
   ENDFOR
```

Notes: 1. $r_i(C, t)$ represents a guideline stored in the Guideline Frame, while C represents a set of conditions, $t$ is the technique that is supportive or unsupportive under the condition C.
2. $A_j = y_j$ represents the definitions of all attributes of a given project. $A_j$ is the attribute, $y_j$ is the value of $A_j$

KASRET approach. The guidelines of the approach are represented as production rules and are stored in guidelines frames. This treatment allows the reasoning to be carried out during the run-time of the KASRET tool. The general idea of the frame is illustrated in Fig. 9. The algorithm of a pseudo-code example for the reasoning process is given in Fig. 10. In this example, "the next step" shown in Fig. 9 of using the approach is going to proceed to "step 2" where the Case-Based Reasoning operation (implemented by the CBR_Routine) is carried out.

### 4.3 Relational reasoning

Relational reasoning is a mechanism that helps find techniques that are functionally comparable, functionally complementary, or functionally exclusive. Relational reasoning builds on the results of techniques clustering. A fuzzy clustering method is used as the data set derived is fuzzy in nature. The clustering can either use all the attributes in the schema (see Table 2) by default or a subset of the most relevant ones. For example, requirements engineers might be very interested in the following set of attributes: {Ability to help facilitate communication, Ability to help identify various viewpoints, Ability to help facilitate negotiation with customers, Ability to help model and understand requirements}. Thus, the clustering will be carried out based on these attributes only, instead of all 31 attributes. Requirements engineers can also assign appropriate weights to each attribute depending on their importance for the new project. The result of the clustering are sets of techniques that are functionally comparable and functionally complementary to a given technique. The inclusion of these sets of techniques into the techniques recommendation space provides more choices for requirements engineers to select the most suitable techniques for a



www.manaraa.com

**Fig. 9** Guidelines frame for the KASRET approach

```
FRAME: Approach_Guidelines
OVERALL STATUS: Not Finished
STEP 1:                              /*Scoring the attributes of the given project*/
        OPERATION: Scoring_Routine
        STATUS: Finished
STEP 2:                              /* Derivation of the initial recommendations of the RE techniques by using CBR*/
        OPERATION: CBR_Routine
        STATUS:  Not Finished
STEP 3: Clustering                   /* Analysis of the RE techniques by using the clustering method*/
        STATUS: No
        SUBSTEP_1 :                     /* Selection of a set of technique attributes */
               OPERATION: Selection_Routine
               STATUS: Not Finished
        SUBSTEP_2 :                     /* Clustering of all RE techniques in RETKL */
               OPERATION: Clustering_Routine
               STATUS: Not Finished

…..
ENDFRAME

Notes:
1. OPERATION indicates what operation shall be done in this step. The value given to the OPERATION is a name of the predefined routine (e.g.,
   Scoring_Routine) that will be called during the reasoning process.
2. STATUS indicates whether a step of the approach is finished or not. It has two possible values: "Done", which means this step is finished, or
   "No", which indicates this step is not done completed.
```

**Fig. 10** Example of the reasoning process

```
IF Approach_Guidelines. Overall Status=No AND Approach_Guidelines. Step_1.Status=No
THEN @ Scoring_Routine                  /* call Scoring_Routine here into Step 1 */
ELSE
    IF Approach_Guidelines. Overall Stautus=No AND Approach_Guidelines. Step_1.Status=DONE AND Step_2. Status=NO.
    THEN @ CBR_Routine                  /* call CBR_Routine here into Step 2 */
    ELSE
      …….

Notes: "@P" indicates the execution of the routine "P"
```

given project. An abstract procedure of this type of reasoning is shown in Fig. 11.

## 4.4 Objective function

The objective for selecting techniques for a given project is to find that combination of techniques that maximizes the overall ability of the techniques and minimizes the overall application cost. The objective function is a tool that helps achieve this aim by focusing on the quality of requirements specifications, the time to market, as well as the complexity and cost of the RE techniques related to their application and necessary training. The techniques selected from the $T_{RS}$ must meet the criteria set by the objective function, which is formally defined as follows:

$$F_C : \mathbf{p}(\mathbf{T}) \rightarrow \text{Real}$$

$$F_C(T_i) = \sum_{t \in T_i} \text{Ability}_t \tag{3}$$

$$\text{Ability}_t = \sum_{i=1}^{28} t(i) - 2^*(B^* t(29) + t(30) + t(31)) \tag{4}$$

$$T_C = \text{Max}\, F_C(T_i) \text{ for all } T_i \in T_{RS}, \tag{5}$$

where

- $F_C$ represents the objective function.
- $\mathbf{T}$ is the set of all existing RE techniques $t$; $\mathbf{p}(\mathbf{T})$ denotes the power set of $\mathbf{T}$.
- $T_i$ is one of the techniques combinations in $T_{RS}$ and contains a set of techniques. Each $T_i$ includes requirements elicitation techniques ($T_e$), requirements analysis and negotiation techniques ($T_a$), requirements documentation techniques ($T_d$), and requirements verification and validation techniques ($T_v$). i.e. $T_i = T_e \cup T_a \cup T_d \cup T_v$;
- $F_C(T_i)$ represents the value of the objective function for the recommended set of RE techniques $T_i$;
- $\text{Ability}_t$ indicates the normalized numerical value of the overall ability of technique $t$;
- $T_C$ denotes the recommended solution for the given project with the maximum value of $F_C(T_i)$ among all $T_i$s;

$t(1)$ to $t(31)$ represent the normalized numerical values of the RE technique attributes 1 through 31 (see Table 2); $B$ is a coefficient that represents the requirements engineers' knowledge about a technique:


www.manaraa.com

**Fig. 11** Abstract description of the relational reasoning process

```
/*The Relational Reasoning process for identifying techniques that are functionally comparable and functionally complementary to technique t */

Select the most important attributes of RE techniques based on the characteristics of the given project: A^P ={a_1 , …, a_m}
Number-of-clusters=9;   /* 9 is an experience value */
Cluster the RE techniques based on the attributes in A^P using the Fuzzy Clustering Algorithm
FOR each technique t in Cluster C_i , i=1 to 8
    FOR each technique t' in C_k, k=i+1,…,9
        IF |t(j) − t'(j)| ≤ 0.4   for all j=1, …, m,          // 0.4 is an experience value
            THEN
                BEGIN
                    /* t' is a functionally comparable technique to t; */
                    FT(t)=FT(t) ∪ { t' }
                END
        ENDIF
        IF |Σ_{j=1}^{m} (t(j) − t'(j))| ≤ 0.8   for all j=1, …, m,        // 0.8 is an experience value
            THEN
                BEGIN
                  /* t' is a functionally complementary technique to t; */
                    CT(t)=CT(t) ∪ { t' }
                END
        ENDIF
    ENDFOR
ENDFOR


Notes: 1. C_i and C_k represent the cluster i and k respectively.
       2. t(j) indicates the value of the jth attribute of the technique t.
       3. FT(t) represents a set of functionally comparable techniques to technique t. CT(t) represents a set of techniques that are functionally
          complementary to technique t.
       4. A^P indicates the given software project with m attributes a_1 , …, a_m
```

- $B = 1$, if the requirements engineers do not know a technique at all, i.e. the introduction cost attribute of the technique needs to be taken into consideration.
