



**HAL**  
open science

## Knowledge-enriched Requirement Specification for One-of-a-kind Complex Systems

Svetan Ratchev, Kulwant S. Pawar, Esmond Urwin, Dirk Mueller

► **To cite this version:**

Svetan Ratchev, Kulwant S. Pawar, Esmond Urwin, Dirk Mueller. Knowledge-enriched Requirement Specification for One-of-a-kind Complex Systems. *Concurrent Engineering: Research and Applications*, 2005, 13 (3), pp.171-183. 10.1177/1063293X05055003 . hal-00571181

**HAL Id: hal-00571181**

**<https://hal.science/hal-00571181>**

Submitted on 1 Mar 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Knowledge-enriched Requirement Specification for One-of-a-kind Complex Systems

Svetan Ratchev,<sup>1,\*</sup> Kulwant S. Pawar,<sup>1</sup> Esmond Urwin<sup>1</sup> and Dirk Mueller<sup>2</sup>

<sup>1</sup>*School of Mechanical, Materials, Manufacturing Engineering and Management, University of Nottingham, UK*

<sup>2</sup>*Institute of Mechanical Engineering, Technical University of Clausthal, Germany*

**Abstract:** Requirement engineering (RE) process is becoming a key factor for the success of complex one-of-a-kind products. The RE process is commonly viewed as an early system engineering phase with a major bearing on response time, quality, and cost. This study reports on the knowledge acquisition and sharing for requirement engineering (KARE), approach for requirement specification of one-of-a-kind complex systems. The approach provides a generic view of key RE processes clustered into three groups of activities – requirements elicitation, analysis, and negotiation. The process is supported by a set of knowledge functions aimed at facilitating the requirement engineers in matching customer requirements to product characteristics. At the analysis stage, the customer requirements are transformed into product requirements, which can be compared to existing company knowledge, for example, previous products, technology platforms, and production capabilities. The specified product requirements are then interactively evaluated and negotiated against customer and supplier performance indicators.

**Key Words:** requirement engineering, knowledge management, system engineering.

### 1. Introduction

Requirement engineering (RE) has evolved as a key stage in the overall system engineering process [1]. The increasing product complexity, market place globalization, and the changes in product life cycles have underlined the need for increased reuse of components, information, and knowledge across projects to deliver efficient and cost-effective product solutions. This is particularly valid in large one-of-a-kind projects requiring significant effort at the initial product specification stages, where a large proportion of the product cost is committed. In fact, in developing products with high complexity in industries, such as aerospace, it is widely accepted that almost 60% of the product cost is allocated during the first 5% of the product development cycle. The success of the requirement specification in new design projects largely depends on an accurate match between customer requirements and company product and process knowledge. The successful understanding of the user requirements and their transformation into clear product requirements therefore becomes a critical element within the overall realization of a successful product [2].

Poor understanding of the RE process and inaccurate assumptions made during the elicitation of user requirements can have significant negative implications on the design and manufacture of the product affecting quality, lead time, and cost [3].

Customers usually encode the knowledge related to their requirements of needs, into text, drawings, or verbal messages, i.e., into information. The conversion of this knowledge into explicit information stems from the transformation of tacit knowledge [4], i.e., personal knowledge that is implicit and cannot easily be communicated. The main disadvantage of such form of knowledge is that it avoids critical discussion and there are doubts about its level of objectivity [5,6]. This, therefore, not only makes the management of the RE process difficult, but on occasions can also obstruct critical product and process considerations and hence, the development of the knowledge on RE. The complexity of the customer–supplier interaction is further exacerbated by intercultural differences between business partners. Requirement defects can thus be created when a supplier guesses as to what is implied by an incomplete or ambiguous requirement statement that often loses part of the explicit meaning through an incorrect translation of the customer's implicit knowledge [7,8]. Despite the recent developments in both RE and knowledge management, there is still a lack of generic methods and models that could support knowledge acquisition and sharing for RE in a one-of-a-kind environment.

\*Author to whom correspondence should be addressed.  
E-mail: svetan.ratchev@nottingham.ac.uk

According to Hooks [9], customer statements often produced without sufficient product-related information, are generally initial requirements that are either ill-defined or are goals, objectives, and design statements rather than actual requirements [10]. Such statements are subject to changes partly due to the lack of vision of the new product to be developed and possible technological changes that might occur during the product development process. These potential changes need a requirement change management process, which must be performed throughout each phase of the requirement specification process. It is, therefore, of interest to analyze specifically the contents and links between each of the different phases. In this analysis, the complexity of the system to be developed and its requirements, the available knowledge within the application domain, the level of skills within the project team (including the customer), and organizational culture must be taken into account.

The requirement specification process for a one-of-a-kind system can also be very specific [10]. The understanding of its content and structure requires the monitoring of the activities, which have been carried out in real life. The modeling of these activities has become a major research area because of the complexity of integrating several application domains that involve different groups of individuals with different problems, solutions, and interests at stake. These activities should be intensively supported by knowledge, which includes data about the system or product and organizations (market place, legislation, technological data, standards, etc.). This data is collected and used differently at widely dispersed locations. Different rules are often used to turn this data into knowledge, resulting in a lack of consensus between knowledge experts [4]. Therefore, the mechanisms for acquiring and distributing knowledge within multidimensional domain applications are essential for providing organizations with flexible ways of bridging gaps between requirements and knowledge engineering.

The current research in engineering requirements has been summarized in [11]. Although research in these fields has been performed for general design projects, these cover specific areas for each specific case. No study has been reported that encapsulates all the areas with application to a single field, i.e., using prescriptive methods to provide automated or semiautomated design together with descriptive means.

Furthermore, the differences between the application of RE in software and engineering disciplines are outlined where engineering design aims to 'arrive at a complete, concise and correct description of the design need, expressed in natural language.' This involves the formalization of design requirements into a structured methodology, whereas software requirements are more flexible and do not need to be processed in this way.

Bray [12] highlights the need for RE and provides a basic definition of the principles. Furthermore, the importance of design requirements is acknowledged by Stadzisz and Henrioud [13] who describe how requirements are filtered throughout the design cycle as products and product families are created to serve the wants and needs of customers.

Research in this field is generally fragmented with applications developed to satisfy specific needs. For example, the NIBA project [14] recognizes the importance of requirements and the need for better analysis of user requirements. As user requirements are usually elicited in natural language, a linguistically based method for analyzing these requirements is thought to be beneficial. This approach decomposes the syntax of requirement statements and analyzes sentences to ensure that they are syntactically correct. Although this provides a useful basis for completeness and consistency checking, no further work has been reported that uses the output from this method.

