A Framework for Matching Requirements Engineering Techniques to Project Characteristics and Situation Changes

Toshihiko Tsumaki [a] and Tetsuo Tamai [b]

[a] Nihon Unisys, Ltd.

[b] The University of Tokyo

Abstract

One of the most difficult jobs for requirements engineers is to select an appropriate RE method for a project at hand. Good engineers make good choice and have skills in applying the selected technique appropriately. Poor engineers usually have a narrow choice range limited by their training and biased by their experience. Once a requirements engineering technique that does not fit the current project is selected, the project is doomed to fail. In this paper, we propose a framework to characterize typical requirements engineering techniques and use it as a base for selecting appropriate techniques at the time of starting a project as well as at the time of recognizing a change in the project nature or encountering an obstacle in defining a suitable set of requirements.

Keywords: Requirements engineering techniques, Project characteristics, Situation change, Domain decomposition, Goal oriented, Scenario based, Brainstorming

1 Introduction

One of the major causes for the difficulty of requirements engineering tasks is the wide variety of target domains and project characteristics. As the result, a number of requirements engineering techniques have been created, advocated and practised in their efforts to cover the variety of scope. This brings about another problem of how to select an appropriate requirements engineering technique for a given project.

Projects can be characterized by various factors such as target domains, system type and size, and organization cultures, each of which may be further decomposed into more concrete attributes. If a requirements engineering method that does not fit to the character of the project is chosen, the project is doomed to fail. However, most of requirements engineering literature, including papers, textbooks, tutorials and tool manuals are just eager to emphasize merits of some specific techniques without sufficiently mentioning their suitable target areas or their limitations. Admittedly, there are some work dealing with the problem of requirements method selection (Maiden, 1996; Hickey et al., 2003), but we think it is important to have a general framework that relates the requirements method space with the project characteristics space.

Another issue in applying requirements engineering methods is how to cope with the project situation changes over time. During the requirements process and even after the requirements specification is completed, the project may go through a substantial change, due to changes in business environments, stakeholders and technologies. In some cases, although the project characteristics themselves have not changed, understanding of the project nature by the requirements engineers changes. We often experience cases when the requirements process runs into obstacles and find that the initial conception of the project properties was in-correct. In such situations, we have to adopt a new requirements engineering method to overcome the obstacles.

 Proceedings of SREP'05, Paris, France, August 29–30, 2005
Eds. Ralyté J, Ågerfalk PJ and Kraiem N

In this paper, we propose a framework to characterize typical requirements engineering techniques and use it as a base for selecting appropriate techniques at the time of starting a project as well as at the time of recognizing a change in the project nature or encountering an obstacle in defining a suitable set of requirements.

The structure of the paper is as follows. In section 2, a classification framework for requirements engineering techniques is introduced. It is represented as a two-dimensional space named RE technology map. In section 3, various project characteristics are studied in relation to the RE map and compatibility of techniques with project characteristics are discussed. The problem of situation change in the project is dealt with in section 4 and strategies for re-selecting RE methods according to the situation change are discussed. A case study of a conference management system for corporate in-house software engineering symposiums is explained in section 5. Related work is surveyed in section 6 and our work is concluded in section 7.

2 Classification of RE Techniques

Among a number of requirements engineering techniques, we pick up four typical and widely practised ones: the domain decomposition, goal oriented, scenario based and brainstorming approaches to study requirements engineering method properties. We mainly focus on the early requirements process, dealing with requirements elicitation techniques. Modelling techniques including object-oriented modelling, state based modelling, etc. and their notations are important and some requirements analysis methods are strongly tied with such modelling techniques but we intentionally draw a boundary and concern only with pure requirements elicitation methods. These four techniques are not only important but clearly differ from each other in nature.

Let us briefly overview the four approaches.

Domain decomposition: The target domain is decomposed into sub-domains in a step by step manner. It is easy to list up requirements without oversight if the sub-domains to be handled are decomposed into sufficiently small, manageable size. The process can also be executed in the bottom-up way, composing finer grained sub-domains or requirements into larger ones. The study on such kind of domain analysis methods has a long history. A typical example is the work by Rieto-Diaz (1990), which can be traced back to the Draco project by Neighbors (1984). The approach can be generalized as categorization methods and we may include approaches of classifying other entities related to requirements besides problem domains, e.g. business processes and groups of stakeholders.