- $B = 0$, if the requirements engineers have extensive knowledge about a technique, this means that the introduction cost attribute does not need to be considered because the introduction cost of the technique is not an issue.
- Intermediate values are possible to allow for partial knowledge of a technique.

The numerical factor 2 in formula (4) is an experience factor derived from our case studies and trials, and ensures that the economic factors are adequately weighted [43].

It is worth mentioning that the assignment of the techniques in $T_{RS}$ (a techniques recommendation space) to $T_i$ (a technique combination) can be done manually by requirements engineers or automatically by computers based on established rules. The explanation of these rules is beyond the scope of this paper and subject to further research.

As has been mentioned earlier, the purpose of the objective function is to look for that RE technique combination $T_i$ that has the highest overall ability among all possible combinations of the RE techniques in $T_{RS}$ and the lowest overall cost and complexity.

## 5 The overall process of the KASRET approach

The KASRET approach facilitates the selection of RE techniques. It defines a systematic process in which the major components of RETKL, the reasoning mechanisms, and the decision making models work together. The overall KASRET approach (see Fig. 12) consists of six steps which are described in the following:

Step 1. Project assessment.

The requirements engineers assess the given project by scoring the attributes of the project. The score can be based on initial estimates according to expert experience if detailed information is not yet available. This step provides information for the retrieval of similar cases stored in the RETKL which will be done in Step 2.

Step 2. Case-based reasoning.

CBR is conducted in RETKL in order to derive the initial set of recommended RE techniques $T_{IR}$ that were used in previous project cases that are similar to the current project (see Sect. 3.2.3). However, the retrieved solution needs to be revised to reflect any differences between the new project and the retrieved case. This is particularly important if RETKL does not contain many project cases. The retrieved RE techniques will be further analyzed in Step 3.

Step 3. Analysis of RE techniques using clustering.

The task of this step is to analyze all the techniques in RETKL using the Fuzzy Clustering algorithm [43]. The objective of the clustering is to analyze all the techniques in RETKL, and to identify functional complementary and functional comparable RE technique of the techniques in $T_{IR}$ derived in Step 2. This step, in turn, includes the following sub-steps:

(a) The requirements engineer selects a set of technique attributes which are considered important for the



المنارة للاستشارات
www.manaraa.com

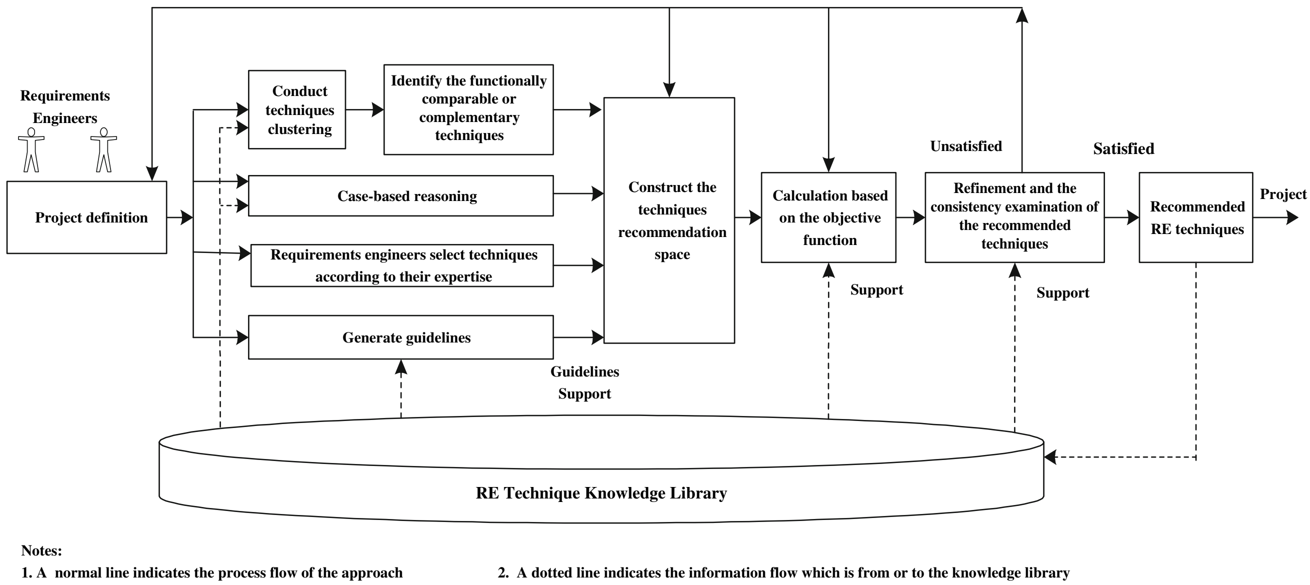



**Fig. 12** Overall process of the KASRET approach

given project. The default is that all attributes are selected for the clustering.

(b) The requirements engineer assigns weights to each selected attribute. The weight of each attribute is determined by the requirements engineers based on their experiences and judgment of the characteristics of the given software project. The weight will be 5 (the highest value) if the attribute is considered essential for the RE process. For example, the "Ability to help identify stakeholders" is considered as a very important issue for the project and is therefore given a weight of 5.

(c) The number of clusters has to be determined. Our experience shows that 8 to 10 clusters provide the best results [43].

(d) The RE techniques stored in RETKL are assigned to the most appropriate clusters by using a Fuzzy Clustering algorithm.

The information regarding the classification and the relationship between the RE techniques identified in this step will be used in Step 4 to construct the techniques recommendation space $T_{RS}$.

Step 4. Construction of the techniques recommendation space $T_{RS}$

The techniques recommendation space $T_{RS}$ is constructed by including techniques suitable for the project. This is done by requirements engineers supported by the RETKL. It includes the following sub-steps:

(a) Analyzing the techniques in $T_{IR}$ (the initial set of recommended RE techniques), to ensure that all the techniques are compatible with the new project, i.e. no technique conflicts with the characteristics of the new project based on the guidelines and rules used in the frame-based reasoning. For example, the "Dissent Guidelines" generated from the frame-based reasoning mechanism are used to examine the suitability of each technique in $T_{IR}$. The result of this reasoning process is a list of techniques, denoted as $T_{US}(T_{US} \subset T_{IR})$ that are NOT suitable for the given project. These techniques will be removed from $T_{IR}$, ie. $T_{IR} = T_{IR} - T_{US}$, $T_{US}$ could be an empty set.

(b) Selecting a set of RE techniques based on requirements engineers' past experience. This set of techniques is denoted as $T_{ER}$ (ER stands for Engineers' Recommended techniques). This allows the requirements engineers to use his/her expertise in the decision making process. $T_{ER}$ could be an empty set.

(c) Identifying all techniques that are functionally comparable and functionally complementary to all the techniques in $(T_{IR} \cup T_{ER})$ by using the results of the clustering in step 3.

(d) Combining the techniques identified in steps (a), (b) and (c) to construct the technique recommendation space $T_{RS}$. The requirements engineers are involved in this step.

