

# MISSING REQUIREMENTS AND RELATIONSHIP DISCOVERY THROUGH PROXY VIEWPOINTS MODEL

SEOK WON LEE AND DAVID C. RINE

---

**Abstract.** This paper addresses the problem of “missing requirements” in software requirements specification (SRS) expressed in natural language. Due to rapid changes in technology and business frequently witnessed over time, the original SRS documents often experience the problems of missing, not available, and hard-to-locate requirements. One of the flaws in earlier solutions to this problem has no consideration for missing requirements from multiple viewpoints. Furthermore, since such SRS documents represent an incomplete domain model, manual discovery (identification and incorporation) of missing requirements and relationships is highly labor intensive and error-prone. Consequently, deriving and improving an efficient adaptation of SRS changes remain a complex problem. In this paper, we present a new methodology entitled “Proxy Viewpoints Model-based Requirements Discovery (PVRD)”. The PVRD methodology provides an integrated framework to construct proxy viewpoints model from legacy status requirements and supports requirements discovery process as well as efficient management.

**Keywords:** Software Requirements Engineering, Proxy Viewpoints, Requirements Categorization, Missing Requirements Discovery

---

## 1. Introduction

This paper addresses the problem of “*missing requirements*” in software requirements specification (SRS) expressed in natural language. Due to rapid changes in technology and business frequently witnessed over time, the original SRS documents often experience the problems of missing, not available, and hard-to-locate requirements. Such problems can be further decomposed into the following sub-problems: 1) Earlier solutions do not consider missing requirements from multiple viewpoints; 2) SRS documents with many missing requirements typically tend to be poorly structured and maintained as well as hard-to-trace (by not providing links to related requirements); 3) SRS documents with missing requirements

represent an incomplete domain model; and 4) Manual discovery (identification and incorporation) of missing requirements is highly labor intensive and error-prone. These inherent rigid subproblems do not allow efficient adaptation of SRS changes and improvements. Most SRS documents today are plagued by a combination of one or more of these problems, and they become even more prevalent while dealing with legacy status [AS99] SRS. Therefore, there is a strong need to develop a new methodology that can provide improved solutions to these problems and lengthen the life span of SRS.

In this paper, a new methodology entitled “*Proxy Viewpoints Model-based Requirements Discovery (PVRD)*” is presented to meet this need. Through the requirements discovery and analysis process, the PVRD methodology provides a way to construct proxy viewpoints models from legacy status natural language SRS documents. “*Proxy viewpoints*” is a surrogate and approximation of original viewpoints that would have been constructed if the requirements of the domain were well-engineered from the beginning of a software development life cycle by using one of the viewpoints oriented requirements engineering methods such as VORD [KS96, KS98].

The PVRD methodology consists of four models: viewpoints model, enterprise model, missing requirements types categorization model, and requirements discovery and analysis model. The *viewpoints model* represents different perspectives or views for coverage of direct and indirect stakeholders that need to be identified and incorporated into the legacy status software system requirements. The *enterprise model* provides a way of categorizing requirements based on systems engineering design process models. The *missing requirements types categorization model* provides a method to project a requirements space that may contain a specific type of missing requirements. The *requirements discovery and analysis model* provides a method to retrieve requirements of interest by using the requirements term expansion method that automatically generates a list of “potential query terms” which could assist analysts in acquiring more knowledge about the domain of interest by performing a “complete search” of available requirements resources.

Based on this integrated framework, the PVRD methodology is able to create a proxy viewpoints model and provides a new way of discovering missing requirements while improving the requirements representation space through the new indexing structure that supports multiple viewpoints from many stakeholders in a large-scale complex software system.

Well-designed explanatory scenario-based multiple case studies [Yin94, LR04a] are developed in the finance application system domain and educational information management system domain, not only as a way to validate the methodology but also to show its uniqueness and novelty and to provide exemplary guidance for researchers from academia and real practitioners from industry [Lee03]. Various evidence and findings that support the propositions of this study validate that the PVRD methodology provides an integrated environment that supports a requirements discovery and analysis process as well as efficient management. In this paper, we present some examples from the first case study in the finance application system domain.

This paper is organized as follows. The next section deals with related work and the section 3 presents the proposed methodology with descriptions of each model in it. Section 4 presents a case study with explanations of each step of the methodology. We present some concluding remarks with future research.

## 2. Related Work

Viewpoints approach to requirements engineering [KS96, NKF94, SFE96] provides many ways of requirements organization and management. From the systems modeling aspect, some methods such as SADT [RS77] and CORE [Mul79] only considered viewpoints as sources or sinks of data flows, and VOSE [FKNG92] as engineering viewpoints. However, these methods can be inappropriate for many cases in real practice when organizational requirements and concerns need to be taken into account. For this reason, the PVRD methodology favors the similar viewpoints concepts of VORD [KS96, KS98] or PREView [SS97], which go beyond the sources or sinks of data flows and provide environments for requirements

elicitation, structuring, and management. While these approaches need to be applied at the very beginning stage of software development life cycle, the PVRD methodology provides a way of constructing proxy viewpoints model from the poorly structured requirements of legacy status software systems through several methods in the requirements discovery and analysis model.

[PVB95, KA97] discuss several inspection techniques for detecting, diagnosing, and correcting errors in natural language requirements documents. Among them, checklist technique contains a list of questions that the analyst may use to assess each requirement. [KS98] suggests the list should not normally include more than 10 items and the questions on the checklist should usually be general than restrictive. Improvement of requirements quality through defects discovery involves many issues such as the types of requirements and their representation, types of defects, and the efficiency of the methods and applicability. Therefore, there is no single best universal approach for the requirements defects discovery and should depend upon many factors.

In REVERE [RGS00], a probabilistic natural language processing (NLP) tool is applied to free-text documents to retrieve requirements information. It uses statistical likelihood of the words for the classification purpose. The probabilities are derived from a large corpora of free-text that have already been analyzed and manually tagged for each term's certain set of categories (i.e. syntactic, semantic, or lexical categorizations). The generated log-likelihood figure heavily relies on the correctness and compatibility of the pre-tagged corpora to the application domain. However, the requirements term expansion technique in the PVRD methodology focus on the complete search of requirements of interest through the improvement of quality of query terms.

### **3. The PVRD Methodology**

The following sections describe the framework models incorporated in the PVRD methodology.