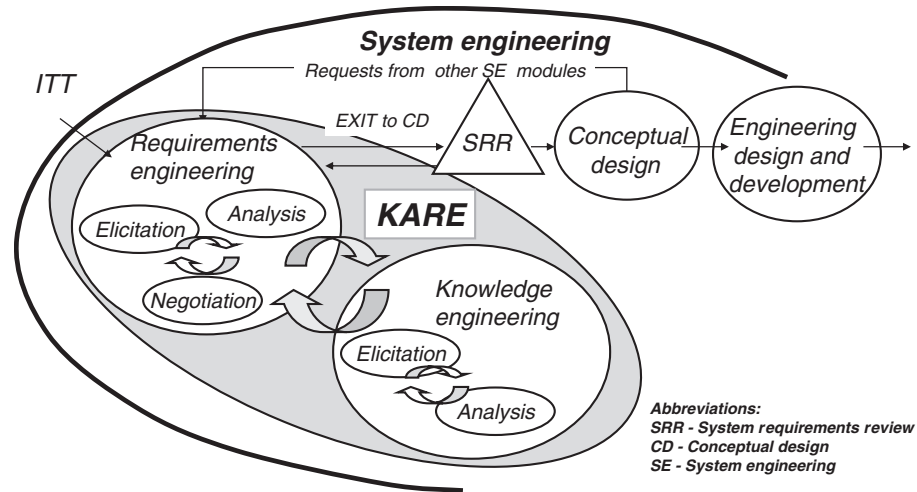
Despite the recent developments, there is still a lack of transparency and consistent definition of the activities in RE. There is also a lack of structured methods for capturing relevant enterprise knowledge and deploying it in support of decision making for requirement specification.

The article reports on a new knowledge-'enriched' RE approach for one-of-a-kind complex products. The methodology is based on the development of a generic knowledge model that supports the key RE steps of requirement elicitation, analysis, and negotiation.

## 2. Overview of the KARE Methodology

The knowledge acquisition and sharing for requirement engineering (KARE) [15,16] methodology is aimed at supporting the RE process in a one-of-a-kind environment. The RE process (Figure 1) starts with an invitation to tender (ITT) and finishes with a requirements baseline. KARE provides an interface for the system requirement review (SRR), which ensures that system requirements are sufficient to meet mission objectives, the expected performance, and that cost figures of merit are realistic. The approved requirements at the SRR are then passed on to the conceptual design (CD) stage.

The KARE approach has two main integrated components a requirement engineering (RE) process and a knowledge engineering (KE) process [12]. The KARE RE process comprises a three-step process: (1) requirements elicitation, (2) requirements analysis, and (3) requirements negotiation. The model assumes that the customers' ability to articulate their needs varies and accordingly, defines the elicitation of customer

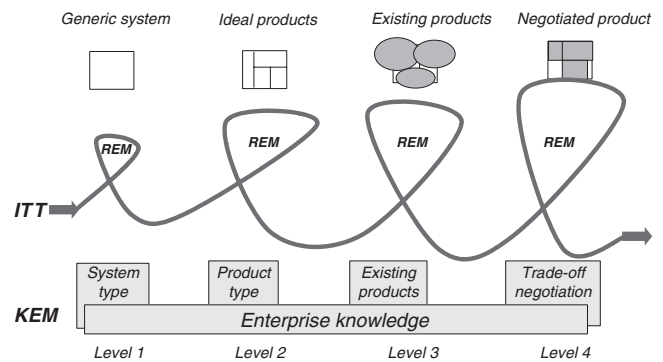


**Figure 1.** The KARE knowledge-enriched requirement engineering approach.

requirements as the first step in RE. Requirements elicitation is a process of discovering system requirements through consultation with stakeholders, from system documents, domain knowledge, or any other means of information. Requirements elicitation and analysis are closely linked processes. As requirements are discovered during elicitation, some analysis is inevitably carried out [7]. The requirements analysis investigates the requirements and then different stakeholders confer to decide on which requirements are to be accepted. The RE processes establish an iterative sequence, sometimes referred to as stages [17]. The KE module supports this sequence at all the stages.

The customer and the supplier confer requirements during the requirements negotiation. Both the parties may agree on top-level requirements. Additional requirements may be derived according to customer needs; requirements may be updated and then, re-enter the requirements elicitation process. The customer and the supplier negotiate the requirements supported by the requirement request engine. Every negotiation potentially relaunches the requirements elicitation process until a baseline is defined.

The RE process is developed in four stages – (1) the identification of system type, (2) identification of product range, (3) comparison with products, and (4) trade-off studies. The four stages of the process are to a great extent arbitrary and whether these are applied depends naturally on product complexity and on the company-specific practices. However, the requirements elicitation, analysis, and negotiation activities are performed at each stage resulting in an incremental requirements update in accordance with customer–supplier negotiations. Figure 2 illustrates the major phases of this process. Here, the emphasis changes from capturing customer needs at the first refinement iteration to consolidating requirements at the following refinement phases. Finally, prior to specifying the baseline definition, the suitability analysis



**Figure 2.** KARE requirement definition stages.

is conducted. The purpose of this phase is to specify suitable products with a level of trade-off in defining the particular product features. Such solutions are the basis for the selection of the product, which defines the baseline for the follow-up conceptual design.

Each of the RE processes are supported by knowledge functions. The underlying component of the KARE system that supports and enriches the RE process is the KE process module (refer Figure 1). KE supports the RE process through eliciting, analyzing, and answering support requests. A second objective of the KE module is the transformation of implicit knowledge into explicit knowledge through conceptual modeling. The KE process features two types of activities – knowledge elicitation and knowledge analysis. In the KARE context, the knowledge elicitation is defined as a process of collection, capture, and formalization of knowledge. The knowledge content is investigated in detail in the knowledge analysis process to validate and quantify the knowledge before the knowledge repository is added to and/or updated. The knowledge elicitation and knowledge analysis processes constitute a recursive loop that supports RE at any stage throughout the RE process.

### 3. KARE Requirement Engineering Process

The RE activities address the broad and diverse domain of customer needs. Customers may have very detailed concepts of their needs, or in some cases have only an idea about a product. It is very likely that the majority of the requests for specification of customer requirements of needs would rank somewhere between these two extremes. As described earlier, the KARE generic RE process model is based on three iterative knowledge-supported decision making steps – (1) requirements elicitation, (2) requirements analysis, and (3) requirements negotiation. Through a number of iterations, the supplier and the customer negotiate the requirements, which result in requirement updates until a product requirement baseline is defined [1].

#### 3.1 Requirements Elicitation

The goal of the requirements elicitation process is to collect and capture the relevant customer needs. These are then formatted, i.e., checked for ambiguous words, adequate grammar rules are applied, and the customer and supplier terminologies are related using a glossary. The process comprises three steps – requirement collection, capture, and formatting (Figure 3). The requirement collection is triggered by ITT with a set of user requirements of need, or further requests for elicitation resulting from downstream system engineering

activities. The requirement collection requires customer–supplier interactions and is technically limited to the collection of all the information that contains possible requirements. The input includes documents, files, drawings and the like, which are input to the requirements capture. Furthermore, it may contain requirements, which the subsequent requirements formatting step has considered as incomprehensible and as such have been referred back to the requirements capture stage. Requirements collection may require interviews with customer representatives, which call for considerable skills and can have a considerable impact on the customer–supplier relationship [7].

The requirements capture step segregates the requirements from unrelated information. This step is required to determine which of the collected customer requirements, are indeed the requirements and which are not. Within the KARE environment, the requirement engineer can request knowledge messages to support the decision making in the form of requirement identification rules.