Mathematically, the technique recommendation space can be represented as:

$$T_{RS} = T_{IR} \cup T_{ER} \cup \{t' | t' \in C(t) \wedge t \in (T_{IR} \cup T_{ER})\} \cup \{t'' | t'' \in F(t) \wedge t \in (T_{IR} \cup T_{ER})\}.$$



المنارة للاستشارات
www.manaraa.com

where $C(t)$ and $F(t)$ represent functionally complementary techniques and functionally comparable techniques to technique $t$ as defined in Table 3.

The techniques in the $T_{RS}$ derived throughout the last four steps are candidate techniques for the software project based on attributed defined in Step 1. In the next step, developers will look for the best combination of techniques contained in $T_{RS}$ that maximizes the overall ability and minimizes the overall application cost of the combination of techniques.

Step 5. Calculation.

In this step, the requirements engineers select various combinations of RE techniques from $T_{RS}$, and derive the best combination based on the calculation of the objective function defined in Sect. 4.4. This step includes the following sub-steps:

(a) Combine techniques within $T_{RS}$ to form a set of $T_i$ which includes those techniques that can be used in the RE process. The requirements engineer constructs the candidate set $T_i$ within $T_{RS}$.
(b) Compute the overall ability of all techniques combinations $T_i$ created in step (a) using the objective function defined in the last section.

The RE technique combinations constructed in step (a) are ordered according to their overall techniques ability calculated in this step. The RE technique combination with the highest overall ability, denoted as $T_C$, will be chosen and recommended for the software project. The RE technique combinations now have to be further analyzed in the next step to ensure the consistency and necessity of the techniques in each combination based on the requirements engineers expertise.

Step 6. Refinement and consistency examination of the recommended techniques.

The recommended techniques in $T_C$ derived in the last step are adjusted according to the outcome of the frame-based reasoning and the experience of the requirements engineers. For example, the rules and guidelines of the techniques can be retrieved through frame-based reasoning to ensure the consistency of the recommended techniques in $T_C$. Moreover, the completeness of the final recommendation will be examined.

## 6 A case study

Conducting a case study is a common approach to examining the merits of a new conceptual framework. However, it is also a very challenging task, particularly in the software engineering domain. Many issues and variables have to be taken into consideration before a final conclusion on a framework can be made. In this research, three case studies from different domains have been conducted: (1) a Port Scheduling System (PSS) system, (2) a Web-based Herb Trade system, (3) an Intelligent Power Optimization System (IPOS). The IPOS case study is summarized in this section.

The Intelligent Power Optimization System (IPOS) is an industrial project in company Y (the name of the company is withheld for reasons of confidentiality), which is a medium-sized software organization. The major objective of IPOS is to provide advanced, real-time supply chain management solutions to optimize power networks within a geographical area of 685,000 km$^2$ with a population of over 2 million people. The allocation of electrical power can be done both automatically and manually. The tariff structure consists of three layers: peak price, partial peak price and normal price which reflects the objective of minimizing power production costs and of meeting stringent emission regulations. Companies in which sudden power outages can lead to significant financial loss and disastrous accidents have to be protected from power outages if at all possible.

The case study was designed by following the steps proposed in [44]:

1. Definition of the hypothesis
2. Selection of the case study project
3. Selection of a suitable method for comparison of the results with other projects and selection of the criteria for the validation
4. Consideration of the effects of confounding factors
5. Planning of the case study
6. Conducting and monitoring of the case study against the plan
7. Analysis of results and generation of report

The case study was carried out in close collaboration between developers at company Y and the authors. The following contains a brief description of steps 6 and 7 of the case study.

### 6.1 Execution of the case study

Step 1. Project assessment

After the initial analysis of the project, the requirements engineers developed the basic project definition partially shown in Fig. 13b.

Step 2. Case-Based Reasoning

Case-based reasoning was conducted with the information stored in RETKL and a project case that is similar to



المنارة للاستشارات
www.manaraa.com

**Case retrieved from RETKL**

**FRAME:** Integrated_Furnaces_Control_System
**Project Name:** Integrated Furnaces Control System
**CaseID: 10**

**New case - project under development**

**FRAME:** Power_Optimization_System
**Project Name:** Intelligent Power Optimization System
**CaseID:**

**Project Situation:**

| Case retrieved from RETKL (a) | $\lvert F(X_{i,j}, Y_j) \rvert$ | $W_i$, Weight of *i*th attribute | New case (b) |
|---|---|---|---|
| **Major attributes:** | | | **Major attributes:** |
| **Project size:** Very Big | ← 0.25 | 5 → | **Project Size:** Big |
| **Project complexity:** High | ← 0.25 | 5 → | **Project complexity:** Very High |
| **Requirements volatility:** Low | ← 0.25 | 5 → | **Requirements volatility:** Very low |
| **Organization-customer relationship:** ITT (stands for the software company getting an invitation to tender ) | ← 0.5 | 2 → | **Organization-customer relationship:** SCR (SCR stands for responding to a Specific Customer Request) |
| **Project category:** Embedded System | ← 0 | 5 → | **Project category:** Embedded System |
| **Degree of safety criticality:** High | ← 0 | 5 → | **Degree of safety criticality:** High |
| **Quality standard**: High | ← 0 | 4 → | **Quality standard**: High |
| **Product type:** New | ← 0 | 5 → | **Product type:** New |
| **Time constraints:** Low | ← 0 | 5 → | **Time constraints:** Low |
| **Cost constraints:** Medium | ← 0 | 2 → | **Cost constraints:** Medium |
| **Team size:** Medium (62) | ← 0 | 5 → | **Team size:** Medium (59) |
| **Acquaintance of the domain:** Medium | ← 0.25 | 3 → | **Acquaintance of the domain:** High |
| **Degree of RE knowledge:** Medium | ← 0 | 5 → | **Degree of RE knowledge:** Medium |
| **Degree of knowledge about requirements:** Medium | ← 0 | 1 → | **Degree of knowledge about requirements:** Medium |
| **Other Attributes:** | | | **Other Attributes:** |
| **Availability of skilled facilitator:** Medium | ← 0 | 1 → | **Availability of skilled facilitator:** Medium |
| **Stakeholder heterogeneity:** High | ← 0 | 1 → | **Stakeholder heterogeneity:** High |
| **Degree of innovation of the project:** Medium | ← 0.25 | 1 → | **Degree of innovation of the project:** High |
| **Customer availability:** High | ← 0 | 1 → | **Customer availability:** High |
| **Degree of the importance of reusability:** Medium | ← 0 | 1 → | **Degree of the importance of reusability:** Medium |
| **Degree of the importance of eliciting implicit knowledge:** Low | ← 0 | 1 → | **Degree of the importance of eliciting implicit knowledge:** Low |
| **Degree of outsourcing:** Very Low | ← 0 | 1 → | **Degree of outsourcing:** Very Low |

(a) **Recommended RE techniques:**
**Elicitation:** 1. JAD 2. Interview
**Analysis & Negotiation:**
1. Scenario-Based Approach 2 Goal-Oriented Approach 3. SDL 4. AHP
**Documentation :**
1. UML 2. SDL 3. Goal-Oriented Approach
**Verification & Validation:**
1. Formal Requirements Inspection 2. SDL 3. Scenario-Based Approach.
**Tool support:**
RequisitePro, TAU

**Heuristics:**
Tailoring some of these techniques can require significant effort.