Goal oriented: A number of goal-oriented approaches have been proposed (Mylopoulos et al., 1992; Dardenne et al., 1993; Anton, 1996; Lamsweerde, 1991). Among them, KAOS (Lamsweerde et al., 1998) is particularly a formal approach, whereas GBRAM (Anton, 1996) is particularly an informal approach.

Scenario based: Scenario based approaches are also quite popular (Potts et al., 1994; Holbrook,1990). A use case is considered as an integrated set of scenarios. Thus, use case approaches (Jacobson et al., 1992; Dano etal., 1997) are basically in the same category but more oriented towards the later RE phase.

Brainstorming: Brainstorming is effective in eliciting requirements particularly in the area where system development experience is scant or a new kind of soft-ware is expected. The KJ method, for example, a sophisticated brainstorming method, was created by Jiro Kawakita more than 40 years ago and has been widely practised in industries not necessarily in the software industry (Kawakita, 1991). Application to software requirements engineering is reported by Takeda et al.(1993).

To characterize the differences, we consider two dimensions, one concerning elicitation operation types and the other concerning target object types. We divide the operation type dimension into two categories: static and dynamic and also divide the object type dimension into closed and open types. Table 1 maps the four techniques on this two dimensional space.

Table 1: Techniques and dimensional space.

operation / object	static	dynamic
closed	domain decomposition	scenario
open	goal	brainstorming

The first dimension captures how the requirements elicitation and acquisition process is conducted, either focusing on static structures or dynamic behaviours.

1. Static

Requirements are elicited from the domain examining its static structure. They are collected and sorted in a structured and systematic way. There are rules or guidance in decomposing/composing or classifying/clustering requirement related objects, including (sub) domains, goals and requirements themselves.

2. Dynamic

Requirements are elicited from the domain focusing on their dynamic context. For many stakeholders, it is often easier to think in terms of procedural behaviours or situational changes over time. Requirements can be un-methodically enumerated or generated by stake-holders or by analysts. Thus, the way of eliciting requirements is also regarded as an “dynamic” activity, encouraging inspiration and imagination.

The second dimension concerns with the properties of the target space to be analyzed, either relatively closed or open.

1. Closed

The object space is relatively stable, known and closed. It can be grasped by focusing mainly on forms.

2. Open

The object space is relatively unstable, unknown, changing and open. To explore the space, meanings have to be much considered.

Now, other various requirements engineering techniques can also be mapped on this plane. We call the plane RE technology map or RE map in short. Fig. 1 shows the RE map with typical techniques allocated. Some patterns are observed on the map, which are interesting to note.

