Requirements Engineering Knowledge Management based on STEP AP233

Heimannsfeld, K.; Müller, D.

Im Rahmen des Europäischen Forschungsprojekts KARE - Knowledge Acquisition and sharing for Requirement Engineering - wird ein Ansatz für wissensbasiertes Anforderungsmanagement unter Einbeziehung des sich in Entwicklung befindlichen Standards für repräsentierende Systemdaten, ISO 10303-STEP AP233, entwickelt. Dieser Ansatz unterstützt den Prozess einer systematischen Umwandlung von natürlichsprachigen Anforderungen in eine modellbasierte Darstellung der Anforderungen. Mit diesem Ansatz sollen konsistente und vollständige Produktspezifikationen in kürzeren Zeiten und somit geringeren Kosten und einer verbesserten Qualität erzeugt werden können.

In the European research project KARE - Knowledge Acquisition and sharing for Requirement Engineering - an approach to knowledge-supported requirements engineering and the relation to the emerging standard for representing systems engineering data, ISO 10303-STEP AP233, is being developed. The approach features the systematic transformation from natural language requirements into a model-based representation of requirements. The approach aims at producing consistent and complete product specifications in shorter times at substantially lower cost and with improved quality.

1 Problem definition and introduction

More often developers of technical products are faced with customer requests for a specific product, product adaptation or an invitation-to-tender (ITT). The building of a competitive tender, which means a profit for customer and supplier, is a significant factor for the success of a company. For the supplier and customer of build-to-order or one-of-a-kind products, a competitive bid for a complex product requires a substantial amount of time and money. As the complexity of the product increases the complexity of preparing a bid grows exponentially.

The acquisition process is based on the requirements of need statements supplied by the acquiring organization or enterprise. Ambiguity, incompleteness and inconsistencies characterize these customer requirements. In the specific domain of one-of-a-kind products like civil or military airplanes, train or transportation systems, the answering to a request, proposal or invitation-to-tender is a major cost factor. The development of a specification, on which a bid is based, is a time-consuming process with so far little automation support. Besides the cost of a bid itself, the cost for later rectification of inconsistencies and incompleteness of the initial specification will lead to significant higher costs later.

The ultimate goal is the reduction of the product definition time needed to prepare a bid and to increase the quality of the resulting specification by using model based product requirements, enterprise knowledge and inter-/intranet distributed engineering environments.

2 Requirements engineering knowledge approach

A vast amount of literature deals with the domain of knowledge engineering and management. It is not the aim of this paper to give an introduction. We will rather rely and describe the well-established facts. The knowledge engineering models can generally be categorised into three levels /5/:

- Domain knowledge
- Inference knowledge
- Task knowledge

The **domain knowledge** describes the main static information and knowledge objects in an application domain. It contains the structural information of a domain. From a STEP perspective all application protocols provide domain specific knowledge. Inside the KARE project, two parts are differentiated. First, there is the requirements engineering domain knowledge that is needed to represent requirements. The second part is the enterprise domain knowledge that is used to support knowledge functions operating on requirements engineering domain knowledge.

Inference knowledge describes how the static structures of the domain knowledge can be used to carry out a reasoning process. The category of the **task knowledge** supports the way to achieve goals by applying knowledge.

In KARE, the inference knowledge and the task knowledge are encapsulated in a specific knowledge management tool.

The main goal of KARE is the development of a requirement engineering process that is supported by knowledge. This places several requirements on requirements and knowledge engineering tools:

- The requirements domain knowledge must be semantically defined
- The enterprise domain knowledge must also be semantically defined
- The enterprise domain knowledge types must be extensible
- The knowledge functions encapsulating inference and task knowledge must also be extensible