A KARE requirements dictionary supports the requirements formatting step. The dictionary provides requirements terms and meanings for the formatting process, thus allowing the customer and the supplier terminologies to be related. Moreover, a list of inappropriate terms and knowledge of requirement grammar is provided to support the process. The formatting step is facilitated by requirement templates, which provide the format in which the customer

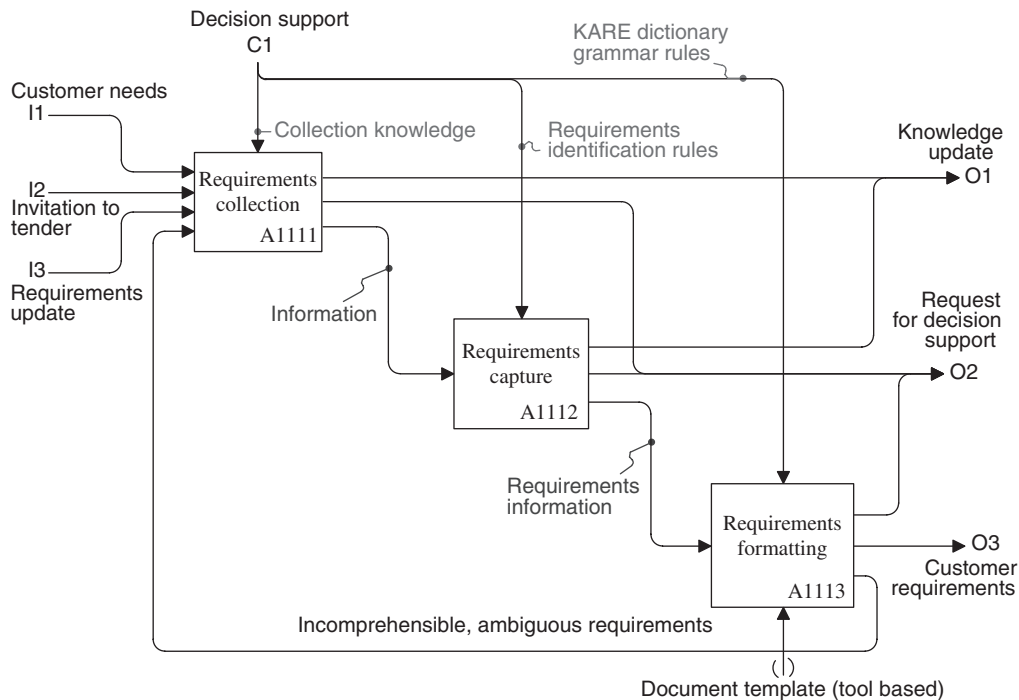


Figure 3. Requirements elicitation process – an overview.

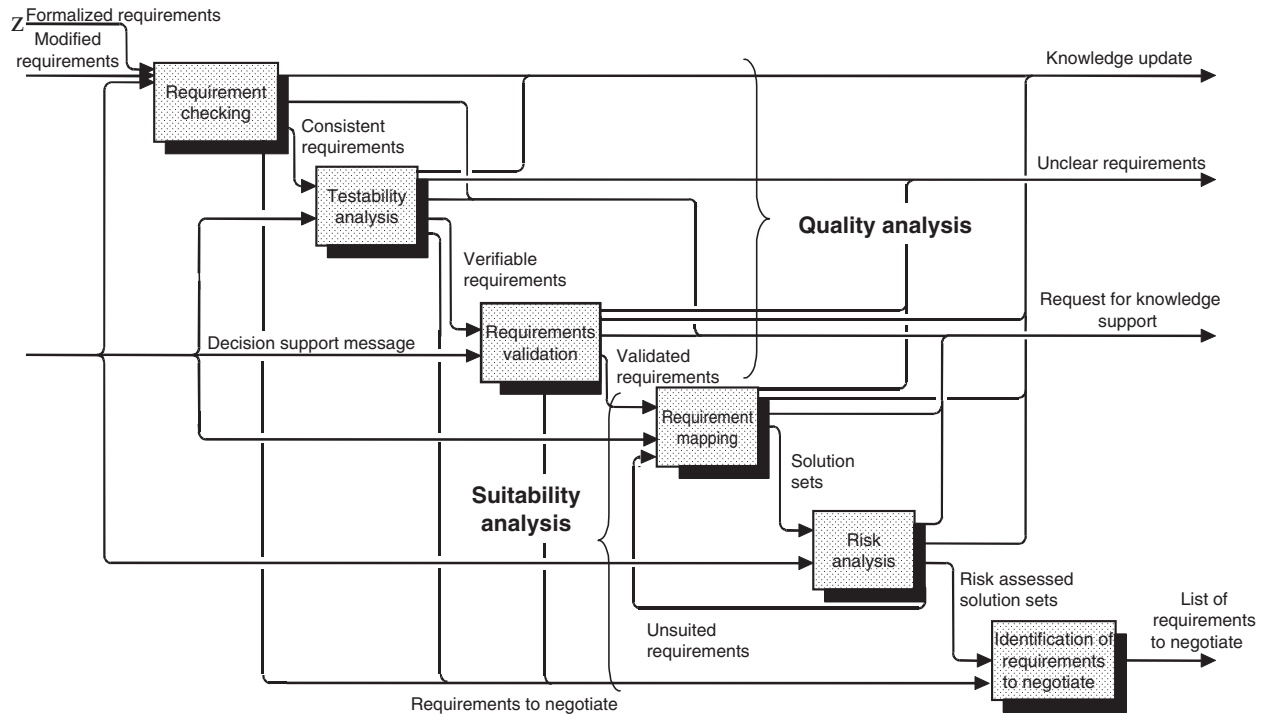


Figure 4. Requirements analysis process – an overview.

requirements are referred to in subsequent processes. Ambiguous and incomprehensible requirements are referred back to requirements collection stage to seek further information or verify specific statements.

### 3.2 Requirements Analysis

The goal of the requirements analysis stage is to generate the reference requirements that would form the product baseline. The requirements analysis is a highly recursive and knowledge-intensive process (Figure 4). It consists of three major steps – (1) requirements quality analysis, (2) suitability analysis, and (3) preparation for requirement negotiation.

The requirements quality analysis investigates the quality of the requirements themselves, through requirements checking, testability analysis, and requirements validation. The goal of these processes is the production of requirements that are consistent, valid in terms of feasibility and necessity, and are quantifiable and verifiable.

The suitability analysis features three steps – requirements mapping, risk analysis, and identification of requirements to negotiate. It starts with the validated requirements being mapped to existing functional and nonfunctional product requirements. For requirements that cannot be mapped to existing products, new products must be developed. Therefore, requirements mapping relies on knowledge about products that have

been produced. A trade-off analysis is used to consider the effects of requirement alterations and changes. Some requirements are usually mandatory (e.g., legal obligations), while others leave scope for alterations or may possibly be changed altogether. Considerations depend upon technical importance with regard to the customer needs, technical difficulty, environmental operating considerations, costs, etc. [18].