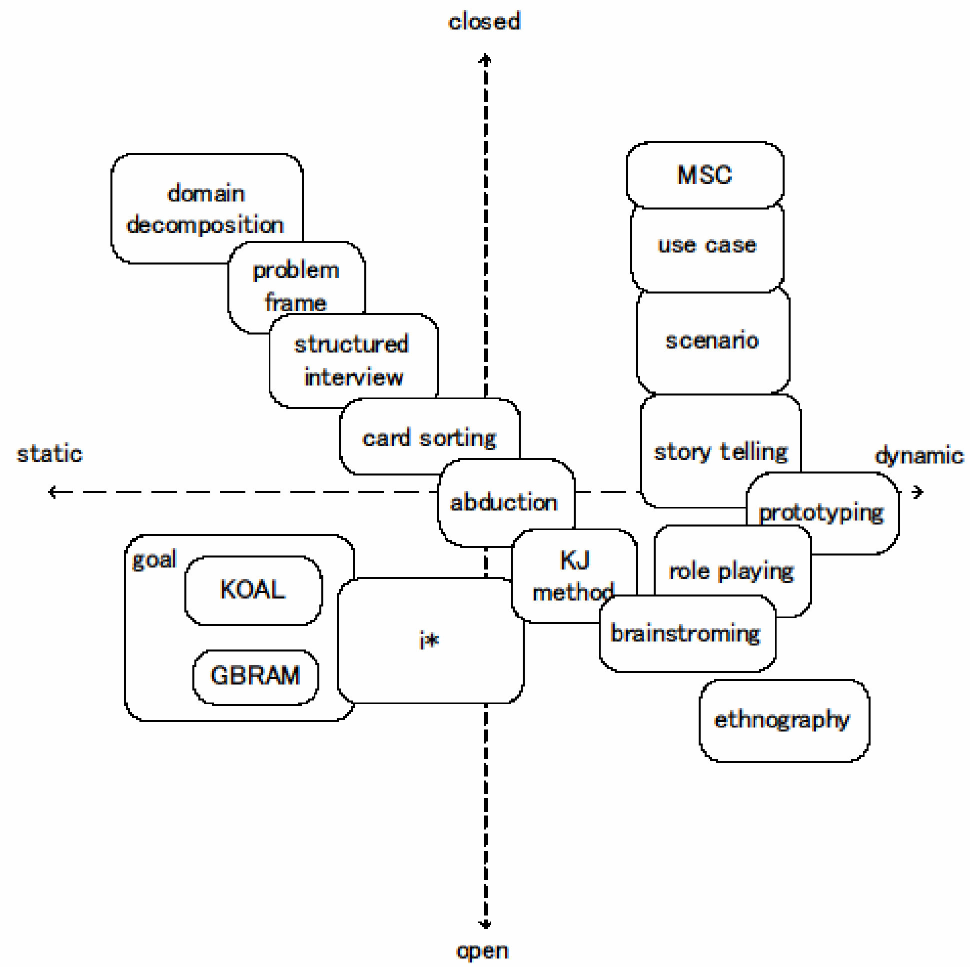

Figure 1: RE Technology Map.

1. In the area of the right hand side half plane, a sequence of scenario based techniques can be observed: from top to bottom, MSC (Message Sequence Chart), use case, scenario, story telling and role playing. They share the property of eliciting requirements along the course of dynamic event progress but different in formality. Techniques in the upper region are relatively suitable for investigating a closed space in form or syntax conscious manner, while in the lower region are techniques that are suitable for investigating an open space, thus tending to be more human oriented. Also related is the prototyping approach. Its major objective is to simulate dynamic interaction.

2. Along the diagonal from the upper left to the lower right is observed a sequence of knowledge acquisition and classification type approaches: from upper-left to lower-right, domain decomposition, structured interview, card sorting, abduction, KJ method, brainstorming, and ethnography. The diagonal direction can be interpreted as the convergent (upper-left)/ divergent (lower-right) axis.
3. Among the goal-oriented approaches, KAOS is for-mal and thus positioned upper, while more informal GBRAM is positioned lower. The i* framework (Yu, 1997) overlaps with the goal oriented approach but extends to the right direction. There are many proposals that unite the goal oriented approach and the scenario based approach (Rolland at al., 1998). They bridge the lower-left quadrant to the upper-right quadrant (, which is not shown in the RE map).
4. As mentioned before, both the top-down and bottom-up approaches of decomposing/composing entities are included in the upper-left quadrant. The KJ method has a process of clustering cards produced by brain-storming and thus located closer to upper-left. The problem frame approach (Jackson, 2001) can be regarded as an-other way of decomposing the problem domain, al-though the decomposition is not structured in top-down hierarchy but in parallel juxtaposition.
5. Various techniques of idea generation type can be included in the lower-right quadrant. For example, the abduction based approach such as (Russo et al., 2002) can be allocated here.

Some methods combine techniques classified in different quadrants. For example, multiple viewpoint approaches can be positioned in the structured-closed quadrant in that they elicit requirements based on the classification of various viewpoints but they are usually combined with scenarios, goals or other techniques to handle requirements from each viewpoint (Nuseibeh et al., 1994).

3 RE Techniques Compatibility to Project Characteristics

Requirements engineering techniques have been chosen and applied to projects at hand by requirements engineers according to their intuition or experience. Good engineers make good choice and have skills in applying the selected technique appropriately. Poor engineers usually have a narrow choice range limited by their training and biased by their experience. Once a requirements engineering technique that does not fit the current project is selected, the project will trace the road to failure.

The properties of requirements engineering techniques introduced in the last section should be matched with project characteristics and used for guiding the method se-lection process.