……
**ENDFRAME**

**(a)**

(b) **Recommended RE techniques:**
**Elicitation:**
**Analysis & Negotiation:**
**Documentation:**
**Verification & Validation:**
**Tool support:**

**Heuristics:**
……
**ENDFRAME**

**(b)**

$$\text{Similarity}(X_i, Y) = \sqrt{\frac{1}{\sum_{j=1}^{k} W_j * (F(x_{i,j}, y_j))^2}} = 0.77$$

Notes: 1. $W_i$ is the weight of the *i*th attribute. 5 represents "very important", 1 represents "unimportant"
2. The left and right arrows indicate the mapping between the retrieved case and the new case.

**Fig. 13** Similarity between a retrieved case and new case

the IPOS project was identified. The retrieved case, called Integrated Furnaces Control System, contains a set of recommended techniques which were used during the development of the Integrated Furnaces Control System and served as an initial recommendation to the new project. Figure 13a shows the retrieved case, Fig. 13b shows the new IPOS project, and at the bottom of the figure the calculation of the similarity between the two cases is shown. These initially recommended techniques make up $T_{IR}$ (see second column of Table 5).

Step 3. Analysis of RE techniques using clustering

The following tasks are part of this step:

(a) Based on the scores of the attributes of the given project, the requirements engineers chose technique attributes which were considered essential for the selection of RE techniques for the given project. These attributes are shown in Table 6.
(b) Assign a weight to each technique attribute selected in step (a) (see Table 6). The technique attributes shown in the first column of Table 6 were considered very important for the IPOS project. Consequently, the weights of these attributes were assigned high values (numeric value 4 or 5) by requirements engineers. The weights of other attributes (not shown in the table) were set to 1, which indicates unimportant. Such attributes are not considered when clustering is conducted.
(c) Set the number of the clusters $P$. Based on past experience of the authors, the number of clusters was selected to be $P = 9$ (see the explanation given in Sect. 5).


المنارة للاستشارات
www.manaraa.com

**Table 5** Technique recommendation space

| | Initial recommendation based on CBR ($T_{IR}$) | Recommendation based on requirements engineers' expertise | Functionally comparable techniques | Functionally complementary techniques | Recommendations from assent guidelines |
|---|---|---|---|---|---|
| Elicitation techniques ($T_e$) | JAD, Interview | | | Focus Group, Ethnography | Brain Storming |
| Aanalysis techniques ($T_a$) | Scenario-Based Approach (Use Case), Goal-Based Approach, SDL, AHP, | State Machine (Deterministic finite) | OO Analysis | | Fault-Tree Analysis |
| Documentation techniques ($T_d$) | SDL, UML, Goal-Based Approach | | | | |
| Verification and validation techniques ($T_v$) | Formal Inspection, SDL, Scenario-Based Approach | | | | |
| Tools | RequisitePro | | | | |

$T_e$ recommended elicitation techniques; $T_a$ recommended analysis and negotiation techniques; $T_d$ recommended documentation techniques; $T_v$ recommended verification and validation techniques; *Tools* recommended tools for the given project. The suitability of the tools is evaluated separately

**Table 6** The most important technique attributes selected by requirements engineers

| Most important technique attributes | Weight of attributes |
|---|---|
| Ability to help get domain knowledge | 5 |
| Ability to help identify stakeholders | 4 |
| Ability to help identify non-functional requirements | 5 |
| Ability to help model and understand requirements | 5 |
| Ability to help analyze non-functional requirements | 5 |
| Ability to help model interface requirements | 5 |
| Ability to help verify requirements automatically by using the notation | 4 |
| Ability to help write unambiguous and precise requirements by using the notation | 4 |
| Ability to help write complete requirements | 5 |
| Ability to help management of requirements | 4 |
| Ability to help identify interaction (inconsistency, conflict) | 5 |
| Maturity of supporting tool | 5 |

(d) Conduct technique clustering. Using the Fuzzy clustering algorithm, the RE techniques were assigned to the nine clusters. The result of the clustering is presented in Table 7.

Step 4. Construction of the technique recommendation space $T_{RS}$

The construction of the technique recommendation space was done by requirements engineers in the case study. This step, in turn, includes the following sub-steps:

(a) Analyzing the techniques in $T_{IR}$, to ensure that all the techniques are compatible with the new project. No technique had any conflict with the new project based on the Guidelines and Rules retrieved through frame-based reasoning. Therefore, $T_{US}$ is an empty set and $T_{IR}$ remains unchanged in this step.

(b) Selecting "Deterministic Finite State Machine" (one type of "State Machine") as one of the modeling tools for the project was suggested by the requirements engineers, since they were familiar with this technique and considered it necessary for the modeling and documenting of the requirements. Thus, $T_{ER}$ = {Deterministic Finite State Machine}. However, the inclusion of the "State Machine" into the recommendation space triggers the "Cost Reduction" rule. The "Cost Reduction" rule suggests that the two formal methods State Machine and SDL should not both be used in the same project. Thus, these two formal methods have to be assigned to different combinations in the recommendation space later in Step 5.

(c) Identifying all techniques that are functionally comparable and functionally complementary to all the techniques in ($T_{IR} \cup T_{ER}$) by using the results of the clustering in step 3. As presented in Table 5, both "Focus group" and "Ethnography" are considered as functionally complementary techniques to "Interview". "OO Analysis" is considered as functionally comparable to "Scenario-Based Approach" based on the clustering in the last step. The inclusion of functionally comparable techniques and functionally complementary techniques in each cluster provides a good opportunity to explore alternative techniques that can be chosen for the new project.

(d) Additionally, two techniques "Brain Storming" and "Fault-Tree Analysis" are recommended for the project based on the Assent Guidelines retrieved as



المنارة للاستشارات
www.manaraa.com

**Table 7** Result of the clustering

| Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 | Cluster 5 | Cluster 6 | Cluster 7 | Cluster 8 | Cluster 9 |
|---|---|---|---|---|---|---|---|---|
| Interview | Evolutionary Prototypes | Viewpoint-Based Documentation | Designer as Apprentice | SDL | Real-Time Structured Analysis | Object-Oriented Analysis (OOA) | Goal-Oriented Verification | Focus Group |
| Contextual Inquiry | Exploratory Prototypes (throw-away prototype) | Structured Natural Language Specification | Ethnography | Z | Structured Analysis (SA) | Representation Modeling | Formal Requirements Inspection | JAD |
| Brain Storming and Idea Reduction | XP | XP | AHP | LOTOS | Problem Frame Oriented Analysis | Unified Modeling Language (UML) | Requirements Checklists | SQFD (software QFD) |
| Viewpoints-Based Elicitation | | | Card Sorting | State Charts (also known as State Diagrams) | Decision Tables | Scenario-Based Approach | Requirements Testing | Soft Systems Methodology (SSM) |
| XP (eXtreme Programming) | | | Repertory Grids | State Machine | Functional Decomposition | | Utility Test | Future Workshops |
| | | | Laddering | Petri-Nets | Fault-Tree Analysis | | Viewpoint-Based Verification and Validation | |
| | | | | Goal-Oriented Approach | Viewpoint-Based Analysis | | Document Mining (document inspection) | |
| | | | | | Entity-Relationship Diagrams (ERDs) | | | |

It is worth mentioning that XP, taken as a technique in this research, appears in three clusters. The main reason is that the membership of XP is almost the same within each of the three clusters. This reveals the fact that XP partially addresses the requirements elicitation stage (using "customer online method"), the requirements analysis stage (mainly use "prototyping method"), and the documentation stage (mainly use "user story card"). Arguably, XP can also be used for requirements validation. However, since its membership to cluster eight (the cluster containing requirement verification techniques) is very low, it is not included in this cluster



www.manaraa.com

the result of the Frame-based reasoning according to the project attributes.