The risk analysis process allocates and assesses the risk to product components and refers unsuited components back to the mapping process. To support the data analysis, a matrix showing components and company-specific risk criteria is generated and used to evaluate the risk for every component on a predefined scale.

The last step is the identification of requirements to negotiate, where decisions about the requirements of the potential solution sets need to be negotiated with the customer.

## 4. KARE Requirement Engineering Knowledge Model

The KARE RE process is supported by a knowledge model which has three distinctive levels [19] – domain knowledge, inference knowledge, and task knowledge. This makes explicit all the rules, reasoning processes, and knowledge within the processes. Keeping this in mind, a measure of control and standardization can be

applied throughout the entire RE knowledge modeling process.

The knowledge model categorizes knowledge into two main areas [19]:

- *Control knowledge* which defines both the content and the structure of task- and inference-specific knowledge in a procedural form. It consists of two knowledge categories:
  - *Task knowledge* which describes how to decompose the top-level reasoning task and how to impose control on this decomposition.
  - *Inference knowledge* which describes how to use domain knowledge in elementary reasoning steps (inferences).
- *Domain knowledge* which is static knowledge and consists of the concepts, relations, and facts that are needed to reason about a certain application domain. It defines both the content and the structure of domain-specific knowledge in declarative form.

The KARE approach differentiates between declarative and procedural RE knowledge. The declarative knowledge, for instance, is the knowledge about products and requirements, which is processed by the procedural knowledge. The procedural knowledge is defined as the knowledge about how to transform requirements. KARE differentiates between procedural

knowledge, for which explicit rules can be identified as KARE knowledge and that which eludes rule-based descriptions. The former, the KARE procedural knowledge, is coded into automatic processes and incorporated into the KARE workbench. The latter is relevant insofar that it defines the processes where human decision making is essential.

The KARE approach in managing and maintaining the RE knowledge is based on four tasks – knowledge elicitation, analysis, mapping, and sharing.

#### 4.1 Knowledge Elicitation

The knowledge elicitation function provides declarative knowledge for the RE process. The acquisition of such knowledge is a constant process throughout each of the RE processes. From each RE process, a number of requirements are elicited, which are collected as a part of the knowledge acquisition process (Figure 5).

This potential new knowledge is then captured with the aid of a knowledge engineer, which is then checked against the current knowledge base as to whether or not it already exists and is utilized. Potential new knowledge that is not present proceeds to the knowledge formalization process, where it is formatted according to predefined forms. Once the three knowledge elicitation inferences have been completed, the potential new knowledge is passed to the knowledge analysis process for validation and verification.

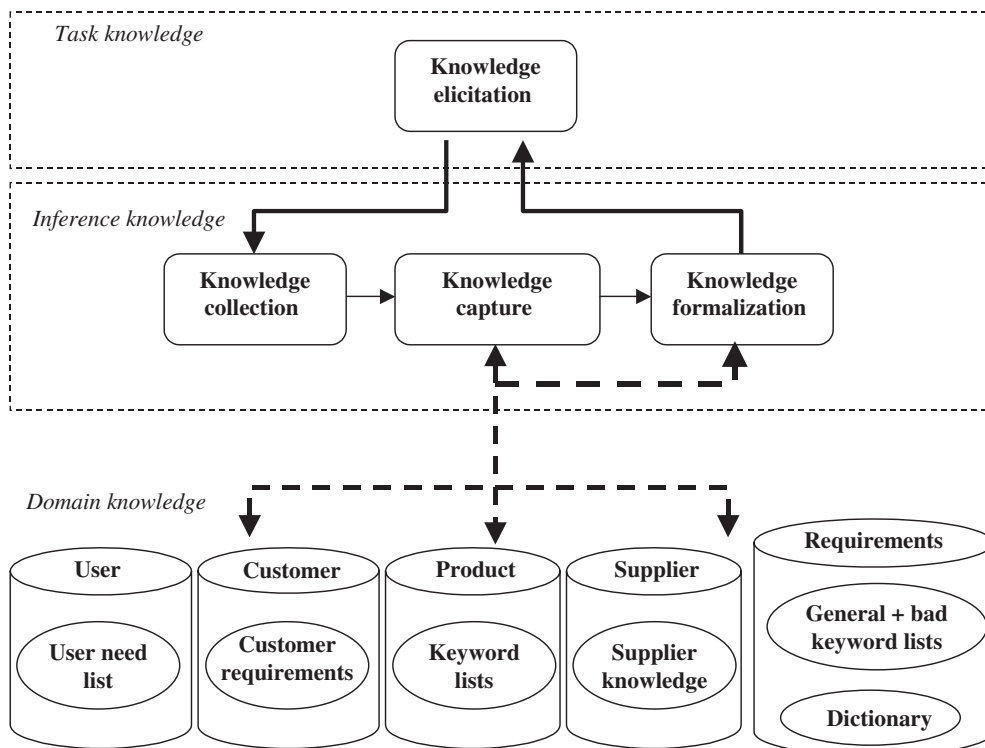


Figure 5. Knowledge elicitation model.



### 4.2 Knowledge Analysis

The knowledge analysis task seeks to verify and check new knowledge for completeness and consistency. It is supported by two inferences for knowledge review and knowledge negotiation (Figure 6). The knowledge review inference receives the formalized knowledge and checks its the completeness and consistency. It applies the declarative knowledge stored in the knowledge base, for example for the completeness of the new knowledge, the knowledge formatting rules are accessed from within the relevant domain (Figure 6). These are applied against the item in question and a decision is made as to whether or not it is complete. If complete, the item proceeds to the negotiation phase, if not, input will be requested from a knowledge engineer to complete the item. Once reviewed, the knowledge is passed to the knowledge negotiation inference. The validity of the knowledge is checked using validity rules and the input of a knowledge engineer. Once both the inferences have completed the necessary iterations, the task of knowledge analysis is to update the knowledge base with the new knowledge.

### 4.3 Knowledge Mapping

For the knowledge base to be a useful and an intuitive item to use, the knowledge must be logically related to the physical world, i.e., RE. New knowledge

will have meaning attached to it, but can be considered as stand-alone items with no relationships or links to any of the RE knowledge. Therefore, some mapping of the knowledge must take place to add the relationships and links between the requirements knowledge and other areas of the knowledge base. The knowledge components are linked to their process applications. For instance, the customer knowledge is mainly applied to customer requirement elicitation, whilst requirement knowledge is applied throughout the RE process.

The secondary function of the task is the characterization of knowledge. The process starts with the identification of the knowledge utilized for each activity and breaking each activity into its constituting components. These are then classified as either declarative or procedural knowledge. The procedural knowledge is classified according to either rule-based behavior (RBB) or knowledge-based behavior (KBB). Table 1 provides an example of the knowledge classification and shows the functional mapping of requirements, which is a subset of requirements mapping activity.