We capture characteristics of projects from five factors: application domain, requirements engineer types, information resources, user involvement and requirements proper-ties. We will map each factors on the RE technology map basically as one dimensional line that indicates the direction of method fitness. We will show it symbolically by a double-headed arrow in a square like Fig. 2. The topology of the square corresponds to the plane of Fig. 1.

3.1 Application Domain

The direction shown in Fig. 2 corresponds to application domain stability. On one side is a type of applications that are relatively stable. Knowledge of the target domain and experience of developing similar systems are well accumulated, thus the domain can be perceived as a closed space and can be examined by a mechanical, syntactic means.

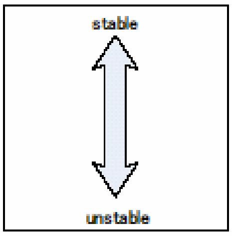

Figure 2: Domain Stability Factor.

On the other side is a type of relatively new, unknown or unstable applications. To explore requirements of such applications, analysis handling semantics at a certain deep level is required. The domain is perceived as an open space.

In both cases, the static-dynamic dimension is relatively irrelevant. For example, to elicit requirements of an unfamiliar application domain, either the goal-oriented type approach or the brainstorming type approach will do and the choice should be made by considering other factors.

3.2 Requirements Engineers Type

The direction of Fig. 3 corresponds to requirements engineers' disposition. The left hand side engineers tend to like logical and systematic way of thinking. The right hand side engineers tend to prefer imaginative and intuitive way of thinking. Put in another way, the left side people are relatively theory oriented, while the right side people are relatively experiment oriented. To engineers of each group, techniques found in the corresponding half space of the RE map will fit better.

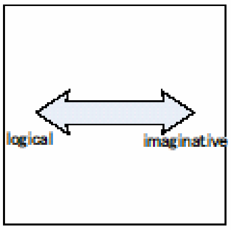

Figure 3: Engineers' Disposition Factor.

3.3 Information Resource

When available information is abundant in the form of documents or knowledge base or through investigating existing systems, the space to be explored can be confined and the closed side techniques may be suitable. Information can be arranged and put in order through a mostly syntax based means. On the other hand, when available information is scarce or implicit, requirements have to be explored in an open space, using the open side techniques.

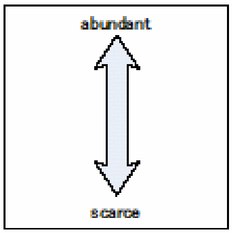

Figure 3: Information Resource Amount Factor.

3.4 User Involvement

The user involvement factor is related to the information resource factor. If information is available as formal knowledge, user involvement is not necessarily required to collect requirements, although it is still desirable to have users' collaboration. But in order to proceed with methods like goal searching or brainstorming, users' participation is indispensable.

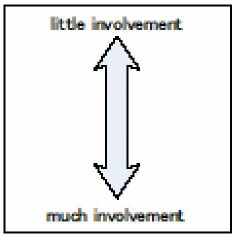

Figure 4: User Involvement Factor.

3.5 Requirements Quality

Quality of requirements is assessed by a number of criteria. The IEEE standard 830-1998 lists up correctness, un-ambiguity, completeness, consistency, ranking, verifiability, modifiability and traceability. All these properties are important but the priority between them may vary according to the project nature.

Techniques in the static-closed quadrant are suitable for covering the domain in a methodical way, so that completeness in the sense of complete coverage of the target area is attainable. Techniques in the dynamic-closed quadrant, typically the scenario based approach, are suitable for attaining behavioural consistency. Specifically, inconsistency in the order of work flow and data flow can be detected relatively easily. Also, as the requirements are captured in the dynamic context, they can be matched to the behaviour of the implemented system, so that traceability is in general easily attainable. Techniques in the static-open quadrant, typically the goal-oriented approach, are good at detecting conflicts between goals. Together with some conflict resolving method, it can be expected to obtain a logically consistent set of requirements with this type of techniques. Also, they are effective in comparing different goals to set ranking among the requirements. Techniques in the dynamic-open quadrant are good at producing a wide variety of requirements from different angles, which may include prospected future requirements, so that they will help foresee requirements adaptation.