(e) Combining the techniques identified in steps (a), (b), (c) and (d) resulted in the technique recommendation space $T_{RS}$. The requirements engineers are involved in this step. The result of the recommendation space is shown in Table 5 (all the techniques in columns 2–6).

Step 5. Calculation

Requirements engineers analyzed the techniques in $T_{RS}$ with the help of the Assent Guidelines and the frame-based reasoning mechanism in RETKL. They then selected different combinations of RE techniques from $T_{RS}$ (see Table 8). Function (5) was used to select that combination of RE techniques that has the highest capability at the lowest cost. As illustrated in Table 9, the abilities, cost and the overall ability (sum of abilities and cost) of each technique was calculated based on formula (4). The scores computed using the objective function (formula 3) for each techniques combination are shown in Table 10. As can be seen from Table 10, the techniques combination 5 ($T_5$) has a score of 29.2 which is the highest overall ability among all the combinations of techniques. Thus, $T_5$ was recommended to the software project.

Step 6. Refinement and consistency examination of the recommended techniques

The final recommended combination of RE techniques $T_5$ generated in Step 5 is shown in the third column of Table 11. However, this recommendation was still subject to the judgment of the requirements engineers as the objective function is only used for decision support rather than decision making. Based on specifics of the case study, the requirements engineer decided that informal priority assignment is sufficient and therefore AHP was removed. No exclusive techniques were identified in the consistency examination using frame-based reasoning. However, the Completeness Reference Model suggested that an RE management tool be included in the final technique set. In this project, two software tools Rational Rose and RequisitePro were strongly recommended to support the RE process. The company had already licenses to both of them. Rational Rose was used to aid during requirements analysis and verification; RequisitePro was used to aid requirements documentation, management and analysis. The final outcome of the RE techniques selection process is given in the fourth column of Table 11. In order to apply the combination of RE techniques to the given project, training is provided before the actual application of the RE techniques began. The training included RE techniques, process management, and team work. We found that training was essential for the requirements engineers "to own" the requirements engineering process and techniques and implement them in the project.

## 6.2 Results analysis

The requirements engineers used the recommended RE techniques in the RE process for the given project in software company Y. Data was collected based on metrics predefined in the plan of the case study in order to examine the effects of the recommended RE techniques on the software project. The first column in Table 12 contains the

**Table 8** Technique combinations

| No. | Technique combination | $T_e$ | $T_a$ | $T_d$ | $T_v$ |
|---|---|---|---|---|---|
| 1 | $T_1$ | Interview, Focus Group, Ethnography | OO Analysis, AHP, Fault-Tree Analysis, State Machine | UML | Formal Inspection |
| 2 | $T_2$ | Interview, Brain Storming, Ethnography | Scenario-Based Approach, SDL, AHP, Fault-Tree Analysis | UML | Formal Inspection, SDL |
| 3 | $T_3$ | Interview, JAD, Ethnography | Goal-Oriented Analysis, AHP, Fault-Tree Analysis | Goal-Oriented Approach | Formal Inspection |
| 4 | $T_4$ | Interview, Focus Group, Ethnography | SDL, AHP, Fault-Tree Analysis | SDL Documentation | SDL, Formal Inspection |
| 5 | $T_5$ | Interview, Focus Group, Ethnography | State Machine, AHP, Fault-Tree Analysis, Scenario-Based Approach | UML | Formal Inspection |
| 6 | $T_6$ | Interview, Brain Storming, Ethnography | State Machines, AHP, Fault-Tree Analysis, Scenario-Based Approach | UML | Formal Inspection |
| 7 | $T_7$ | JAD, Interview, Ethnography | State Machines, AHP, Fault-Tree Analysis, Scenario-Based Approach | UML | Formal Inspection |



المنارة للاستشارات
www.manaraa.com

**Table 9** The calculation results of the abilities of the techniques in recommendation space ($T_{RS}$)

| Name of techniques in TRS | Abilities (Attribute 1–28) | Cost of the techniques for the project | Overall ability (Attribute 1–32) | Notes |
|---|---|---|---|---|
| Interview | 5.2 | 0.6 | 4 | $B$ = 0, user knew the technique |
| Focus Group | 5.8 | 1.2 | 3.4 | $B$ = 0, user knew the technique |
| JAD | 5.6 | 1.8 | 2 | $B$ = 1, users did not know the technique |
| Ethnography | 4.8 | 1.4 | 2 | $B$ = 1, users did not know the technique |
| Brain Storming | 3.8 | 0.8 | 2.2 | $B$ = 0, user knew the technique |
| State Machine | 9.6 | 1.4 | 7.6 | $B$ = 0, user knew the technique |
| Scenario-Based Approach | 10.2 | 0.8 | 8.6 | $B$ = 0, user knew the technique |
| OO Analysis | 3.6 | 0.4 | 2.6 | $B$ = 0, user knew the technique |
| AHP | 1.6 | 1.2 | −0.8 | $B$ = 1, users did not know the technique |
| Goal-Oriented Approach | 11.6 | 2.4 | 6.8 | $B$ = 1, users did not know the technique |
| SDL | 11 | 2.8 | 5.4 | $B$ = 1, users did not know the technique |
| Fault-Tree Analysis | 3.4 | 1.0 | 1.4 | $B$ = 0, user knew the technique |
| UML | 6.6 | 1.8 | 3 | $B$ = 1, users did not know the technique |
| Formal Requirements Inspection | 2 | 1 | 0 | $B$ = 0, user knew the technique |

**Table 10** Scores for each combination of RE techniques

| No. | Technique combination | Scores of the objective function computed for each techniques combination based on (3) |
|---|---|---|
| 1 | $T_1$ | 22.2 |
| 2 | $T_2$ | 25.8 |
| 3 | $T_3$ | 15.4 |
| 4 | $T_4$ | 16.8 |
| 5 | $T_5$ | 29.2 |
| 6 | $T_6$ | 28.0 |
| 7 | $T_7$ | 27.8 |

data that was measured in the case study. The data collected during the IPOS project was compared with another project, the so-called 3F System—a system for hospital deployment and management on the battlefield. The project 3F System has very similar project attributes compared to the IPOS project, even though it is from a very different application domain. In addition, data of the 3F System had been recorded previously, thus allowing a comparison of the 3F System and the IPOS project. As can be seen from Table 12, the "Number of developers involved" and the "Number of analysts involved" in the two projects are very similar with only one junior developer added in the IPOS project. Based on data analysis after the completion of the IPOS project, we found strong indicators that the KASRET approach had a positive impact on the software project. Some of the improvements of the IPOS project over the 3F System project are highlighted in the following (more details can be found in Table 12):

- The % of requirements elicited using the requirements elicitation technique is about 20% higher in the IPOS project compared to the 3F project.
- The % of requirements modified during the requirements verification stage was more than twice as high in the IPOS project than in the 3F project.