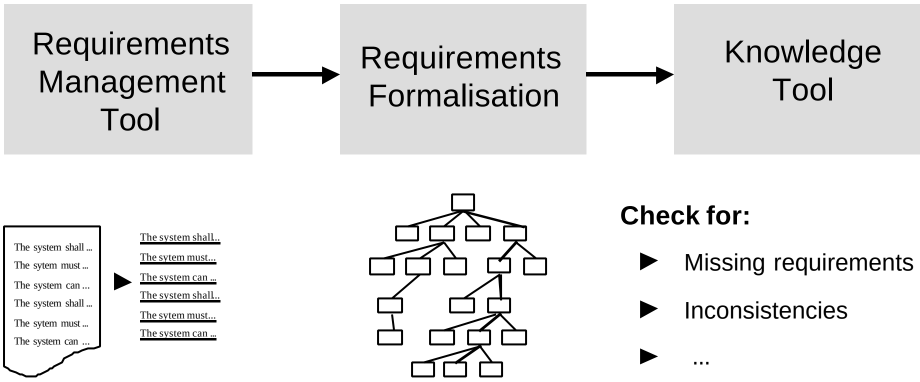

Figure 1: The requirements formalisation process in a conceptual view

To define the requirements domain knowledge, EXPRESS and the related STEP methodology was chosen. The resulting EXPRESS model, the KARE requirements model, was adapted and incorporated into current ongoing work developing the systems engineering exchange standard AP233 on ISO level /1,2,3/. The AP233 model features a model-based representation of requirements. This model-based representation of requirements allows the application of knowledge functions. A major problem today is that most requirements are specified in natural language. Section 3 describes how KARE resolves the problem by transforming natural language requirement statements into a model-based representation of AP233.

Based on the knowledge tool selected for the pilot implementation of the KARE workbench no choice has been made yet in regards to the enterprise domain model. However, it seems natural to adapt EXPRESS as a description language also for this type of knowledge. This would also facilitate the reuse of already existing application protocols as enterprise domain knowledge models and therefore reduce the time needed for knowledge acquisition, analysis and customisation in some relevant domains like shipbuilding, civil and chemical engineering.

3 Requirements formalisation

The specification of requirements in natural text is one of the main reasons for ambiguity and inconsistencies. Approaches to express natural language requirements in models are therefore a current topic of research /9/.

The KARE project intends to promote the current acquisition practices from a mainly document based approach to a systems engineering model based approach that allows the electronic interchange of all necessary data in a distributed engineering environment. The transformation of natural language requirement statements into a model-based representation of AP233 is named requirements formalisation. In this formalisation process, every requirement is being evaluated step by step in a systematic way. This will be realised by the requirements formalisation tool *demanda II*, which is developed within KARE. The single sequences of the whole formalisation process are illustrated in Figure 1.

3.1 Natural language requirements

Natural language representation is the initial form of defining a requirements statement. Documents with a specification of requirements are the basis for the formalisation process. They include every kind of information linked with the product. If the specifications are only available in paper form they have either to be scanned or have to be treated manually beforehand.

The documents are managed and handled within the requirements management tool DOORS™ on the level of natural language sentences. DOORS™ is used to provide typical requirements management functionality like configuration, change proposal and traceability management. For an extensive discussion of requirements management functionality, see /7,8/.

3.2 The requirements formalisation process

Each requirements statement specified in natural text is transformed into a system model description that represents the same information. This is done by identifying the four elements of a system specification as defined in AP233. These are the physical architecture, functions, behaviour and properties of a system.

In the following, we intend to give a brief description of the requirements formalisation process. The process is divided into three steps.

The lexical and syntactical analysis converts a sequence of characters in an ordered set of symbols and identifies the different parts of the speech. It also provides the hierarchical phrase structure of the sentences. As natural language permits ambiguity in respect to the phrase structure of a sentence, the user may have to resolve the ambiguity manually (Example*: The pupil saw the man with the telescope*).

The second step, word analysis, identifies the meaning of the different symbols in respect to the AP233 model. In other words, symbols describing a conceptual component from the physical architecture, a function, the behaviour or a property of the system are identified and defined.

As last step, the phrase analysis captures the relationships between the identified symbols. Possible relationships are hierarchy of components, interfaces between components and other relationships between components, functions and behaviour as specified in the requirements.

The content of the resulting model is of course not more correct or complete in its quality as the natural language requirements. However, the model offers the possibility to check by rules automatically on inconsistencies, missing or incorrect requirements. As an example, all system functionality can be checked for the existence of input/output specification. Of course, an existing specification does not indicate that it is correct, but a missing input/output indicates that further work is required.