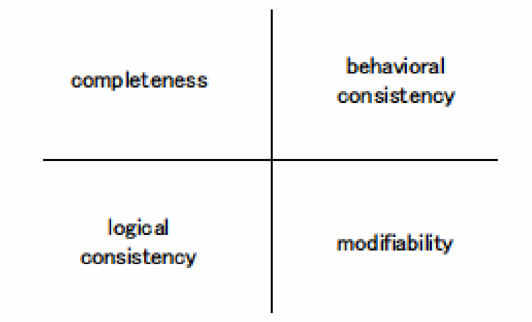

Figure 4: Quality Type.

3.6 Non-Functional Requirements

Some of non-functional requirements (NFR) are more easily elicited by specific techniques. A branch of the goal-oriented approach is well-known for covering NFR employing the concept of soft goals (Mylopoulos et al., 1992).

Security concerns can be handled effectively by using the scenario based approach, because it is essential for defending system attacks to explore all possible scenarios attackers may take in invading the system. Listing up possible scenarios is also effective in avoiding other kind of risks that may possibly be materialized.

Ease of use may be particularly well covered by the prototyping method, giving users opportunities to experience interactions and touch and feel the interface.

4 Coping with Situation Change

A project starts with an appropriate requirements engineering method selected based on analysis explained in the previous section. However, a project goes through transition whenever a new issue is raised and solved (Tsumaki, 2004). A project may encounter obstacles during the requirements analysis phase or the requirements specification may be found inadequate in the later phases. Based on our experience and consulting literature like (Christel et al., 1992), we made a list of problems that may arise in the requirements elicitation activity as shown in Table 2.

These are, in a way, negation of requirements quality treated in section 3.5 but here we are interested in the events when such problems are detected. When they are detected, some measures should be taken and changing RE techniques is one of the most effective strategies to consider. We have shown affinity between requirements qualities and groups of elicitation techniques but when requirements problems as listed in Table 1 get exposed, the strategy to take is not necessarily simple application of corresponding techniques in affinity. Below are some cases worth noting.

Table 2: Requirements Elicitation Problems.

Incomplete requirements	
	incomplete understanding of needs

	incomplete domain knowledge
	poor users' collaboration
	overlooking tacit assumptions
Incorrect requirements	
	ill-defined system boundary
	misunderstanding of system purpose
Ambiguous requirements	
	synonymous and homonymous terms
	un-testable terms
Inconsistent requirements	
	un-solid intentions of requesters
	different views of different users
Unfixed requirements	
	fluctuating requirements
	continuous acceptance of additional requirements
Excessive requirements	
	unorganized bulky information source
	too many requesters
	over commitment by sales persons
	unnecessary design consideration

Completeness: Techniques in the upper left quadrant are effective in detecting incompleteness but not necessarily good in filling in insufficient requirements. To acquire missing requirements, techniques in the lower half of RE map, e.g. the goal-oriented approach and the brainstorming, had better be employed. After collecting new requirements through such methods, it would be desirable to go back to the upper left quadrant to sort and place them under a classification scheme to check completeness.

Conflict: When conflicts between requirements are detected, straightforward application of the scenario based approach in case of behavioural conflict and the goal-oriented approach in case of logical conflict may be effective.

Excessive requirements: Some techniques in the upper left quadrant are used for top-down decomposition while some are used for bottom up clustering. When the problem is requirements excess in amount, such clustering type techniques will be effective. When the problem is how to give priority among them, the goal oriented approach coupled with an evaluation mechanism like AHP (Analytical Hierarchical Process) may be effective.

Ambiguity: Ambiguity arising from the way of expressing requirements is basically orthogonal to requirements eliciting techniques but dependent on modelling and representation languages. However, if the problem is lack of concreteness, moving towards the upper direction of the RE map may help. For example, proceeding the goal-oriented approach to obtain operational goals or formalizing scenarios as message sequence charts.

5 Case Study

As a case study, we take a project of developing a conference supporting system. The system is for a series of in-house software engineering symposiums having been held in the first author's company. At the time of the project, our framework for classifying RE techniques and guiding method selection was not completed and explicitly documented. The project was basically conducted based on the requirements analyst's experience and skill but our observation and consultation on the project helped us grow and refine the idea described in this paper.