- The % of requirements that changed during the design stage was cut into half in the IPOS project compared to the 3F project.
- Both projects went over time. However, the 3F project exceeded the planned time by 18.8% while the IPOS project exceeded the schedule by only 9.0%.
- The cost overrun (measured in person-months) of the IPOS project was about 9.8% points less than that of the 3F project.
- The percentage of requirements that changed after the start of design was 5.3% points less in the IPOS project compared to the 3F project.
- The most significant gain from using the recommended techniques is that no major requirements (i.e., requirements that have a major impact on the overall system structure and overall system functionality) changed in the IPOS project after the start of the design. On the other hand, 12 major requirements changed after the start of the design of the 3F project. This is likely the main factor contributing to the reduction of the time delay of the IPOS project (two months behind schedule) compared to the 3F project (six months behind schedule).

Additionally, a survey was conducted after the completion of the IPOS project. Managers, all developers, and


المنارة للاستشارات
www.manaraa.com

**Table 11** Recommendation and final decision for the project

| Categories | Initial recommendation | Final recommendation | Final decision |
|---|---|---|---|
| Elicitation ($T_e$) | JAD, Interview | Interview, Focus Group, Ethnography, | Focus Group, Interview, Ethnography |
| Analysis and negotiation ($T_a$) | Scenario-Based Approach (Use Case), Goal-Oriented Approach, SDL, AHP | State Machine, AHP, Fault-Tree Analysis, Scenario-Based Approach | State Machine, Fault-Tree Analysis, Scenario-Based Approach |
| Documentation ($T_d$) | SDL, UML, Goal-Oriented Approach | UML | UML |
| Verification and validation ($T_v$) | Formal Inspection, SDL, Scenario-Based Approach | Formal Requirements Inspection | Formal Requirements Inspection |
| Tools | | | RequisitePro |

requirements engineers (analysts) involved in the project participated in the survey to ensure the coverage of time, space, and people as required by the triangulation principle. The questions and answers are presented in Table 13. The following conclusions can be made:

- The overall management team really appreciated the overall performance of the RE process and the results of the project.
- All the requirements engineers were pleased with the overall performance of the selected techniques used in the RE process.
- More than 70% of the developers agreed that the selected RE techniques contributed significantly to the success of the RE process and the software project.

In summary, the majority of the members involved in the project appreciated the high quality of the requirements gained by using the recommended techniques and good practices. The RE techniques selected and used in the project were very helpful in reducing the overall delay of the software project developed and improving the overall quality of the software specification.

We acknowledge that the data collected from the two projects might not be sufficient to claim that the improvements are solely due to using the KASRET approach. There are other factors that might have impacted the case study. Some of the likely factors are: the maturity of the team, the accuracy the data collected, the attitude of developers towards the recommended RE techniques, the impact of introducing new techniques into the RE process for the IPOS project, and the impact of the management and authors' involvement in the project. Additionally, the extra effort and cost associated with using the selected RE techniques have to be taken into account. Some of these issues are discussed in the next section. However, it is beyond the scope of the paper to discuss them in detail.

The feedback from the requirements engineers and developers involved in the IPOS project was very positive based on both quantitative and qualitative data as shown in Tables 12 and 13. All the requirements engineers and developers agreed that the savings from reduced rework outweigh the additional effort of applying KASRET. This suggests that KASRET made a significant contribution to the success of the IPOS project.

## 7 Conclusions and future work

In this paper, we proposed a knowledge-based approach called KASRET for supporting the selection of the most suitable RE techniques for a given project. Several observations and findings are derived based on the qualitative and quantitative data collected throughout this research:

- Existing research in RE provides us with considerable information about RE techniques. The collection, organization and management of knowledge are possible, yet can be very challenging due to the diverse nature of the knowledge. The challenges include the diversity, complexity, granularity of RE techniques; lack of suitable guidelines, and help for the identification and use of RE techniques.
- Effective knowledge representation is essential for successful knowledge management and reasoning. The representation of RE knowledge using a multi-dimensional schema to facilitate the reasoning and analysis of the techniques is one of the salient features of the research and novel in RE.
- The RE techniques recommended by the KASRET approach did not require any modifications after the start of the IPOS project. This indicates that the KASRET approach offered the most suitable RE process model and techniques for the project.
- Not all features of a technique have to be used. While doing the case study we found that it is sufficient to use the essential functionality of a technique to achieve the objective of the RE process rather than use all the features of a technique. The partial use of RE techniques has already been discussed in [45].



www.manaraa.com

**Table 12** Comparison between IPOS and 3F System

| Measured Data / Project Name | | Intelligent Power Optimization System (IPOS) | 3F System |
|---|---|---|---|
| RE Techniques used | | $T_e$: Focus Group, Interview, Ethnography<br>$T_a$: State Machine, Fault Tree Analysis, Scenario-Based Approach<br>$T_d$: UML<br>$T_v$: Formal Requirements Inspection<br>Tool: RequisitePro | $T_e$: Informal Focus Group<br>$T_a$: OO Modeling, State Machine<br>$T_d$: Informal Documentation<br>$T_v$: Informal Review<br>Tool: Text-based documentation |
| Total number of (atomic) requirements in the final requirements specification | | 1232 | 1776 |
| Number of analysts involved (playing the role of requirements engineers as well) | | 6 | 6 |
| Number of developers involved | | 60 | 59 |
| Number of original requirements known before the project began | | 725 | 1042 |
| Number of requirements elicited using RE techniques | Absolute | 412 | 496 |
| | % of the total number of requirements | 33.4% | 27.9% |
| Number of requirements added during verification and validation | Absolute | 41 | 65 |
| | % of the total number of requirements | 3.3% | 3.6% |
| Number of requirements modified during verification and validation | Absolute | 164 | 116 |
| | % of the total number of requirements | 13.3% | 6.5% |
| Number of requirements discovered during the design stage | Absolute | 32 | 102 |
| | % of the total number of requirements | 2.6% | 5.7% |
| Number of requirements discovered during the testing stage | Absolute | 22 | 71 |
| | % of the total number of requirements | 1.8% | 4.0% |
| Number of requirements changed after start of design | Absolute | 54 | 173 |
| | %of the total number of requirements | 4.4% | 9.7% |
| Number of *major* requirements changed after start of design | Absolute | 0 | 12 |
| | % of the total number of requirements | 0 | 0.7% |
| Percentage of overall requirements change after start of design | | 4.4% | 9.7% |
| Planned time for the project | | 22 months | 32 months |
| Time actually spent on the project | | Less than 24 months | More than 38 months |
| Effort in person-months | Planned | 1320 | 1888 |
| | Actually spent | 1440 | 2242 |
| Cost overrun in terms of effort in person-months | Number | 120 | 354 |
| | % increased over planned | 9.0% | 18.8% |

3F is an abbreviation for a system for hospital deployment and management in battlefield carried out by company Y. A major requirement is defined as a requirement which has a major impact on the overall system structure and overall system functionality

$T_e$ Stands for "Elicitation Technique"; $T_a$ Stands for "Analysis and Negotiation Technique"; $T_d$ Stands for "Documentation Technique"; $T_v$ Stands for "Verification and Validation Technique"; *Tools* indicates "Requirements Management Tool"

- Ethnography is a relatively unknown technique and not regularly used in software projects. However, it was strongly recommended in this project due to the fact that the project team was not familiar with the problem domain of the project. The use of Ethnography in this project led to the discovery of essential scheduling and management functions which would have otherwise been overlooked.