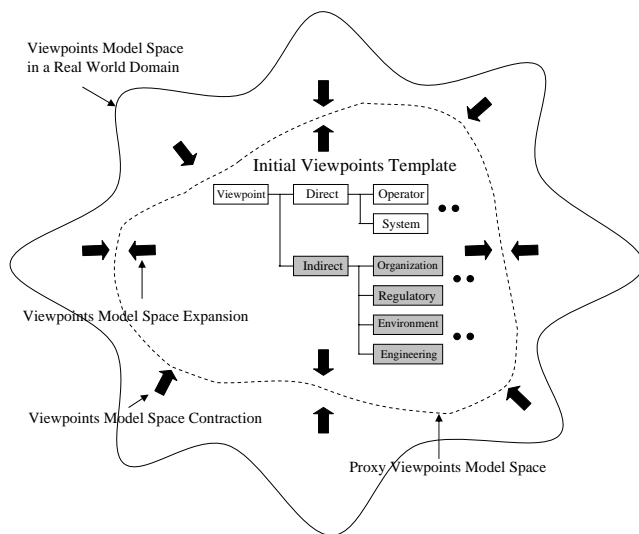


Figure 3.1: Evolutionary Viewpoints Model Space

### 3.1. Viewpoints Model

*Viewpoints model* represents different perspectives or views for coverage of direct and indirect stakeholders that need to be identified and incorporated into the legacy status software system requirements. We assume that legacy software system requirements were initially developed without considering viewpoints concept. The identification and conceptualization of initial viewpoints model takes place at the early stage of PVRD. However, these initial set of perspectives or views are partial and incomplete descriptions. As shown in Figure 3.1, the viewpoints model should adapt to necessary changes and evolve towards an optimal number of viewpoints and descriptions through the PVRD.

Although the need for developing viewpoints model in requirements engineering is well studied in [KS98], constructing viewpoints model from the legacy SRS, that was originally elicited and maintained without viewpoints consideration, remain a hard problem. The viewpoints model in PVRD methodology builds a good approximation of viewpoints and also facilitates the requirements discovery and analysis process while providing a good requirements management environment.

A good approximation of the viewpoints' characteristics can be achieved through some analysis techniques such as an interviewing process with subject matter expert/witness, a statistical sampling of viewpoints as strata (i.e. stratified sampling technique [Coc77]), or a documentary study of viewpoints.

We take the approach of "evolutionary viewpoints model" shown in Figure 3.1, which starts with an initial viewpoints model that includes a minimum set of viewpoints such as "direct viewpoint" and "indirect viewpoint" in VORD [KS96, KS98]. Alternatively, the model allows for multiple viewpoints that are partially built from any other available resources from the specific domain. If any "viewpoints template" of a particular domain exists, such template can also be used as an initial viewpoints model as well.

### 3.2. Enterprise Model

*Enterprise model* (EM) is a categorization of requirements that are used to define the design problem at various levels of detail in systems engineering [Bue99]. As shown in Figure 3.2, typically EM consists of six sets of requirements categories such as enterprise policies, mission need statements, operational concepts, initial requirements, derived requirements, and actual design requirements in the systems engineering design process.

In PVRD methodology, EM provides a way of organizing and managing requirements from the systems engineering design process point of view. Each requirement is indexed based on the defined roles/scopes of each category of EM as well as the identification of the corresponding viewpoint. The level of abstractness of individual requirement in each category of EM is also determined based on the granularity of viewpoints model. Because each requirements category in EM inherits the characteristics of each viewpoint in viewpoints model, the new indexing structure facilitates the requirements discovery and analysis process as well as the characterization of the requirements. Customization of the categories and their specific roles/scopes depends on the specific domain application.

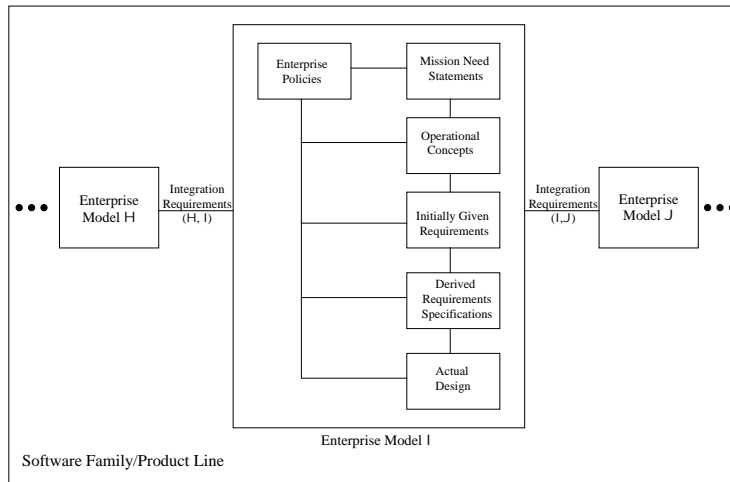


Figure 3.2: Requirements Categories in Enterprise Model

### 3.3. Missing Requirements Types Categorization Model

The *missing requirements types categorization model* provides a method to project a requirements space that may contain a specific type of missing requirements. [Mer01] describes an explorative study of possible missing requirements types in SRS. In general, they are: non-inclusion of all significant requirements; non-conformity to SRS standard; undesired event handling; omitted non-functional requirements; missing requirements due to a single point of failure for a critical system; non-reachable states or operational modes etc. In PVRD methodology, this ad-hoc classification scheme of missing requirements types is applied in a systematic way to the projection of requirements space that is associated with corresponding viewpoint and category of EM.

Figure 3.3 shows the projected requirements space from three different classifications which are: viewpoints, EM category, and missing requirements categorization.  $RS(i,j,k)$  represents all requirements space that belongs to viewpoint  $VP_j$ ,  $i$ th category of EM:  $EM_i$ , and  $k$ th missing requirements category:  $MRC_k$ . Theoretically  $RS(i,j,k)$  should satisfy all constraints from all three dimensions. For instance, let's assume that  $MRC_k$  is the "non-conformity to SRS standard". More specifically, those standards are "Z39.50 document interface standard" and

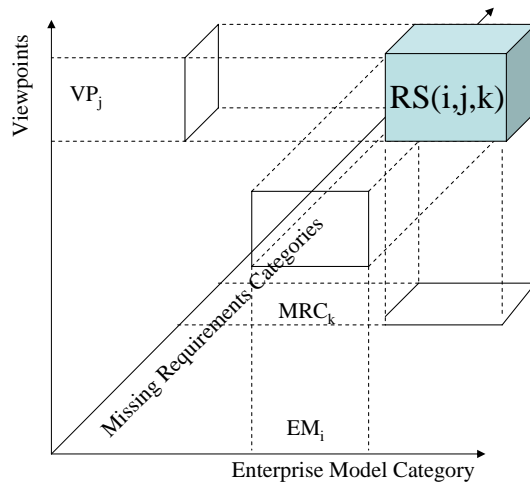


Figure 3.3: Requirements Search Space Projection through Missing Requirements Types

“ISO 10160-1 document ordering standard”.  $VP_j$  is the “document standards” and  $EM_i$  is the “actual design requirements”. Therefore, all requirements in  $RS(i,j,k)$  should conform to those standards. Thus,  $RS(i,j,k)$  is assessed for the possibility of missing requirements type  $MRC_k$ , and this projection provides a narrow, accurate and effective search space for discovering various types of missing requirements.