5.1 Overview of the project

The symposium has been held each year, calling for paper submissions from members of the company. It is organized by the secretariat section; actually one person in the section has been in charge of the symposium for the past ten years. As the retirement age of the secretary was approaching, there was an urgent need to preserve his knowledge. Another objective of the system development was to make the symposium management task more efficient.

The decision to start the project was made but it was found that the secretary was too busy at the time to participate in the project. Instead, a pile of unsorted documents of the past ten years, including minutes of meetings, exchanged messages, submitted papers and unofficial memos of the secretary, were passed to the requirements analyst.

5.2 Project Characteristics

The basic characteristics of this project can be summarized as follows.

- The target domain is relatively limited and small.
- On the other hand, the development organization did not have experience of developing similar systems, so that the domain was rather new to them.
- Stakeholders' involvement was weak.
- Available information source was large but knowledge was unorganized.

5.3 RE Process

The requirements analysis process was actually taken in four steps.

Step 1 – arranging information to capture the domain: The first step was to understand the process of organizing symposiums and tasks to be taken by the secretary. As the secretary's commitment was not obtained, the only source was the unorganized materials. The classifying technique in the static-closed quadrant fitted well for this purpose. The fact that the domain was relatively small and closed also enhanced its compatibility. As for a criterion for arranging the information, division by action processes was selected, because the work for organizing symposiums naturally flows from planning, calling for papers, paper selection, holding the symposium and to publishing the proceedings. This judgment was corroborated by the fact found by the analyst who examined the materials that an author who has submitted a paper goes through the stages of the general employee, the paper contributor, the accepted contributor, and the speaker.

Each process was further divided into sub-processes, e.g. the planning process was divided into setting up the organizing committee, setting up the program committee and assigning reviewers, holding the organizing and program committee meetings and setting schedule.

All the documents were arranged based on this process division structure and redundant or unimportant information were discarded.

Step 2 – understanding the tasks of organizing symposiums: The documents had been sorted, which helped understanding the structure of the process but the dynamic features of each concrete task were not well grasped yet. To clarify the tasks, the scenario writing method in the dynamic-closed quadrant fitted well. The analyst himself had to write scenarios based on the sorted documents supplemented by some interviews to people who had participated in the past symposiums as committee members. An example of the scenario is:

- Call for papers
  1. Set a theme for the special session.
  2. Write a call-for-paper announcement.
  3. Send the call-for-paper to the PC members to solicit comments and update the call-for-paper if necessary.
  4. Send the call-for-paper to the organizing committee to get approval.
  5. Order printing of the call-for-papers.
  6. Distribute the call-for-papers with submission forms to all members of the company.

The written scenarios were sent to the secretary to verify its correctness. Some missing information was returned from him.

Step 3 – identifying requirements: The preceding steps were for the analyst to understand the target domain and process. Requirements for the system to be built had to be explicitly identified. For this purpose, the classification technique again in the static-closed quadrant was employed. This time, two-dimensional criteria were used to decompose the requirements space; one criterion was the requirement types: functional, performance and quality, the other was the requirement sources: business, operational users, and system. At this stage, the secretary became able to spare some time for the project, so weekly meetings of the analyst with the secretary were held to proceed this step. A highly simplified version of the results is shown in Table 3.

Table 3: Requirements Arranged in the Matrix.

	functional	**performance**	**quality**
business	plan schedule, call for papers, select reviewers, re-view, accept papers, notify authors	data volume: 300 papers	protect authors' data, protect reviewers' data, protect from illegal ac-cess
operational users	operation by intra-net, use of templates	operation latency: 30 seconds	easy operation, guide & help
system	alternate pass for intra-net	MTBF: 1 day	paper file 1 day

Step 4 – refining requirements: Requirements had been acquired but it was found that the level was not concrete enough. To refine the requirements, the goal-decomposition technique was employed to decompose goals down to the operational level. Some new requirements were discovered through the work, which were allocated in the requirements matrix made in Step 3. Also, the goals were extended to the upper direction through asking "why" questions so that the result was delivered to the management to confirm the project approval.

5.4 Discussion

As the target domain of the system is limited and small in size, the requirements elicitation techniques used in this case study project are mostly located in the upper half plane of the RE map. This fact is consistent with the observation presented in section 3.1. While the domain was small and relatively closed, the application type was rather new to the software engineers of the project, which made them perceive the "openness" of the domain. That must have been one of the reasons why the goal-oriented approach was used in the final step. However, the reason for using the goal decomposition was mainly to refine the requirements to the operational level as explained above so that the nature of the goal-oriented technique actually applied was more bent towards the formal or closed direction. Another objective of its use was to show the project justification to the management.