Functional mapping requires both declarative and procedural knowledge. The explicit declarative knowledge is applied to requirement characteristics and product properties. Procedural knowledge is encoded into RBB provided that the rules and algorithms are explicit or can be elicited. When the need for KBB is highlighted for example, when a new product must be defined,

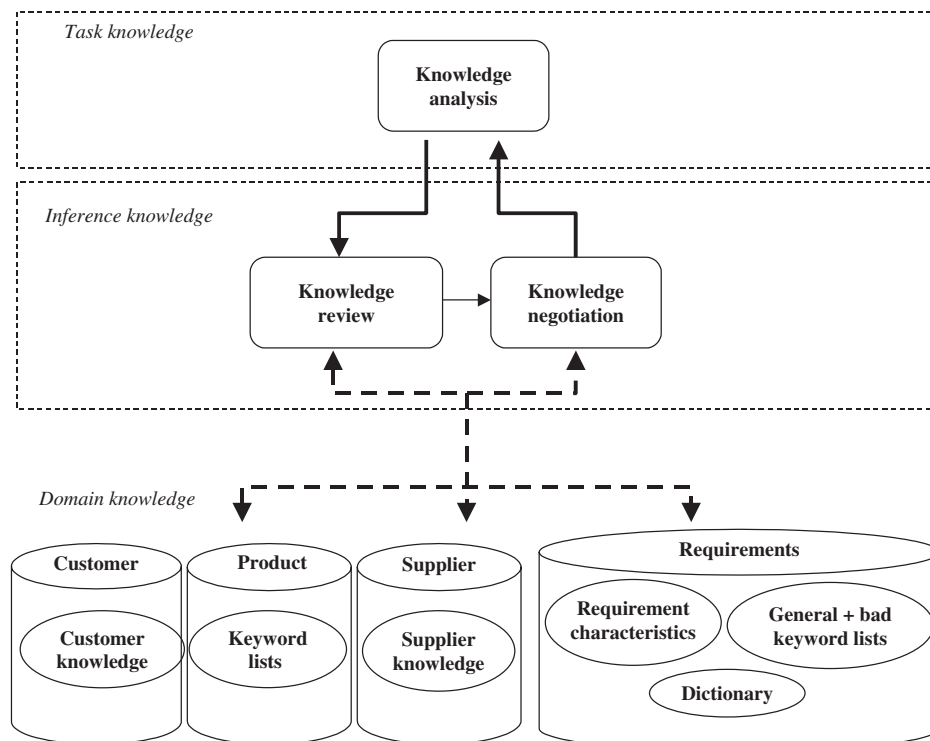


Figure 6. Knowledge analysis model.



the input and control of a human is needed to complete the complex procedural knowledge process (Table 2).

#### 4.4 KARE Knowledge Domains

The KARE knowledge base is organized into five different top-level knowledge domains (Figure 7). Each of the knowledge domains comprises different entities, for example the product domain, is comprised of artifact classes. Such artifact classes are likely to be company specific.

Figure 8 shows an example, which differentiates between system types (e.g., weapon system, surveillance system), the product ranges of a system type, existing and new products based on technology that will be developed and new technology platforms for new product ranges.

The product domain is linked to the supplier domain. The supplier knowledge domain features a list of suppliers with their artifacts and artifact properties. In addition, the domain includes knowledge about competitors and competing artifacts and attributes, such as price and delivery data, which are used to evaluate the market position of the supplier during a product invitation to tender/bid phase. The supplier terminology describes declarative company knowledge about requirements expressions and terminology, for example,

ambiguous wordings. This knowledge can be linked to customer knowledge about customer terminology and other background knowledge about the customer (e.g., preferences, cultural issues).

Overall, the RE knowledge content tends to be very company specific and dependent upon the business processes a company deploys for RE. Therefore, the RE processes as well as their knowledge support processes have to be configured to meet company-specific characteristics. The KARE methodology provides generic processes and procedural knowledge, which are supported by different knowledge domains and need to

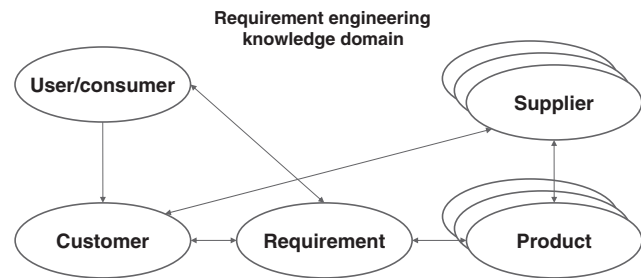


Figure 7. KARE knowledge domains.

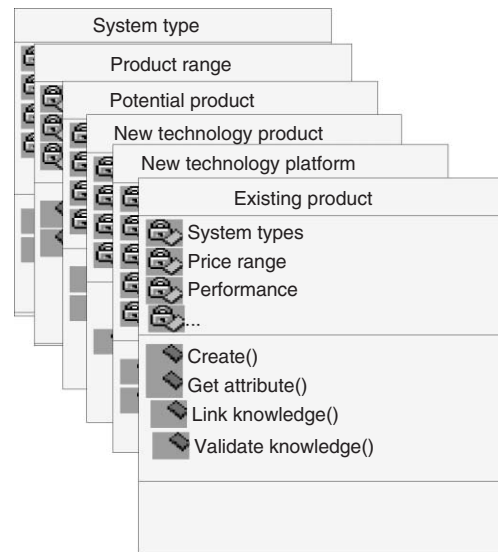


Figure 8. Product knowledge categories.

Table 1. Knowledge classification.

Activity	Knowledge	Type	
Functional mapping	Requirement characteristics	Declarative	
	Product properties	Declarative	
	Matching of properties and characteristics	Procedural	RBB
	Define new products	Procedural	KBB
	Define new properties of new products	Procedural	KBB
	Similarity threshold	Declarative	
	Discrimination threshold	Declarative	
	Define (new) similarity threshold	Procedural	KBB
	Define (new) discrimination threshold	Procedural	KBB

Table 2. Linking knowledge to domains.

Process	Knowledge	Type	Domain
Functional mapping	Requirement characteristics	Declarative	Requirements knowledge
	Product properties	Declarative	Product knowledge
	Matching of properties and characteristics	Procedural	Supplier knowledge
	Define new products	Procedural	Coded process
	Define new properties of new products	Procedural	Human control
		Procedural	Human control

be customized to capture the company- and product-specific knowledge.

#### 4.5 Knowledge Sharing

The knowledge sharing function in the KARE approach is focused on the RE process and is supported by the knowledge acquisition, analysis, and mapping processes. The knowledge acquisition and analysis processes consist of rule-based procedural knowledge, which is coded into the KARE workbench.

The knowledge sharing activity provides knowledge support for the RE processes. During the operation of these processes, requests for knowledge are sent to the knowledge engine. It responds by supplying the relevant knowledge based upon the request wherever possible. For example, after the capture of a customer's requirements, they must then be identified and formatted to standardize their representation. The knowledge engine provides support for the process by supplying requirement terms, meanings, grammar rules, and a list of bad words. These are applied to each of the requirements to allot it a relative identity and meaning. The applicable customer document templates are then supplied to the formatting process to enable the correct representation of the identified requirements.