- Requirements engineering is not the sole duty of requirements engineers [46]. The involvement of developers and senior management in the RE process under the leadership of requirements engineers had a positive impact on the project. This conclusion is consistent with results reported in [47].

It is worth mentioning that the case study presented in this paper is used as an example to indicate the help that the KASRET approach provided for the IPOS project and the contribution it made to the success of the project. We also acknowledge that comparing the data from the two projects



المنارة للاستشارات
www.manaraa.com

**Table 13** Qualitative data from survey

| Types of people involved in the survey | Number of people involved in each type | Questions for the survey | Number of people selecting “strongly agree” | Number of people selecting “agree” | Number of people selecting “medium” | Number of people selecting “disagree” | Number of people selecting “strongly disagree” |
|---|---|---|---|---|---|---|---|
| Management | 5 | (1) The selected RE techniques are really the most suitable techniques for project IPOS | 2 | 3 | 0 | 0 | 0 |
| | | (2) The selected RE techniques helped to reduce rework for the overall project 1 | 1 | 4 | 0 | 0 | |
| | | (3) The RE techniques used in the project were very helpful in reducing the overall delay of the software project | 1 | 3 | 0 | 0 | 0 |
| | | (4) The overall satisfaction of the customers regarding the final requirements specification is high | 2 | 3 | 0 | 0 | 0 |
| | | (5) The RE process is better organized than before | 2 | 3 | 0 | 0 | 0 |
| Developers | 60 | (1) The RE techniques can be understood easily | 23 | 19 | 14 | 3 | 1 |
| | | (2) The frequency of the requirements changes is reduced with comparison to the software project 3F System and similar projects done before. | 21 | 23 | 12 | 2 | 2 |
| | | (3) The requirements are more understandable than before | 20 | 19 | 17 | 3 | 1 |
| | | (4) The notations used in the specification are acceptable and easily understandable | 18 | 24 | 13 | 3 | 2 |
| | | (5) The specification is easily traceable | 19 | 22 | 15 | 3 | 1 |
| | | (6) The RE techniques used in the project were very helpful in reducing the overall delay of the software project | 22 | 19 | 14 | 4 | 1 |
| | | (7) The overall quality of the documentation is high | 19 | 21 | 15 | 3 | 2 |
| Requirements Engineers (including Analyst, Architects) | 6 | (1) The RE techniques used address the major issues of IPOS project | 3 | 2 | 1 | 0 | 0 |
| | | (2) The notations are suitable for modelling the requirements | 2 | 4 | 0 | 0 | 0 |
| | | (3) Some other notations such as UML will be very helpful for the modeling | 1 | 2 | 3 | 0 | 0 |
| | | (4) It will be more productive if some more powerful tool such as Rational Rose and Rational Pro are used for requirements documentation and modeling | 0 | 3 | 3 | 0 | 0 |
| | | (5) The difficulty and cost to apply a new technique is not very high if appropriate training is available even though it has never been used before. | 1 | 3 | 2 | 0 | 0 |
| | | (6) The RE process is more organized than before | 3 | 3 | 0 | 0 | 0 |
| | | (7) The KASRET approach is very helpful for developing the most suitable project model for a particular project | 2 | 3 | 1 | 0 | 0 |



www.manaraa.com

cannot be used as formal proof that the KASRET approach will always provide the best solution for a software project. The following factors reduce the validity of the case study:

- Management commitment: The management of the two projects had slightly different levels of commitment to the RE process. Management in the IPOS project provided good support for using the RE process and techniques during the case study. The increased involvement of management and the authors in the case study had a positive impact on the success of the IPOS project. However, the positive impact might have been counteracted by the negative impact that a change of the development process has on the first project implemented with the new process. A detailed discussion about the negative impact of process changes on an organization can be found in [38]. We therefore believe that the involvement of management and authors in the case study was not a major reason for the positive results of the IPOS project.
- Learning effects and training. Learning effects from project to project play a considerable role if projects are in the same domain. However, since the two projects are in two different domains, the learning effects were considered minimal and did not noticeably contribute to the success of the IPOS project.
- The accuracy of the data derived from the case study. It is not realistic to assume that all the data derived have equal levels of accuracy. However, the authors' involvement throughout the entire RE process of the case study has reduced the likelihood of errors and ensured overall accuracy of the data.
- Other factors: Factors related to the personal attitudes and experiences in the application of the KASRET approach in the project might also have influenced the selection of RE techniques.

Moreover, the following aspects are also considered as limitations of the research:

- A full-fledged tool is still not available. This reduces the applicability of the methodology in practice since the generation of a solution is very time consuming.
- The current version of the RETKL contains information about 46 RE techniques. Although this is one of the most comprehensive collections of information about RE techniques, it still is not complete.
- Several steps in the KASRET approach still require the involvement of requirements engineers. Automating the entire RE technique selection process is subject to future research.

In summary, this research made the following key contributions:

- A requirements technique library was established which includes detailed knowledge of RE techniques.
- KASRET combines different advantages of knowledge representation schemata and reasoning abilities which make the knowledge retrieval process more efficient. Particularly, this combination provides mechanisms for the efficient management of diverse requirements technique knowledge.
- The objective function used in the approach provides a criterion for effectively helping the selection of RE techniques.
- An explicit link is established between project attributes and the characteristics of RE techniques. This provides a mechanism to ensure that the selected RE techniques are suitable for the given project.

Our future work will focus on the refinement of the approach and on building a comprehensive tool based on the approach. Once this is completed, it will provide more opportunities for the evaluation of the approach in practice.

Coincidently, the case study supports the fundamental assumption made by the RE community that getting high-quality requirements early on will reduce rework and overall development cost [1, 48]. One of the encouraging facts is that the company is committed to collaborating with us in future and applying the KASRET approach in future projects.

**Acknowledgements** The authors would like to thank the editors and the 4 anonymous reviewers for their very detailed and valuable comments that helped improve this paper.

## 8 Appendix

Table 14.