For its project size, the requirements analysis process in this case took an elaborate way of going through four steps. The four step process was not planned before starting the project but was the result of ongoing decision making. It gives one example of real requirements elicitation processes transiting through situational changes as discussed in section 4.

6 Related Work

There are not so many studies that share similar objectives as ours. Among them, Hickey & Davis (2003) and Maiden & Rugg (1994) are closely related.

Hickey & Davis made interviews to nine renowned experts, asking how they perform requirements elicitation. The style of the interviews were open-ended and "differed significantly from interview to interview", except that four cases of situations intentionally described relatively vaguely in brief sentences were presented to the interviewees to give answers how they would tackle the problem. The results were summarized to: 1) when to use techniques, 2) normalized responses to the four cases and 3) some other information. The survey is interesting and provides useful information. Relations between requirements techniques and project situations are discussed sporadically in an anecdotal style. Maiden & Rugg's work is more systematic. 12 requirements acquisition methods are selected and they are compared by six facets: purpose of requirements, knowledge types, internal filtering of knowledge, observable phenomena, acquisition context and method interdependencies. The results are given as five tables corresponding to the five facets (excluding interdependencies), each entry to the tables is a check or a symbolized weight. The proposed framework consists of these tables to be used in selecting requirements elicitation methods. They may be useful as checklists for selection but no example of using the framework is reported.

Some papers treat similar topics but their objectives or approaches are considerably different from ours. Yadav et al. (1988) proposes a comparison framework for requirements analysis techniques but what they compared were DFD and IDEF0, which are supposed to be used in the later stage of requirements analysis.

Hudlicka (1996) compares requirements elicitation techniques from the knowledge engineering and cognitive science viewpoint. Three knowledge elicitation techniques are selected and compared: repertory grid analysis, multidimensional scaling and hierarchical clustering. These techniques are not directly related to requirements elicitation but as they deal with information creation and classification techniques, they can be allocated along the lower-right to upper-left diagonal in our RE map. An experiment of comparing the three techniques in the context of airline safety inspection has been conducted, which is informative, but how they can be employed in acquiring system requirements is not explicitly discussed.

Coughlan & Macredie (2002) emphasize communication in the requirements elicitation process. As a result, they compare user participation type methodologies: MUST, JAD (Joint Application Design), ULRC (User-Led Requirements Construction) and SSM (Soft Systems Methodology). The difference of their approach from ours lies not only in the selection of method type but also in their focus on "methodologies" rather than techniques. Techniques at the level of brainstorming, scenarios, etc. are adopted in each methodology, some shared and others not shared.

Bergman & Mark (2003) deals with requirements at a yet higher level so that the requirement determines project selection as opposed to the usual case where a project determines requirements. The NASA program is analyzed, where a number of projects are proposed and the problem is to find an appropriate set of projects that satisfy the NASA higher level requirements. Certainly, such higher level requirements have to be analyzed and determined, which is an interesting part of the paper but the objective is different from our case.

Padula (2004) reports a real case of selecting requirements engineering processes at Hewlett-Packard. Two projects are compared in terms of RE process, not from the viewpoint of elicitation techniques.

7 Conclusion

We proposed a framework for selecting requirements engineering techniques, considering different factors of the project at hand. A broad view of various techniques is obtained as the RE map and selection factors can be projected on the map.

In order to use the framework for selecting techniques systematically, our results may better be summarized as checklists or a set of rules. It is a rather straightforward task to make such checklists or rules but we have not yet done that, because we consider it would be better to take some time in accumulating experience of applying the framework to many real projects.

Acknowledgement

This work has been conducted as an activity of the requirements engineering study group Mahoroba. The authors would like to thank to all members of Mahoroba.

References

ANTON AI (1996) Goal-based requirements analysis. *In Proc. International Conference on Requirements Engineering (ICRE'96)*, 136–144.

BERGMAN M and MARK G (2003) In situ requirements analysis: a deeper examination of the relationship between requirements determination and project selection. *In Proc. 11th International Requirements Engineering Conference (RE'03)*, 11–22.