### 3.4. Requirements Discovery and Analysis Model

#### 3.4.1. Requirements Term Expansion

One of the fundamental problems of the complete search in information retrieval is “word mismatch” [XC00]. Similarly, SRS often contains different terms and descriptions that carry the same contextual information of the domain. Therefore, lack of requirements query terms or non-availability of domain knowledge can result in an incomplete search for specific requirements of interest. The requirements discovery and analysis model provides a method to retrieve requirements of interest by using the requirements term expansion method [Sal71, FO95] that



automatically generates a list of “potential query terms” which could assist analysts in acquiring more knowledge about the domain of interest by performing a “complete search” of available requirements resources.

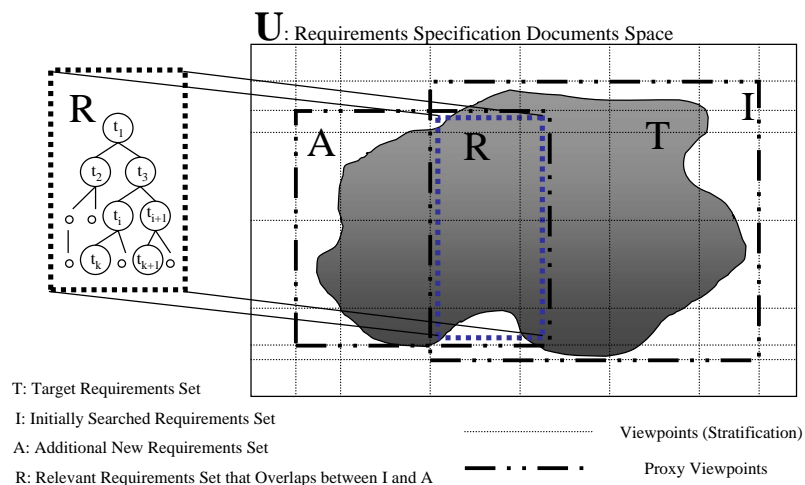


Figure 3.4: Requirements Search Space

Figure 3.4 shows how this technique can be applied to the requirements search space.  $U$  represents the entire requirements space available for searching.  $T$  (shaded area) represents a target requirements space that is most relevant to the user’s interests.  $I$  represents a set of requirements retrieved through the initial query. Our goal is to find and retrieve the set of requirements  $A$  that spans the remaining part of the target requirements space  $T$ , which was not retrieved in the space  $I$ , using the query expansion technique. In order to do that, query expansion technique automatically generates a list of terms in space  $R$  that can potentially characterize the set of requirements  $A$ . These terms are the *key domain words* that are actually relevant to the requirements that only belong to  $A$ . Depending on the granularity of rectangular boxes in the requirements space, it is possible to approximate proxy viewpoints based on the actual original viewpoints that should have been constructed.

### 3.4.2. Requirements Relationship Chain Discovery

Some of requirements often establish certain relationships between them and create relation chains across the categories in EM. For example, these relations include causal, is-a membership, general, special, any feature relationships etc. with many-to-one, one-to-one, or one-to-many correspondences as shown in Figure 3.5.  $r_a$  ( $[R_{ij-1k}, R_{i+1j-1k}], R_{i-1jk}$ ) represents many-to-one relationships.  $r_b$  ( $R_{i-1j-1m}, R_{i-1jk}$ ) represents one-to-one relationship.  $r_c$  ( $R_{i-1jk}, [R_{ij+1k}, R_{i+1j+1k}]$ ) represents one-to-many relationships. Relation chains such as  $r_a r_c$  and  $r_b r_c$  can also be discovered. These relationships represent not only relationships between requirements but also relationships of viewpoints and categories of EM.

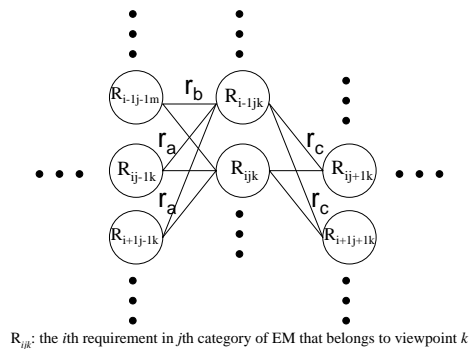


Figure 3.5: Requirements Relationship Chain Discovery

The relation chains can be also applied to Software Product Line (PL) or Software Family (SF) concepts. A set of verified requirements relation chains can be the “requirements feature template”. This template plays a key role in requirements verification of similar module/component in the PL or SF.

### 3.4.3. The PVRD Architecture

In order to provide a framework to reason about missing requirements problems, four models in the PVRD methodology are presented in the previous sections. The PVRD methodology is applicable in the use of existing natural language SRS