**Table 14** Acronyms/symbols and their definition

| Acronyms/ symbols | Definition |
|---|---|
| 3F System | Hospital deployment and management system on the battlefield |
| AHP | Analytic hierarchy process |
| $C(t)$ | Represent functionally complementary techniques to technique $t$ |
| CBR | Case-based reasoning |
| CORE | Concern of requirement engineering |
| $F(t)$ | Represent functionally comparable techniques to technique $t$ |
| FBR | Frame-based reasoning |
| IPOS | Intelligent power optimization system |
| JAD | Joint application development/joint application design |



المنارة للاستشارات
www.manaraa.com

**Table 14** continued

| Acronyms/ symbols | Definition |
|---|---|
| KASRET | Knowledge-based approach for the selection of requirements engineering techniques |
| OO | Object-oriented |
| RE | Requirements engineering |
| REPKB | RE process knowledge base |
| RETKL | RE techniques knowledge library |
| SDL | Specification and description language |
| $T_{ER}$ | Engineers recommended techniques |
| $T_{IR}$ | Initial set of recommended RE techniques |
| $T_{RS}$ | Techniques recommendation space |
| $T_{US}$ | A set of techniques that are NOT suitable for the given project |
| UML | Unified modeling language |
| XP | Extreme programming |

## References


1. Kotonya G, Sommerville I (1998) RE, processes and techniques. Wiley, New York
2. Damian D, Chisan J (2006) An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management. IEEE Trans Softw Eng 32(7):433–453
3. Brooks F (1987) No silver bullet: essence and accidents of software engineering. Computer 20(4):10–19
4. Claus C, Freund M, Kaiser M, Kneuper R (1999) Implementing systematic requirements management in a large software development programme. In: Proceeding of fifth international workshop on requirements engineering foundation of software quality, pp 33–42
5. El-Emam KE, Birk A (2000) Validating the ISO/IEC 15504 measure of software requirements analysis process capability. IEEE Trans Softw Eng 26(6):541–566
6. Jiang L, Eberlein A, Far BH (2004) Evaluating the requirements engineering process using major concerns. In: Proceeding of IASTED international conference on software engineering, February 17–19, Innsbruck, Austria
7. Nikula U, Sajaniemi J, Kalviainen H (2000) A state-of-the-practice: survey on RE in small- and medium-sized enterprises. Telecom Business Research Center, Lappeenranta Research Report
8. Glass RL, Vessey L (1995) Contemporary application-domain taxomonies. IEEE Software 1995, pp 63–76
9. Nuseibeh B, Easterbrook S (2000), RE: a roadmap. In: Anthony F (ed) The future of software engineering. ACM, New York
10. Sutcliffe A (1997) A technique combination approach to requirements engineering. In: 3rd IEEE international symposium on requirements engineering (RE'97), January 05–08, Annapolis
11. Mannio M, Nikula U (2001), Requirements elicitation using a combination of prototypes and scenarios. Technical Report. Telecom Business Research Center, Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland, Lappeenranta, 2001, ISBN 951-764-528-7
12. Macaulay LA (1996) Requirements engineering, applied computing. Springer, Heidelberg
13. Berry D, Damian D, Finkelstein A, Gause D, Hall R, Wassyng A (2005) To do or not to do: if the requirements engineering payoff is so good, why aren't more companies doing it?" Proceeding of Requirements Engineering 2005.
14. Damian D, Zowghi D, Vaidyanathasamy L, Pal Y (2004) An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software. Int J Empir Softw Eng 9(1–2):45–75
15. Hall T, Beecham S, Raner A (2002) Requirements problems in twelve software companies: an empirical analysis. IEE Proc Softw 149(5)
16. Humphrey W, Snyder T, Willis R (1991) Software process improvement at Hughes aircraft. IEEE Softw 8(4):11–23
17. Davis AM (1993) Software requirements, objects, functions and states. Prentice-Hall, Englewood Cliffs
18. Jiang L (2005) A framework for requirements engineering process development. University of Calgary, PhD Thesis, Sept. 2005
19. Extreme Chaos (2001) The Standish Group International
20. Jiang L, Eberlein A, Far BH (2004) A methodology for RE process development. In: ECBS 11th IEEE international conference and workshop on the engineering of computer-based systems, Brno, May 2004
21. Maiden N, Rugg G (1996) ACRE: selecting methods for requirements acquisition. Softw Eng J 11(3):183–192
22. Hickey AM, Davis AM (2003) Elicitation technique selection: how do experts do it? In: Proceedings of the 11th IEEE international requirements engineering conference, pp 169–178
23. Jones C (2003) Variations in software development practices. IEEE Softw 20(6):22–27
24. Macaulay L (1996), Requirements for requirements engineering techniques. In: IEEE second international conference on requirements engineering
25. Hickey AM, Davis AM (2003) Requirements elicitation and elicitation technique selection: a model for two knowledge-intensive software development processes. In: Proceedings of the 36th annual Hawaii international conference on system sciences, 6–9 January 2003, pp 96–105
26. Tsumaki T, Tamai T (2005) A framework for matching RE techniques to project characteristics and situation changes. In: First international workshop on situational requirements engineering processes, in conjunction with 13th IEEE international requirements engineering conference, August 29th–September 2nd 2005, Paris
27. Bickerton JM, Siddiqi J (1993) The classification of Requirements engineering methods. In: Proceedings of IEEE international symposium on requirements engineering, 4–6 Jan 1993, San Diego
28. Davis GB (1982) Strategies for information requirements determination. IBM Syst J 21(1):4–31
29. Browne GJ, Ramesh V (2002) Improving information requirements determination: a cognitive perspective. Inform Manage 39:625–645
30. Lauesen S (2002) Software requirements: styles and techniques. Addison-Wesley, Reading
31. Kumar K, Welke RJ (1992) In: Cotterman W, Senn J (eds) Methodology engineering: a proposal for situation specific methodology construction, challenges and strategies for research in systems development. Wiley, Chichester, pp 257–266
32. Brinkkemper S (1995) Method engineering: engineering the information systems development methods and tools. Inform Softw Technol 37(11)
33. Brinkkemper S, Lyytinen K, Welke R (1996) Method engineering, principles of method construction and tool support. Chapman & Hall, London
34. Reifer DJ (2003) Is the software engineering state of the practice getting closer to the of the art? IEEE Softw 20(6):78–83
35. Turban E, Aronson JE (2001) Decision support systems and intelligent system, 6th edn. Prentice-Hall, Upper Saddle River




المنارة للاستشارات
www.manaraa.com

36. Holsapple CW, Whinston AB (2000) Decision support systems: a knowledge-based approach. West Publishing Company, Minneapolis/St Paul, ISBN 0-314-06510-5
37. Jiang L, Eberlein A, Far BH, Majid M (2004) An innovative requirement engineering technique selection model. Technical Reports, April 31st, 2004, Department of Electrical and Computer Engineering, University of Calgary, http://www2.enel.ucalgary.ca/~ljiang/research/indexOfResearch.htm
38. Jirotka M, Goguen JA (1959) Requirements engineering: social and technical issues. Academic, London
39. Helmer O, Rescher N (1959) On the epistemology of the inexact science. Manage Sci 6(1):25–28
40. Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches, artificial intelligence communications, vol 7:1. IOS Press, pp 39–59
41. Russell S, Norvig P (1995) Artificial intelligence; a modern approach. Prentice-Hall, Englewood Cliffs
42. Fikes R, Kehler T (1985) The role of frame-based representation. Reason Commun ACM 28(9):904–920
43. Jiang L, Eberlein A (2006) Clustering requirements engineering techniques. In: IASTED international conference on software engineering and applications (SEA 2006), November 13–15, Dallas
44. Kitchenham B, Pickard L, Pfleeger SL (1995) Case studies for method and tool evaluation. IEEE Softw, pp 52–62
45. Fitzgerald B (1996) An investigation of the use of system development methodologies in practice. In: Fourth European conference on information systems, pp 143–161
46. Jiang L, Eberlein A, Far BH (2005) Combining requirements engineering techniques—theory and case study. In: ECBS 12th IEEE international conference on the engineering of computer-based systems, Greenbelt, April 2005
47. Chaos (1999) A recipe for success. The Standish Group International
48. Broadman J, Johnson D (1996) Return on investment from software process improvement as measured by U.S. industry. Crosstalk 9(4):23–29


Springer
المنارة للاستشارات
www.manaraa.com

Copyright of Requirements Engineering is the property of Springer Science & Business Media B.V. and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.