4 The architecture of AP233

IS0 10303-STEP AP233 /1,2,3/ is an international standard for the computer-interpretable representation and exchange of systems engineering data. The objective is to provide a neutral mechanism capable of describing systems engineering data. The information exchange takes place in two directions. Horizontally the information is exchanged

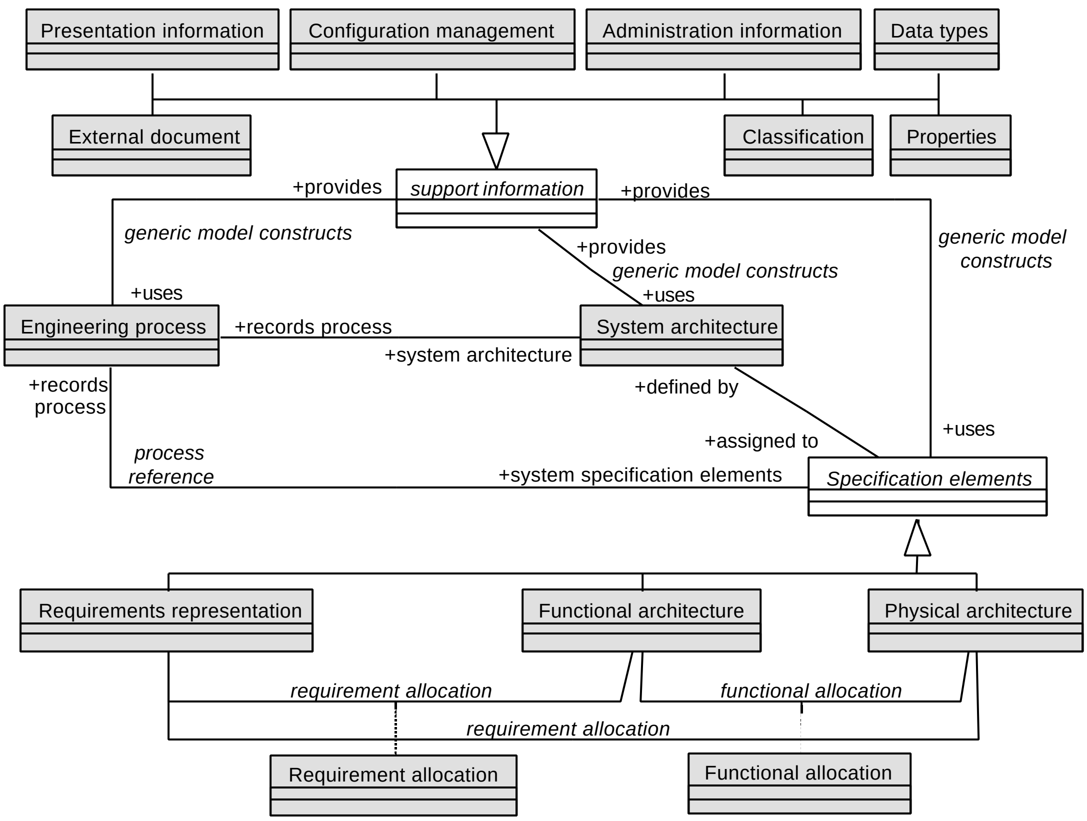

Figure 2: Conceptual view of AP233 as described in /4/

between system engineering tools. Vertically information is passed into lower tiers engineering domains (i.e. mechanical design, thermal design and engineering analysis) to create preliminary designs to support trade-off decisions at the systems engineering level. The application protocol defines the context, scope and information requirements for various development stages during the design of a system. The application protocol is applicable to any form of systems, such as an aircraft. The following is within the scope of the systems engineering application protocol AP233:

- 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 behaviour 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

4.1 Units of Functionality

Below, the Units of Functionality (UoF) belonging to the system model AP233 are summarised:

- System architecture
- Requirements
- Functional design
- Behavioural design
- Data Types
- Physical design / architecture
- Properties
- Graphics
- Configuration Management

Figure 2 shows a conceptual view of the AP233 system model in UML syntax. Every box represents a group of related entities. Respective to /4/ the system model's main groups are:

System architecture – representation of the building blocks for covering all information valid for a system, partial view of a system or system interface. There is also support for representing the system of systems structure.

Specification elements - defining the basic building blocks for representing requirements, functional and physical architecture.

Requirement and functional allocation - defining the mechanisms for tracing requirements to functions (including behaviour), as well as physical architecture elements and functional architecture elements to physical architecture elements.

Engineering process - covering the building blocks for activities in the engineering process, and associating specification information to related activities.

Support information - representing the building blocks for representing supplemental systems engineering information. This large group is composed of groups for representing configuration management information, visual layout information, mechanisms for referencing external documents, administrative information, data types and properties.