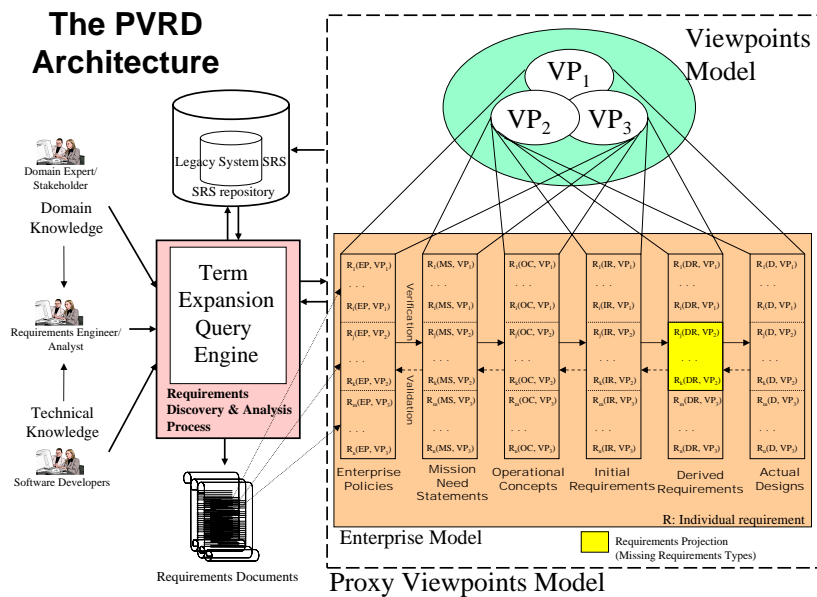


Figure 3.6: The PVRD Architecture

in further development of legacy systems by discovering missing requirements, especially, when it is necessary to reconstruct the original SRS, to elicit new requirements for system changes that will take place or to create a new system from a similar legacy system. In addition, the PVRD methodology creates an environment for comprehending, fixing and maintaining complex requirements interdependencies in software intensive systems. Figure 3.6 shows the PVRD architecture and each step of the methodology is described in the following section (see Figure 4.7).

#### 4. Case Study in a Finance Application System Domain

A case study design, as an evaluation research approach and a generalization from it, builds a basis for valid inferences from the case study events and evidence collected. For an effective case study as an empirical method applied as a validation exercise – applied to an ‘invented software (systems) engineering methodology’, it is necessary for the validation exercise to first have designed a case study

methodology specific to the characteristics of this invented software engineering methodology to be validated using the case study. Case study evaluation research method in [Yin94] is used to validate the PVRD methodology and more detailed case study designed methodology including its components, execution, and results are described in [Lee03, LR04a]. The following subsections describe each step of the methodology (as shown in Figure 4.7) by using some examples from a case study performed in a finance application system domain from a real project environment. The last two steps in Figure 4.7 are beyond the scope of the PVRD methodology.

#### **4.1. Step1: Set goals for the requirements search and investigation**

Any requirements search or investigation process should have a goal or a set of goals. For example, business, organization policy or technology changes may impact the existing software systems, and those change requests need to be adapted within a reasonable response time. In general, such requests need to be consolidated by a team of Subject Matter Experts (SMEs) that usually consists of domain experts, requirements engineers/analysts, and software engineers.

A goal can be represented as a set of *investigative questions* by using *key domain terms*. SMEs can use one of seven facets of defining a ‘complete requirement’: 1) What is it named?; 2) Who uses it?; 3) How is it used?; 4) When is it used?; 5) Where is it used?; 6) Why is it used?; and 7) How well is it used? SMEs can also use the general investigative question reduction logic (as shown in Figure 4.8) together in order to clarify their goals and questions more specifically. In this Step1, it is important for SMEs to acquire and use the key domain terms that best describe the concept of the goals in the investigative questions. Since these key domain terms play an important role in narrowing requirements search space.

#### **4.2. Step2: Selection of domain requirements terms**

From the requirements search or investigation point of view, having a set of specific goals with key domain terms is critically related to the creation of an initial

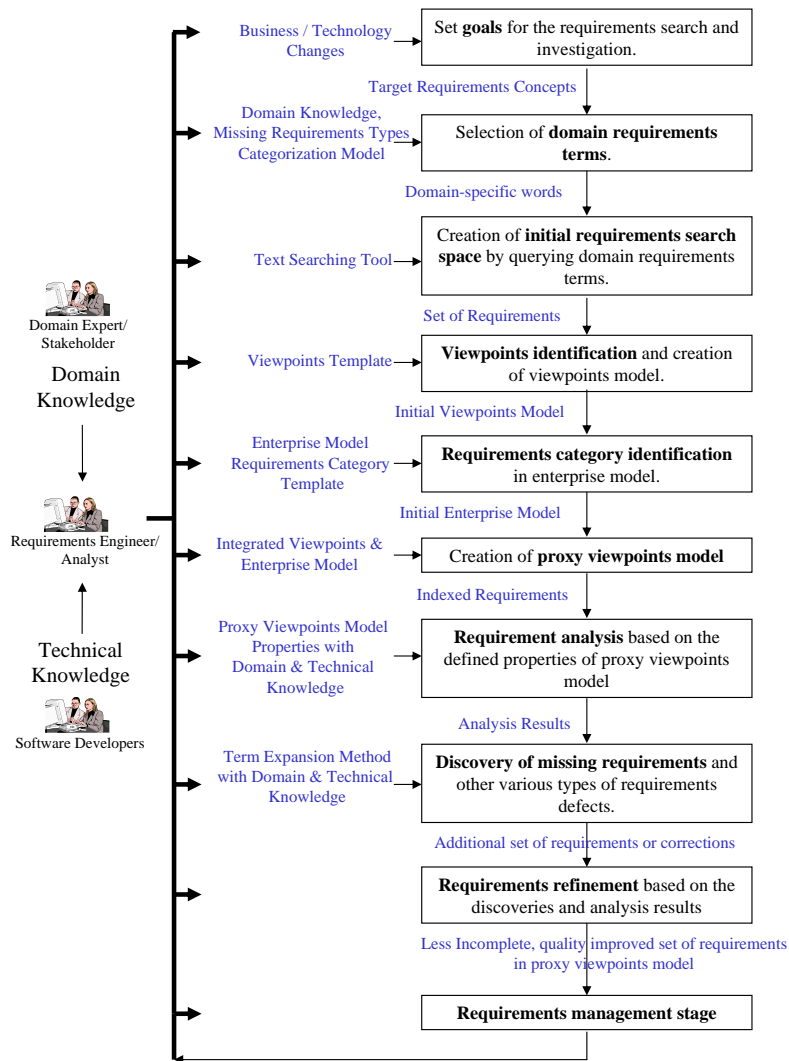


Figure 4.7: Steps of the PVRD Methodology

target requirements space. In other words, the target requirements concept (that can be described by using key domain terms) becomes the basis of *selecting domain requirements terms* that will be used to create an initial requirements search space.