### 5. Contributions to the Development of the System Engineering Data Exchange Standard Step AP233

The requirement model developed within the KARE approach [20] was adapted and incorporated into the ongoing work on developing a systems engineering exchange standard Application Protocol 233 (AP233) at ISO level. The AP233 model features a model-based representation of requirements, which allows the application of knowledge functions. A major problem today is that most requirements are specified in natural language.

The systems engineering AP 233 aims at providing support for system design processes. This application protocol defines the context, scope, and information requirements for various developmental stages during the design of a system and specifies the integrated resources necessary to satisfy these requirements. This application protocol shall be applicable to any type of system including complex aerospace systems, such as launch vehicles, aircraft, and their subsystems, such as avionics, aeroengines, or aerostructures. In general, these systems cannot be simply considered as a sum of single components, the point of view taken must be one of their integrated behavior, which in most cases is real-time dependent and has to be defined, validated, and verified at early product specification stages.

The following categories are within the scope of the systems engineering application protocol [20]:

- products with conformity to the concept of a system;
- system definition data and configuration control data pertaining to the design and the validation phases of a system's development;
- requirements;
- functional analysis data including functional behavior specifications;
- physical architecture and synthesis data providing a high level view on the system under specification;
- elements that are used to represent and trace requirements and the allocation of functions.

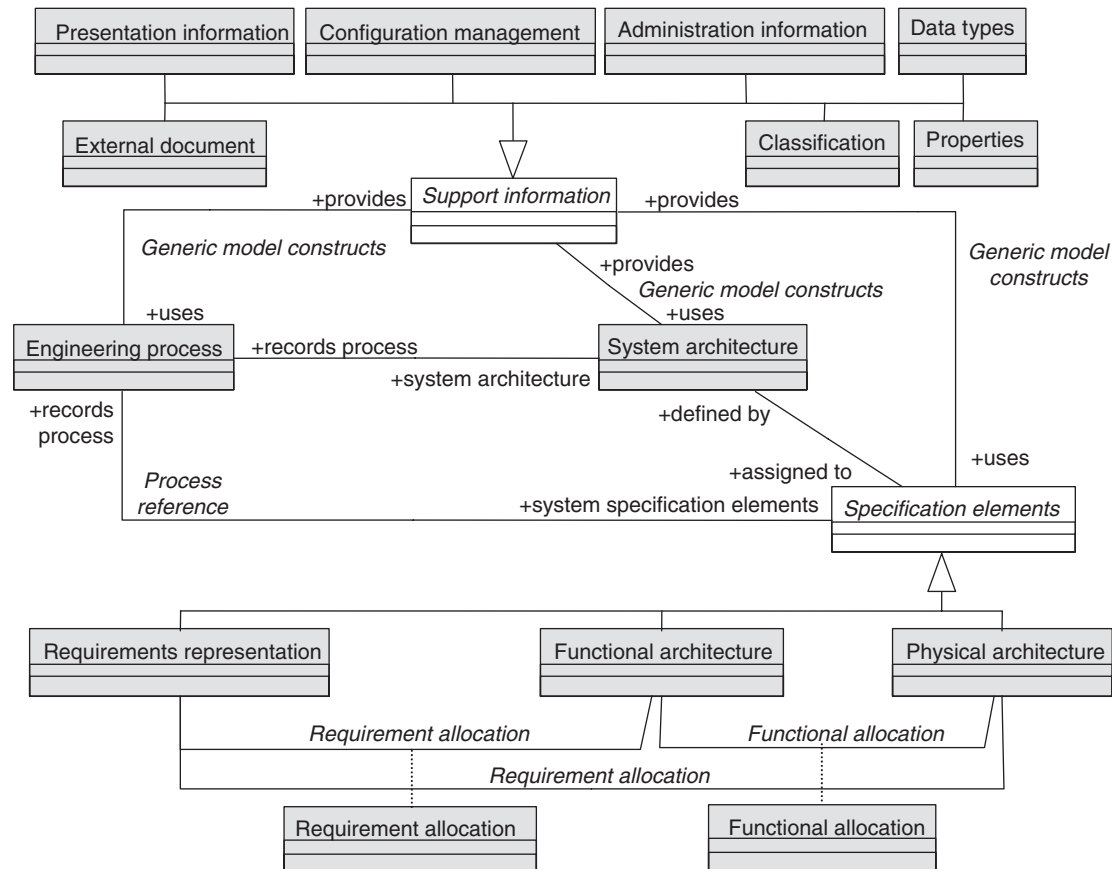
The ISO 10303-AP233 systems engineering application protocol consists of the following modules (Figure 9): system architecture; requirements; functional design; behavioral design; configuration management; physical design/architecture; properties; graphics; data types.

The KARE requirement model that was built within the AP233 has taken into account the functional requirements specification and input from projects, such as SEDRES, QCIM PM, and AP233 working group [20]. The KARE formalized requirements model provides part of the information that is necessary to enable the capabilities defined by the KARE knowledge model and by methodology. Since the AP233 EXPRESS model also defines relationships and constraints, some parts of the model can be considered as knowledge entities. This is due to the fact that the relationships and constraints also imply indirectly the operations or functions applied on them. According to the KARE methodology, all entities of the requirements model represent the knowledge objects, while relationships and constraints indirectly define some parts of the object types. Further research is currently undertaken in extending the KARE requirement model to include knowledge functions that embed the KARE RE knowledge model.

The interfaces developed in the RE data server enables the knowledge module to access the entities that are stored and managed within the KARE RE module.

### 6. System Implementation

The KARE knowledge model and methodology have been implemented in a prototype software 'workbench' aimed at facilitating the RE activities performed by system engineers. The KARE 'workbench' utilizes the newly proposed ISO STEP application protocol AP233 [21] for data modeling and exchange. It has been developed as a plug-in environment to the system engineering tool DOORS<sup>TM</sup> [22]. It utilizes two proprietary packages, a requirement formalization tool



**Figure 9.** The information model AP233 in a conceptual view.

Demanda II [20] and a knowledge management tool WISDOM<sup>TM</sup> [23] that have been integrated to deliver the key features of the KARE approach (Figure 10). The utilization of the STEP file format allows the seamless transfer of information between DOORS<sup>TM</sup>, WISDOM<sup>TM</sup>, and Demanda II whilst requirements are being engineered.

Figure 11 illustrates the menu list of the integrated KARE environment within the system engineering tool DOORS<sup>TM</sup>. It shows the currently implemented elicitation, analysis, and data exchange modules, which are supported by knowledge functions.

The KARE model, methodology, and prototype ‘workbench’ have been validated through a number of industrial cases of collaboration between the customer and the supplier on defining a ‘baseline’ system specification for complex aerospace systems. Three types of RE activities have been supported:

- *Supplier tasks* These include definition of company organization, level of knowledge application by the project team, role of RE, and use of related tools, methods for system architectural design;
- *Customer–supplier exchange tasks* These include elicitation of high-level requirements, refinement

of requirements, suitability analysis, and baseline definition;

- *Requirement and knowledge interface* This includes determination of product range, system type, domain knowledge updates, definition of system raw price, delivery schedule, and missing/incomplete requirements.