5 The KARE workbench architecture

The KARE workbench is a set of tools to support the requirements engineering, formalisation and management process on the acquirer and the supplier side. It consists of three main tools that are connected by different alternatives of interfaces. The workbench consists of a requirements management tool, a requirements formalisation tool and a knowledge management tool. Figure 3 shows the general architecture of the KARE workbench.

To connect these tools a common semantic description of required information in the requirements engineering domain was developed. This semantic description was included in the emerging ISO 10303-STEP AP233 standard (AP233 systems engineering exchange standard). The interface implementations based on ISO 10303-STEP AP233 / Part 21 files (STEP physical files) is currently under development.

The implementation of the KARE workbench is starting initially with simple file exchanges and will mature later to a full-distributed knowledge supported requirements engineering workbench. The integration with available and established requirements engineering tools like DOORS™, RDD-100™ or SLATE™ will initially be only supported by the file exchange interfaces. However, to allow a distributed working environment, it is foreseen that the respective tools are extended with CORBA interfaces to allow a tighter integration.

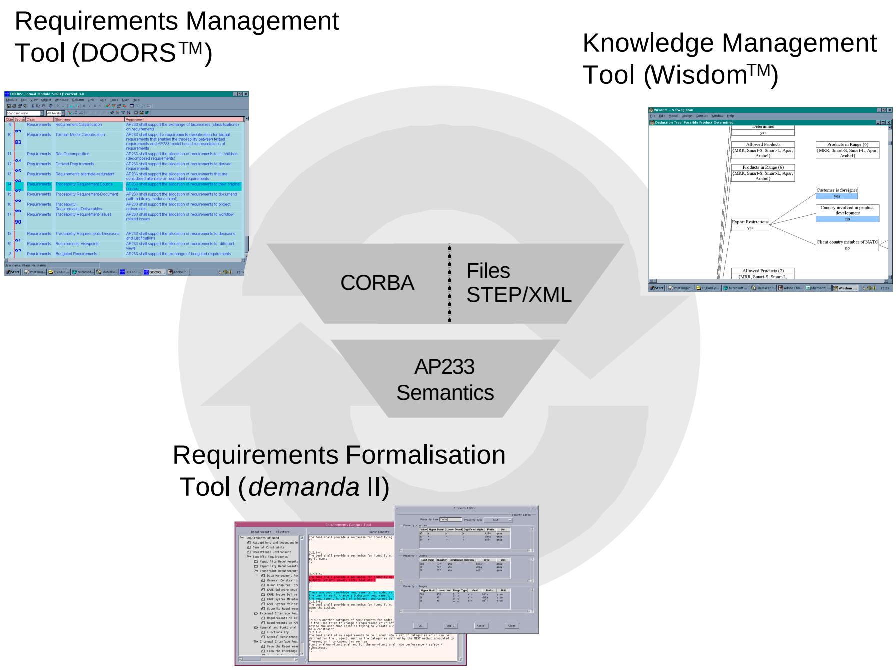

Figure 3: KARE Workbench components and architecture

The requirements formalisation tool includes different components to capture the requirements and to transform natural language requirements into an unambiguous, model based format. The requirements formalisation tool consists of the requirements capture & workflow, the requirements modeller and the requirements navigator. Underlying to all three modules is an AP233 repository.

The **requirements capture & workflow** tool contains the functionality to extract requirements from a customer or acquirer specification document. The workflow part of the tool deals with the exchange of requirements, issues, conflicts and questions between supplier and acquirer.

The **requirements modeller** is a tool, which allows to create or manipulate requirements and to transform them onto a subset of the AP233 model. The modeller identifies ambiguous words and leads the user to phrase requirements in a way that they are unambiguous and tightly defined.

The **requirements navigator** is a tool to view and browse all requirements and related structural and behavioural information. It provides an overview of the requirements traceability links. External traceability links provide the possibility to capture the source and rationale of each requirement or the design elements that address and fulfil the requirements in later design stages. Internal traceability links support various forms of traceability related to the derivation of a final requirements baseline.

A scaled-down application for users without requirements engineering legacy and Small or Medium Enterprises (SMEs) will be available as a stand-alone version of a PDM/EDM system. Underlying is a relational database schema based on parts of STEP AP233.