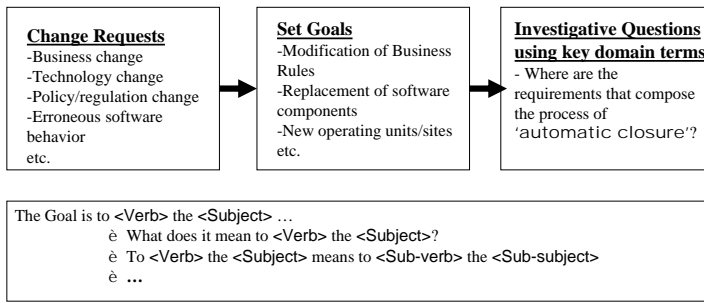


Figure 4.8: Step 1: Investigation Goals

There are four ways for SMEs to select domain requirements terms: 1) A term or a set of terms used in the description of the investigative questions; 2) A term or a set of terms that can characterize the categories of ‘missing requirements types’; 3) A term or a set of terms from the thesaurus if it is available in the project environment; 4) A domain term(s) from the consensus of different stakeholders across the organization. For instance, this can be achieved through the regular stakeholders meeting, discussions, collecting terms that carry out the same investigative goals but from the different stakeholders perspectives or views (i.e. ‘automatic closure’, ‘automatic closing’, and ‘automated processing’ are terms used by different stakeholders but mean the same concept).

### 4.3. Step3: Creation of initial requirements search space by querying domain requirements terms

Once a set of key domain terms are collected from Step2, SMEs use it to create an initial requirements search space. This initial set of requirements is the requirements where the set of domain terms appeared in the given actual requirements document.

#### 4.4. Step4: Viewpoints identification and creation of viewpoints model (VP)

Among SMEs, requirements engineers/analysts should have access to the requirements domain (through the domain experts) and technical requirements knowledge (through the software engineers) to identify an initial set of viewpoints of stakeholders in the initial requirements space created in Step3.

SMEs take general viewpoints templates and create an initial viewpoints model under categories 'direct' and 'indirect' viewpoints. This is a typical practice in general viewpoints oriented requirements engineering. Then, SMEs need to read and assess each requirement from the initial requirements space and identify corresponding viewpoints of each requirement as shown in Figure 4.9.

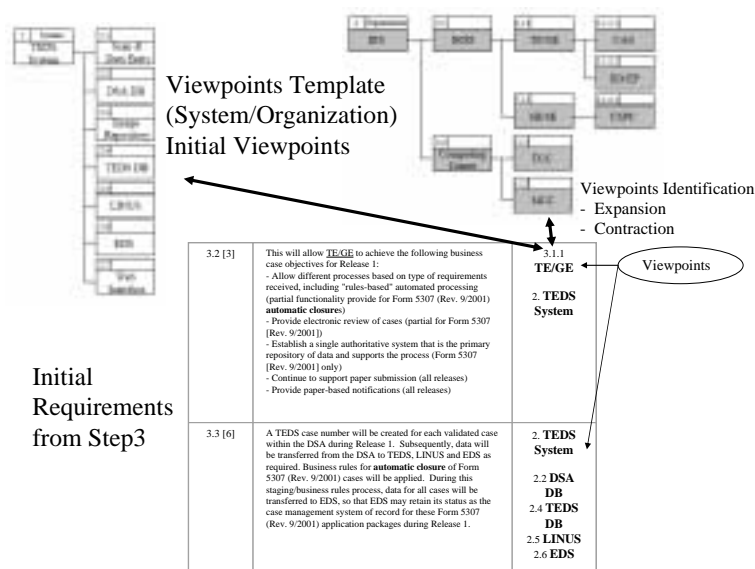


Figure 4.9: Step 4: Viewpoints Identification

This initially created viewpoints model expands whenever new viewpoints are identified and contracts whenever multiple viewpoints need to merge to a single viewpoint. The creation of viewpoints model is an iterative and incremental process.

### 4.5. Step5: Requirements category identification in enterprise model (EM)

SMEs define the requirements categories in the enterprise model that are appropriate for the project. SMEs need to read and assess each requirement from the initial requirements space and identify an appropriate requirements category of EM for each requirement as shown in Figure 4.10. As a result, appropriate allocation of requirements to the corresponding systems engineering process in EM can be achieved.

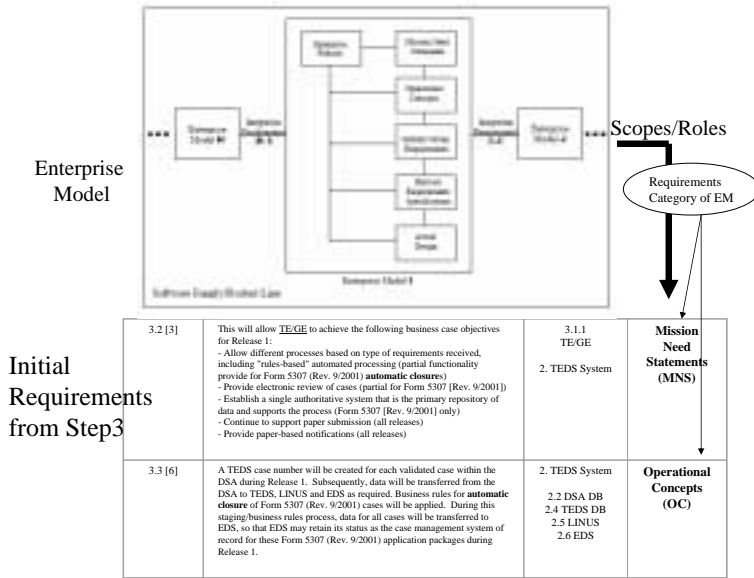


Figure 4.10: Step 5: EM Categorization

Each requirements category in EM (that now has its corresponding requirements in its category) is expanded to inherit and include the characteristics of a viewpoints model and creates a new indexing structure, which is the proxy viewpoints model.



#### **4.6. Step6 & Step7: Creation of proxy viewpoints model & Requirements analysis based on the defined properties of proxy viewpoints model**

At this stage, each requirement from the initial requirements space maintains memberships (or indexes) to corresponding viewpoints model and enterprise model. For instance, from Figure 4.11, requirement 3.2[3] belongs to viewpoints ‘TE/GE’ and ‘TEDs system’, which are under ‘Organization’ and ‘System’ viewpoints, respectively. Also requirement 3.2[3] belongs to the requirements category of ‘Mission Need Statements’ in EM. This indexing scheme can be rewritten formally,  $R_k(VP_i, EM_j)$  in which means the  $k$ th requirement  $R_k$  has a viewpoint  $VP_i$  and belongs to  $EM_j$  category.

SMEs create a proxy viewpoints model and its conceptual PVRD layout can be drawn as shown in Figure 4.11. The links between requirements represent ‘part-of’ relationships. Based on this, each requirement that is consistent to the roles and scopes of each requirements category in EM can be organized and managed throughout the viewpoints model.