CHRISTEL MG and KANG KC (1992) Issues in requirements elicitation. Technical Report CMU/SEI-92-TR-12.

COUGHLAN J and MACREDIE RD (2002) Effective communication in requirements elicitation: a comparison of methodologies. *Requirements Engineering Journal*, 7:47–60.

DANO B, BRIAND H and BARBIER F (1997) An approach based on the concept of use case to produce dynamic object-oriented specifications. *In Proc. 3rd International Symposium on Requirements Engineering (RE'97)*, 54–64.

DARDENNE A, LAMSWEERDE AV and FICKAS S (1993) Goal directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50.

HICKEY AM and DAVIS AM (2003) Elicitation technique selection: How do experts do it? *In Proc. 11th International Requirements Engineering Conference (RE'03)*, 169– 178.

HOLBROOK CH (1990) A scenario-based methodology for conducting requirements elicitation. *ACM SIGSOFT Software Engineering Notes*, 15(1):95–104.

HUDLICKA E (1996) Requirements elicitation with indirect knowledge elicitation technique: Comparison of three methods. *In Proc. International Conference on Requirements Engineering (ICRE'96)*, 4–11.

JACKSON M (2001) *Problem Frames: Analyzing and structuring software development problems*. Addison-Wesley.

JACOBSON I, CHRISTERSON M, JONSSON P and OVERGAARD G (1992) *Object-Oriented Software Engineering: A Use Case Driven Approach*. ACM press.

KAKITA J (1996) *The Original KJ Method*. Kawakita Research Institute.

MAIDEN N and RUGG G (1996) ACRE: Selecting methods for requirements acquisition. *Software Engineering Journal*, 183–192.

MYLOPULOS J, CHUNG L and NIXON B (1992) Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on Software Engineering*, 18(6).

NEIGHBORS J (1984) The draco approach to constructing software from reusable components. *IEEE Transactions on Software Engineering*, SE-10:564–573.

NUSEIBEH B, KRAMER J and FINKELSTEIN A (1994) A framework for expressing the relationships between multiple views in requirements specifications. *IEEE Transactions on Software Engineering*, 20(10):760–773.

PADULA A (2004) Requirements engineering process selection at Hewlett-Packard. *In Proc. 12th International Requirements Engineering Conference (RE'04)*, 296–300.

POTTS C, TAKAHASHI K and ANTON AI (1994) Inquiry-based requirements analysis, *IEEE Software*, 11(2):21–32.

ROLLAND C, SOUVEYET C and ACHOUR CB (1998) Guiding goal modeling using scenarios. *IEEE Transaction on Software Engineering*, 24(12).

RRIETO-DI´AZ R (1990) Domain analysis: An introduction. *ACM SIGSOFT Software Engineering Notes*, 15(2):47–54.

RUSSO A, MILLER R, NUSEBEH B and KRAMER J (2002) An abductive approach for analysing event-based requirements specifications. *In Proc. 18th International Conference on Logic Programming (ILPC-2002)*.

TAKEDA N, SHIOMI A, KAWAI K and OHIWA H (1993) Requirements analysis by KJ editor. *In Proc. IEEE International Symposium on Requirements Engineering*, 98–101.

TSUMAKI T (2004) Requirements engineering pattern structure. *In Proc. Asia-Pacific Software Engineering Conference*, 2004, pages 502–509.

LAMSWEERDE AV (2001) Goal-oriented requirements engineering: A guided tour. *In Proc. 5th IEEE International Symposium on Requirements Engineering (RE'01)*, 249– 263.

LAMSWEERDE AV, DARIMONT R and LETIER E (1998) Managing conflicts in goal-driven requirements engineering. *IEEE Transactions on Software Engineering, Special Issue on Managing Inconsistency in Software Development*, 24(11):908–926.

YADAV SB, BRAVOCO RR, CHATFIELD AT and RA-JIKUMAR TM (1988) Comparison of analysis techniques for information requirement determination. *ACM*, 31(9):1090–1097.

YU E (1997) Towards modelling and reasoning support for earlyphase requirements engineering. *In Proc. 3rd International Symposium on Requirements Engineering (RE'97)*, 226–235.