Although no formal and complete benchmarking to previous products was possible due to the complexity and the uniqueness of the systems and the nature of the industry, the pilot applications presented an effective way of testing different elements of the proposed methodology. The results from the industrial validation of the KARE prototype ‘workbench’ have shown significant improvements in terms of accuracy and consistency of requirements specification and reduced overall lead time. The advantages of using the prototype KARE ‘workbench’ were specifically evident in cases of identifying product types and suggesting key product requirements not specified by the customer. The KARE ‘workbench’ performed less efficiently in cases of definition of validation requirements due to the limited size of the prototype knowledge base and required expert-based decisions.

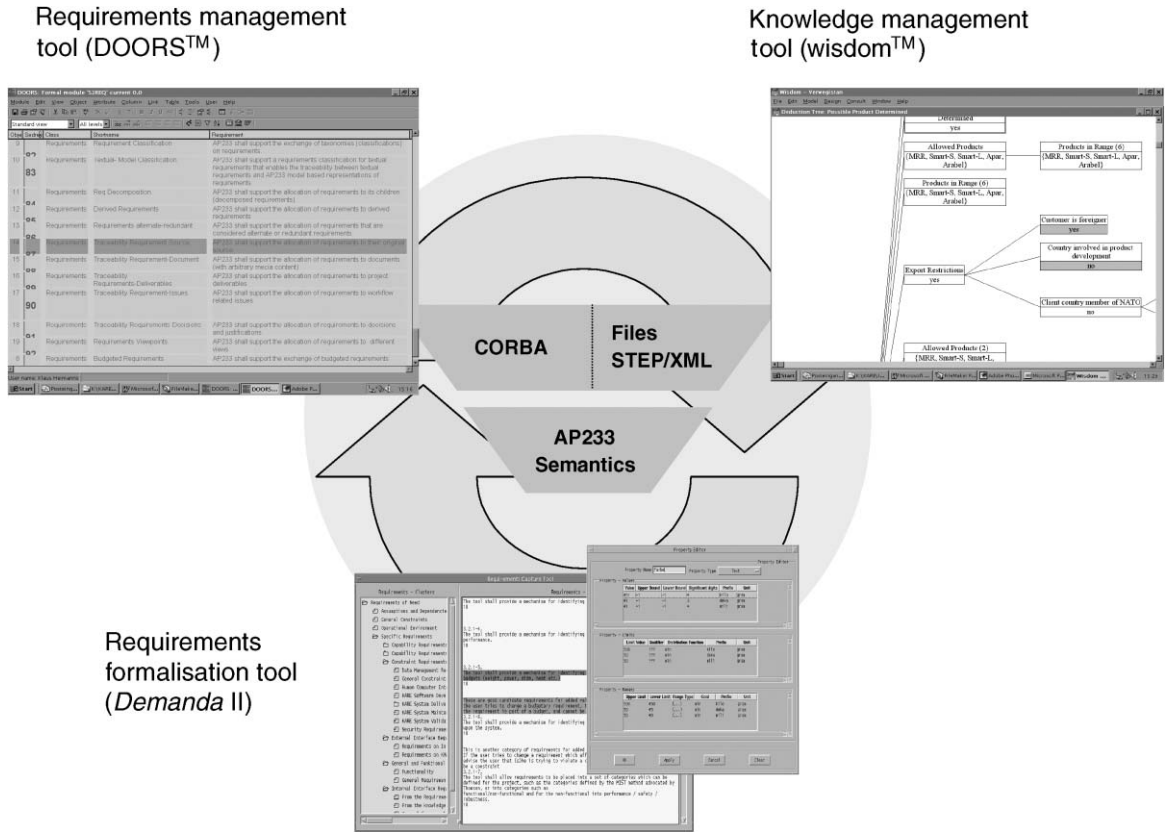


Figure 10. KARE 'workbench' components and architecture.

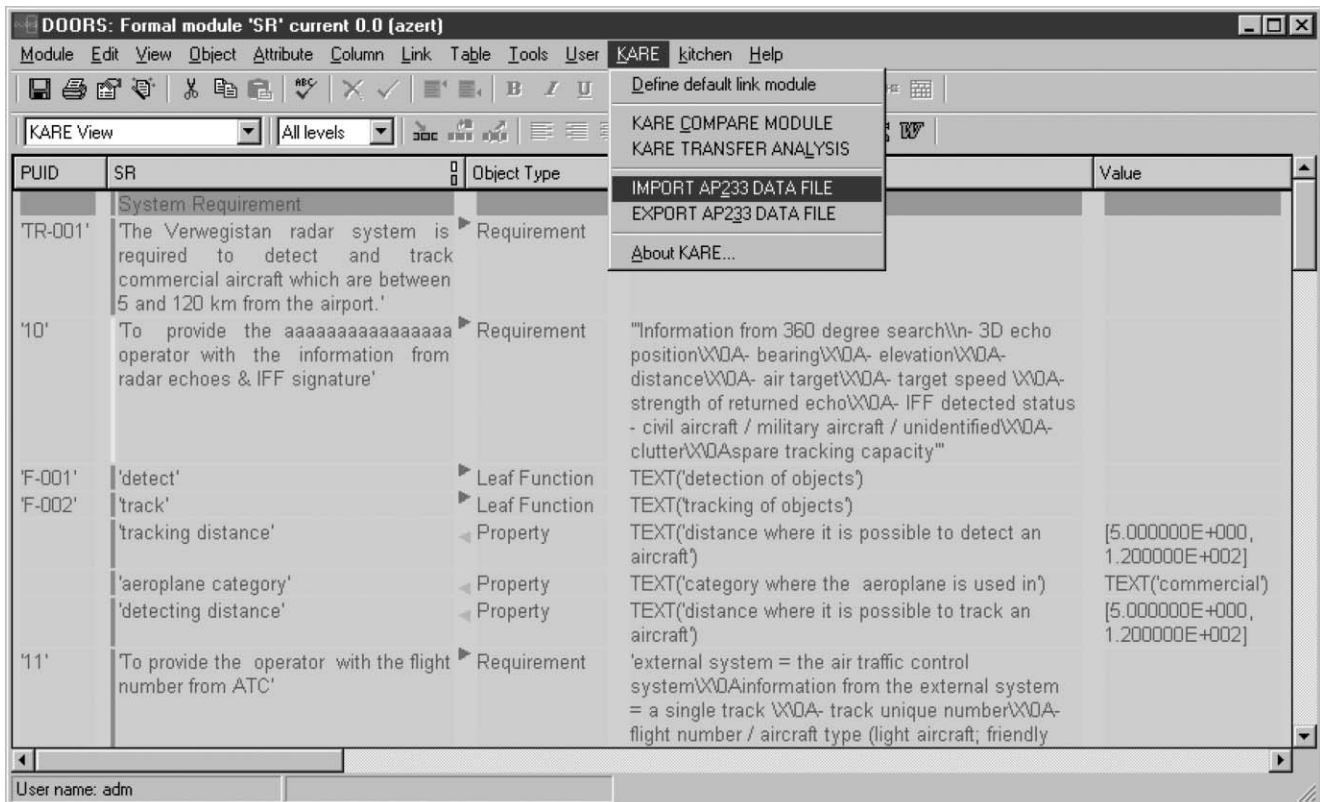


Figure 11. Screenshot of the KARE 'workbench' prototype using DOORS™.