Theoretically, the level of abstraction of requirements descriptions in EM should be adjusted and determined by the granularity of the viewpoints model. In other words, there should be a balance between the viewpoints model and enterprise model in terms of their level of abstraction. For example, if the level of granularity of viewpoints is very fine level, one may want to introduce a more detail engineering process in EM by adjusting the abstraction granularity of requirements categories of EM. However, this is very hard to achieve in a real project environment and no practical research solutions exist yet.

SMEs analyze the potential patterns of PVRD properties by assessing the indexes and associated links of requirements. The following factors should guide SMEs’ analysis process: 1) The defined PVRD properties [Lee03]: Checking any pattern of PVRD properties of requirements incompleteness, inconsistency, redundancy, ambiguity, relationship chain, and workflow process relationship chain; and 2) Use of domain and technical requirements knowledge in understanding and analysis of any potential patterns.

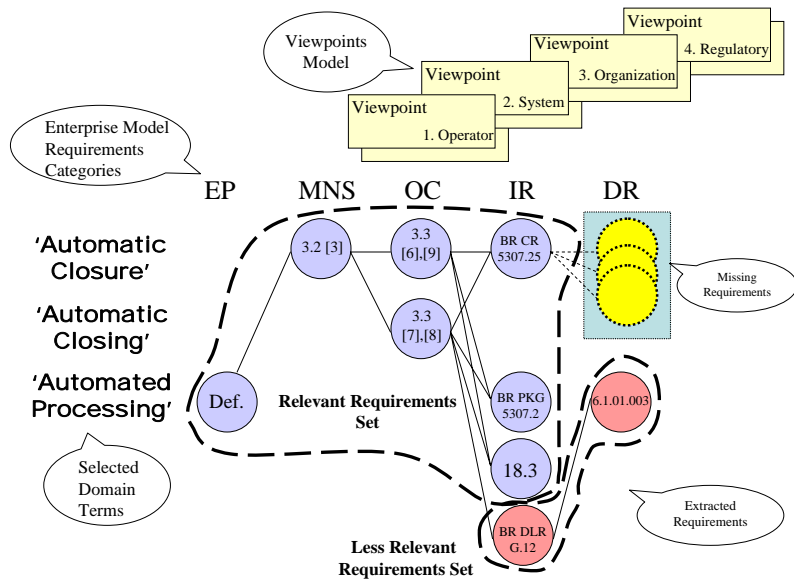


Figure 4.11: Steps 6 & 7: The PVRD Model with Missing Requirements Analysis

For example, in Figure 4.11, SMEs found a pattern of ‘requirements incompleteness’. More specifically, a set of system functional requirements under Derived Requirements (DR) category of EM is missing. Once any such a potential pattern is identified in the PVRD layout, SMEs continue to start the discovery process, which is the most important step but hard to achieve without having a supporting knowledge model, such as PVRD model.

#### 4.7. Step8: Discovery of missing requirements and other requirements defects of various types

Based on the analysis from Step7, SMEs can discover missing requirements, other requirements defects of various types, requirements relationship chains and workflow process relationship chains.

For ‘missing requirements’ discovery, SMEs can use ‘term expansion method’ to retrieve additional specific requirements of interest (i.e. potentially missing requirements). The rational of this method is to automatically generate a list of

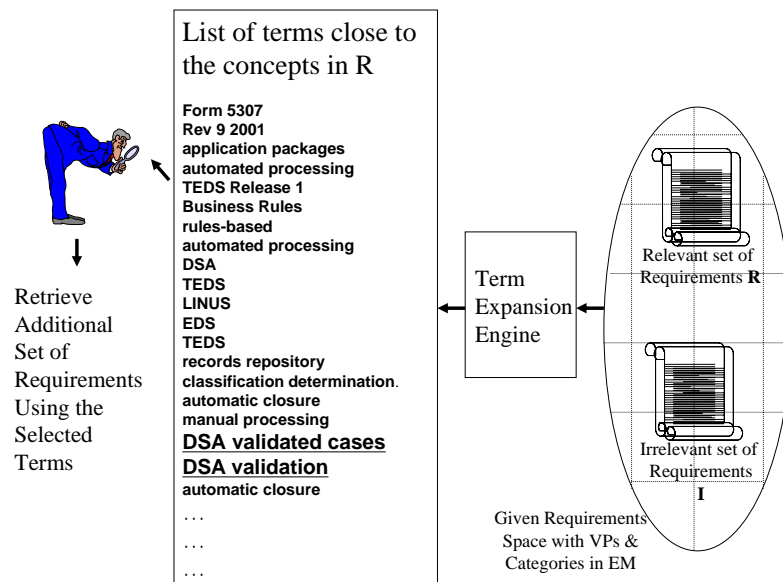


Figure 4.12: Step 8-1: Requirements Term Expansion

domain terms that can potentially be used as ‘queries’ to retrieve a set of additional requirements from the remaining requirements documents or external resources.

For example, as shown in Figure 4.11, SMEs need to categorize the initial set of requirements into two sets, ‘relevant’ and ‘less-relevant’ sets of requirements, to the target concept described in Steps 1&2. Then, SMEs take these two sets of requirements as an input to the term expansion method to generate a list of domain specific requirements terms that can be used in the consecutive requirements search process as shown in Figure 4.12.

SMEs need to assess the generated list of terms (with associated frequency-based weights) and make associations to the target requirements concept in the PVRD model. The goal is to find an additional set of requirements that were missed in the previous search because of the inconsistent use of terms in the requirements, or conceptual *knowledge gaps* between the set of domain terms (i.e. ‘**automatic closure**’) that were used in the creation of initial requirements space and the terms (i.e. ‘**DSA Case Validation**’) generated through the term expansion method. In other words, the conceptual links between these terms were missing.

Figure 4.13 shows an example of the results of missing requirements discovery. In this example, a set of requirements was missing because of the ‘language gap’ between the stakeholders’ language and the systems’ language in the requirements.