5.1 Requirements management tool

For the first prototype of the KARE workbench, the requirements management tool will be based on DOORS™ from Quality Systems & Software (QSS). DOORS™ will be adapted and extended as a client to both the knowledge management tool and the requirements formalisation tool.

Natural language requirements will be formalised in the requirements formalisation tool and passed back to DOORS™. Based on the formalised requirements the knowledge tool can detect missing, inconsistent or imprecise requirements.

5.2 Requirements formalisation tool

In KARE, the *demanda* requirements engineering tool /6/ will be adapted in order to support the formalisation requirements process. This adapted tool, named *demanda II*, will consist of a human computer interface that supports the requirement formalisation and a repository for formalised requirements.

5.3 Knowledge management tool

The knowledge management tool captures and formalises enterprise knowledge in terms of previous cases, enterprise capabilities, and business objectives or measures. For the first pilot showing applicability of the KARE approach the knowledge is implemented using a system that is based on decision tables.

In KARE, the internal Arthur Andersen tool WISDOM™ is used as a demonstrator.

6 Conclusions

With the ongoing development of interfaces and tools within the KARE project it is too early to make a final résumé. However, from an implementation perspective a few lessons learned have been evolved.

The implementation of AP233 (ARM, /1/) into a commercial PDM system with an underlying relational database (ORACLE©) was a manual and therefore resource intensive task. The current development of the AP233 interface for the requirements formalisation tool *demanda II* follows proven technology originally adapted in the development of the *demanda* requirements engineering tool /6/. The implementation is based on the commercial STEP ECCO toolkit from PDTEC GmbH, Germany. The tight integration of ECCO with the Tcl/Tk language used for implementing *demanda* and *demanda II* eases the implementation effort drastically compared to a full C or C++ API implementation.

Further research will look into the possibility to automate the requirements formalisation process through the application of techniques adapted from computational linguistics. Requirements defined in natural language are normally based on a limited subset of that language. Therefore, it is expected that automation can be achieved for most parts of natural language specifications.

The development in KARE is directly connected to AP233. The current developments are based on working draft 4 of AP233. Due to the current AP modularization and interoperability projects, major changes of AP233 are expected.

Further information about KARE can be found under http://www.kare.org/.

The authors gratefully acknowledge the hard work of participants in the supporting projects KARE and SEDRES-1/2. The financial support from the European Commission for the KARE project (Esprit 28916) is also gratefully acknowledged.

7 Summary

This article described the basic principle and structure of the workbench prototype, developed within the European research project KARE. This workbench supports the process of knowledge-supported requirements engineering on supplier side as well as on acquirer side. The representation of the requirements is based on the STEP AP233 data model. A formalisation process, supported by the workbench component *demanda II*, transforms the natural language defined requirements into a model-based representation.

8 Literature

/1/ Ian McDonald: ISO TC184/SC4 N911: Application Reference Model for the Exchange of System Engineering Data, ISO TC184/SC4 document, 2000

/2/ K. Heimannsfeld, H. Frisch: ISO TC184/SC4 N908: Application Protocol: System Engineering: Requirements for a data exchange facility, ISO TC184/SC4 document, 2000

/3/ Sylvain Barbeau: ISO TC184/SC4 N910: System Engineering: Application Activity Model, ISO TC184/SC4 document, 2000

/4/ Erik Herzog: AP233 Information Modelling Philosophy, internal working paper of the SEDRES-2 project, 2000

/5/ A. Th. Schreiber...[et al.]: Knowledge Engineering and Management: the CommonKADS methodology, The MIT Press Cambridge, Massachusetts, London, England, 2000

/6/ A. Große, K. Heimannsfeld: Wissensbasierte Störfall- und Fehleranalyse bei der Entwicklung verfahrenstechnischer Maschinen, Institutsmitteilung Nr. 21, Institut für Maschinenwesen, TU Clausthal, 1996

/7/ G. Kotonya, I. Sommerville: Requirements Engineering - Processes and Techniques, John Wiley, Worldwide Series in Computer, 1997

/8/ Karl E. Wiegers: Software Requirements, Microsoft Press, Redmond, 1999

/9/ J. Dallner, A. Günther, C. Rupp: Die Erstellung eines Objektmodells aus natürlichsprachlichen Beschreibungen - Regelbasierte Identifikation von Modellelementen und Requirement-Patterns zur Konstruktion von modellierbaren Anforderungen in natürlicher Sprache, SOPHIST GmbH, 1999