## 7. Conclusions

The KARE methodology has been developed to support the sharing and reuse of knowledge in capturing, processing, and managing requirements in a one-of-a-kind environment. A new requirements engineering (RE) methodology has been developed that reflects the specific processes in specifying high-cost complex products. A key element of the reported research is the RE knowledge model that comprises task-, inference-, and domain-level knowledge categories specific to the system engineering activities at the early product specification stages.

The KARE RE knowledge model and methodology have been applied using industrial validation cases to illustrate the key benefits of the approach in terms of improved accuracy and consistency of requirement specification, reduced number of iterations and as a result, reduced overall lead time and cost of system specification.

## Acknowledgments

This research is part of the KARE project funded by the EU ESPRIT programme. The authors would also like to gratefully acknowledge the contributions of all partners to the research in the KARE project.

## References

1. Stevens, R., Brook, P., Jackson, K. and Arnold, S. (1998). *System Engineering, Coping with Complexity*, Hemel Hempstead: Prentice Hall Europe.
2. Daugulis, A. (2000). Time Aspects in Requirements Engineering: or Every Cloud has a Silver Lining, *Requirements Engineering*, **5**(2): 137–143.
3. Zhu, H. and Jin, L. (2000). Scenario Analysis in an Automated Tool for Requirements Engineering, *Requirements Engineering*, **5**(1): 2–22.
4. Gaines, B.R. and Shaw, M.L.G. (1992). Integrated Knowledge Acquisition Architectures, *Journal of Intelligent Information Systems*, **1**(1): 9–34.
5. Polanyi, M. (1967). *The Tacit Dimension*, London: Routledge & Kegan.
6. Popper, K.R. (1986). *Objective Knowledge – An Evolutionary Approach*, Oxford: Oxford University Press.
7. Kotonya, G. and Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*, Chichester: John Wiley & Sons.
8. Overmeyer, S.P. (1999). A Methodology for Constructing User-Oriented Requirements Specifications for Large-Scale Systems Using Electronic Hypermedia, *Requirements Engineering*, **4**(1): 1–18.
9. Hooks, I. (1990). Why Johnny can't Write Requirements, In: *Proceedings of AIAA Space Programs and Technologies Conference*, Huntsville, AL, pp. 25–28.
10. Lauesen, S. and Vinter, O. (2001). Preventing Requirement Defects: An Experiment in Process Improvement, *Requirements Engineering*, **6**: 37–50.
11. Darlington, M.J. and Culley, S.J. (2002). Current Research in the Engineering Design Requirement, Proceedings of the Institute of Mechanical Engineers, *Journal of Engineering Manufacture*, **216B**: 375–388.
12. Bray, I.K. (2002). *An Introduction to Requirements Engineering*, Harlow, Essex (UK): Pearson Education Ltd. ISBN 0201 767929.
13. Stadzisz, P.C. and Henrioud, J.M. (1998). An Integrated Approach for the Design of Multi-Product Assembly Systems, *Computers in Industry*, **36**(1–2): 21–29.
14. Fliedl, G., Kop, C., Mayr, H.C., Mayerthaler, W. and Winkler, C. (2000). Linguistically Based Requirements Engineering – The NIBA Project, *Data & Knowledge Engineering*, **36**(2): 111–120.
15. KARE, Knowledge Acquisition and Sharing for Requirement Engineering (KARE), ESPRIT Project No. 28916, www.kare.org.
16. Ratchev, S., Urwin, E., Muller, D., Pawar, K.S. and Moulek, I. (2003). Knowledge based Requirement Engineering for One-of-a-kind Complex Systems, *Knowledge-Based Systems*, **16**(1): 1–5.
17. Kang, K.C., Lee, K.W., Lee, J.Y. and Kim, G.J. (1998). ASADAL/SIM: An Incremental Multi-level Simulation and Analysis Tool for Real-time Software Specifications, *Software-practice & Experience*, **28**(4): 445–462.
18. Wang, J. (1999). Fuzzy Outranking Approach to Prioritise Design Requirements in Quality Function Deployment, *International Journal of Production Research*, **37**(4): 899–916.
19. Schreiber, G., Akkermans, H., Anjewierden de Hoog, R., Shadbolt, N., Van de Velde, W. and Wielinga, B. (2000) *Knowledge Engineering and Management – The CommonKADS Methodology*, The MIT Press, Cambridge, Massachusetts, USA.
20. Muller, D., Heimannsfeld, K. and Muller, N. (2000). Requirements Engineering Knowledge Management based on STEP AP233, In: *ProSTEP Science Days Conference Paper*, 114–123.
21. McDonald, I. (2000). ISO TC184/SC4 N911: Application Reference Model for the Exchange of System Engineering Data, ISO TC184/SC4 Document.
22. DOORS™, Quality Systems & Software Ltd., UK, 1998, www.telelogic.com.
23. WISDOM™, Arthur Andersen Business Consulting, Holland, 1997, www.arthurandersen.com.

## Dr Svetan Ratchev



Dr Ratchev is the Reader in Precision Manufacturing and head of the Precision Manufacturing Group in the School of Mechanical, Materials and Manufacturing Engineering at the University of Nottingham in the UK. Dr Ratchev is a Chartered Engineer and Member of the Institution of Electrical

Engineers. He has 20 years of relevant research experience in modeling, integration, and reengineering of reconfigurable manufacturing systems, collaborative product design and manufacture, knowledge-based systems in manufacturing, precision machining, and microassembly. He has published over 100 articles and has actively participated in a number of international conferences and seminars. He is the founding chair of the International Precision Assembly Seminar (IPAS).

#### **Professor Kulwant S. Pawar**



Professor Pawar is the Director of the Centre for Concurrent Enterprise and Professor of Operations Management at the University of Nottingham Business School. He has over 10 years industrial experience in product design and development, manufacturing engineering, and managerial environment

in large multinational firms. His research interests include managing new product design, linkages between product development and supply chain, managing design teams in virtual enterprises, organizational readiness for new product development, knowledge transfer and share in the extended enterprise. He has published almost 200

articles and actively participated in a number of international conferences. He is the founder and Chairman of the International Symposium on Logistics (ISL) and Co-chairman of the International Conference on Concurrent Enterprising (ICE). He is Editor-in-Chief of the International Journal of Logistics: Research & Application. He is the Coordinator of the Concurrent Enterprising Network of Excellence (CE-NET) in Europe.

#### **Dipl.-Ing. Dirk Müller**



Dipl.-Ing. Dirk Müller is a research engineer working at the Technical University of Clausthal in Germany. He specializes in requirements engineering and design data modeling. Dirk was born in 1969 and graduated from the Technical University of Clausthal in 2000. Since then he has been working as a Research Engineer in a

number of national and international projects. In the KARE project, Dirk was responsible for developing the requirements data model and had contributed to the development of STEP AP 233.