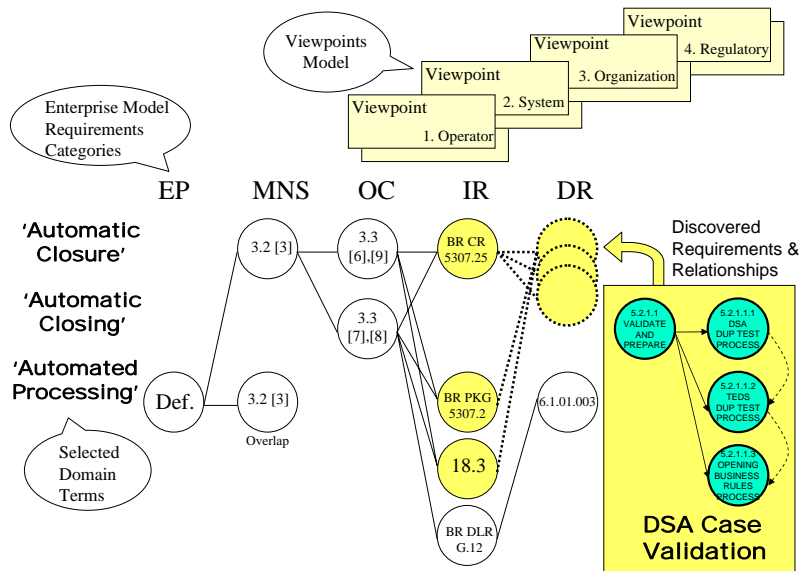


Figure 4.13: Step 8-2: Missing Requirements and Relationship Chain Discovery

SMEs can also discover requirements defects of other various types (i.e. inconsistency, redundancy, and ambiguity), relationship chain between requirements, and workflow process relationship chain, based on the defined PVRD model properties. The workflow process relationship chain (inside of the ‘DSA Case Validation’ box) in Figure 4.13 shows an example of such a relationship chain. The three sub-processes that compose the ‘DSA Case Validation’ process are discovered and represented with the data and control flows.

Figure 4.14 shows how to discover requirements defects of other various types. Based on the viewpoints identified from each requirement, SMEs can also identify a set of missing requirements and their viewpoints-to-be. SMEs need to assess requirements that are connected to each other (or requirements in the same EM category) in the adjacent EM category, to check whether there exist any defects of other various types.

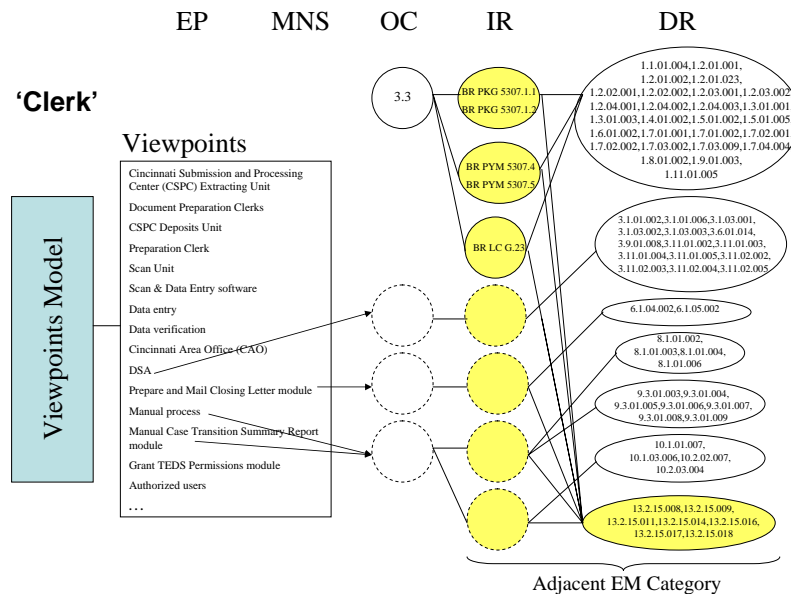


Figure 4.14: Step 8-3: Requirements Defects Discovery

From Figure 4.14, SMEs can identify a 'Many-to-One' requirements relationship that the seven sets of requirements (under Initial Requirements, including missing requirements) depend on a set of requirements under Derived Requirements, which is related to the 'security grant permission module' in the system. In fact, SMEs found a set of inconsistent requirements from this model.

Discovery of this type of requirements relationship and the understanding of its characteristics can provide SMEs more focused area to look for requirements defects.

It is important to note that this missing requirements discovery process in PVRD also concerns the 'requirements distance' problem [JDM01], which is recognized as a hard-to-solve problem in software requirements engineering that cannot be handled though a traditional inspection technique, by discovering missing requirements from far distant sets of requirements or different resources.

‘Stopping condition’ of the discovery process is closely related to the defined evaluation criteria such as metric (i.e. the significance of the discovery) and measures (i.e. whether the discovered requirements are defining, mandatory or optional requirements) in evaluating the results of the discovery [Lee03].

## 5. Conclusion and Future Work

Wieggers addresses the importance and difficulty of “missing requirements” as follows: “*Missing requirements are among the hardest errors to detect. They are invisible.*” in his practical discussion of inspecting requirements [Wie01, Wie03].

The PVRD methodology is unique in its systematic “integrated” architecture that facilitates both discovery and organization/management of natural language SRS documents in software systems resources (SSR). It transforms the static legacy status natural language SRS to active models such as proxy viewpoints models. These models discover not only ‘*discontinued knowledge*’ (i.e. missing requirements and relationships) but also create ‘*new knowledge*’ (i.e. interdependent link connections between requirements through viewpoints model and enterprise model) in SSR and make them available to all other aspects of software development environment.

For example, individual pieces of information finally become valuable when they establish “*links*” with each other from various aspects/dimensions based on a certain set of goals. Such dimensions (i.e. viewpoints and enterprise model), goals (i.e. target requirements concepts), patterns (i.e. interdependencies between requirements in the proxy viewpoints model) are all valuable assets that can be a basis for reusing and sharing by other software development activities such as software design, architecture, product line [BOS00, CSL<sup>+</sup>01, PMZ03] etc.

The flexible architecture of the PVRD also allows applying other techniques to the steps in the PVRD methodology to achieve various goals of natural language requirements engineering research. For example, by relating to the classical traceability problems, many of steps in the PVRD could be performed in a requirements management tool such as DOORS [DOO].

A part of future work includes a development of requirements engineering workbench that can capture the knowledge generated from various stages of the software process through: 1) Software implementation of the PVRD methodology in a proactive way (wizard type smart interface with the combination of automatic natural language requirements engineering technique for text summarization) that guides the SMEs and minimize their manual process in multi-dimensional link analysis; 2) Enhancement of the PVRD methodology by considering aspects from enterprise modeling, business modeling [NR03, Zac87] and their interdependencies to natural language SRS in SSR as ways of organizing and managing business processes and associated requirements; and 3) Bridging the results from the PVRD research to other aspects of software development.

Theoretical background of models and methods in the PVRD methodology, case study design and results from multiple (real) case studies carried out for the validation of the methodology are not presented in this paper but can be found in [Lee03, LR04a]. This paper is an extended version of the paper in [LR04b].

## References

- [AS99] A. Alderson and H. Shah. Viewpoints on legacy systems. *Communications of the ACM*, 42(3):115–116, March 1999.
- [BOS00] J. Bergey, L. O'Brien, and D. Smith. Mining existing assets for software product lines: Product line practice initiative. Technical Report CMU/SEI-2000-TN-008, CMU, Pittsburgh, PA, 2000.
- [Bue99] D. M. Buede. *The Engineering Design of Systems: Models and Methods*. Wiley Series in Systems Engineering. New York: Wiley, December 1999.
- [Coc77] W. G. Cochran. *Sampling Techniques*. John Wiley & Sons, New York, third edition, 1977.
- [CSL<sup>+</sup>01] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag. An industrial survey of requirements interdependencies in software product release planning. In *Proceedings of the Fifth IEEE Int'l Symposium on Requirements Engineering (RE '01)*. IEEE Press, 2001.
- [DOO] DOORS. Requirements management solution. Technical report, <http://www.regulatory.com/doors.htm>.
- [FKNG92] A. Finkelstein, B. Kramer, B. Nuseibeh, and M. Goedicke. Viewpoints: A framework for integrating multiple perspectives in system development. *Int'l. Journal of Software Engineering and Knowledge Engineering*, 2(1):31–58, 1992.
- [FO95] C. Faloutsos and D. Oard. A survey of information retrieval and filtering methods. Technical Report CS-TR-3514, University of Maryland, College Park, Maryland, 1995.
- [JDM01] L.L. Jilani, J. Desharnais, and A. Mili. Defining and applying measures of distance between specifications. *IEEE Transactions on Software Engineering*, 27(8), August 2001.
- [KA97] T.G. Kirner and J.C. Abib. Inspection of software requirements specification documents: A pilot study. In *Proceedings of the 15th Annual International Conference on Computer Documentation*, pages 161–171, Snowbird, UT, 1997.
- [KS96] G. Kotonya and I. Sommerville. Requirements engineering with viewpoints. *BCS/IEE Software Engineering Journal*, 11(1):5–18, 1996.
- [KS98] G. Kotonya and I. Sommerville. *Requirements Engineering: Processes and Techniques*. Worldwide Series in Computer Science. Wiley, New York, September 1998.
- [Lee03] S.W. Lee. *Proxy Viewpoints Model-based Requirements Discovery*. Phd dissertation, School of Information Technology and Engineering, George Mason University, Fairfax, VA, 2003.



- [LR04a] S.W. Lee and David C. Rine. Case study methodology designed research in software engineering methodology validation. In *Proceedings of the Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE '04)*, Banff, Alberta, Canada, June 2004.
- [LR04b] S.W. Lee and David C. Rine. Missing requirements and relationship discovery through proxy viewpoints model. In *Proceedings of the 19th annual ACM Symposium on Applied Computing (SAC 2004), Special Track on Software Engineering: Applications, Practices, and Tools*, Nicosia, Cyprus, March 2004.
- [Mer01] M. Merriman. Fiers: Categorization of missing requirements. Technical report, School of Information Technology and Engineering, George Mason University, Fairfax, VA, 2001.
- [Mul79] G.P. Mullery. A method for controlled requirements specifications. In *Proceedings of the 4th Int'l. Conference on Software Engineering*, pages 126–135, Munich, Germany, 1979. IEEE Computer Society.
- [NKF94] B. Nuseibeh, J. Kramer, and A. Finkelstein. A framework for expressing the relationships between multiple views in requirements specification. *IEEE Transactions on Software Engineering*, 20(10):760–773, October 1994.
- [NR03] S. Nurcan and C. Rolland. A multi-method for defining the organizational change. *Information and Software Technology, Elsevier Science*, 45:61–82, 2003.
- [PMZ03] P. Pavan, N.A.M. Maiden, and X. Zhu. Towards a systems engineering pattern language: Applying i\* model requirements-architecture patterns. In *Proceedings of the 2nd Int'l workshop Software Requirements to Architectures (STRAW) in ICSE '03*. IEEE Press, 2003.
- [PVB95] A.A. Porter, L.G. Votta, and V.R. Basili. Comparing detection methods for software requirements inspection: A replicated experiment. *IEEE Transactions on Software Engineering*, 21(6):563–575, 1995.
- [RGS00] P. Rayson, R. Garside, and P. Sawyer. Assisting requirements recovery from legacy documents. Technical Report CSEG/8/2000, Cooperative Systems Engineering Group, Computing Department, Lancaster University, United Kingdom, 2000.
- [RS77] D. Ross and K.E. Schoman. Structured analysis for requirements definition. *IEEE Transactions on Software Engineering*, 3(1):6–15, 1977.
- [Sal71] G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1971.
- [SFE96] G. Spanoudakis, A. Finkelstein, and W. Emmerich, editors. *Proceedings of the Int'l Workshop on Multiple Perspectives in Software Development (Viewpoints '96) in ACM Symposium on FSE*. ACM Press, San Francisco, CA, October 1996.
- [SS97] I. Sommerville and P. Sawyer. Viewpoints: Principles, problems and a practical approach to requirements engineering. *Annual Software Engineering*, 3:101–130, 1997.

- [Wie01] K.E. Wiegers. Inspecting requirements. *StickyMinds.com Original Column*, July 30 2001.
- [Wie03] K. E. Wiegers. *Software Requirements*. Microsoft Press, second edition, 2003.
- [XC00] J. Xu and B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, January 2000.
- [Yin94] Robert. K. Yin. *Case Study Research: Design and Methods*. Applied Social Research Methods Series Vol 5. Thousand Oaks: Sage Publications, second edition, 1994.
- [Zac87] J.A. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 26(3), 1987.

**Authors addresses:**

Seok Won Lee, Ph.D., Department of Software and Information Systems, College of Information Technology, The University of North Carolina at Charlotte, 9201 University City Boulevard, Charlotte, NC. 28223-0001, USA. seoklee@uncc.edu

David C. Rine, Ph.D., NASA NET Research Group, Department of Computer Science, Department of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, Fairfax, VA. 22030, USA. drine@gmu.edu