Proceedings of the TMCE 2008, April 21–25, 2008, Izmir, Turkey, Edited by I. Horváth and Z. Rusák

A METHODOLOGY FOR KEE SYSTEMS TARGET CASCADING

Marco Bertoni
Department of Mechanical Engineering
Politecnico di Milano
marco.bertoni@polimi.it

Christian Johansson
Tobias Larsson
Division of Functional Product Development
Luleå University of Technology
{christian.johansson, tobias.c.larsson}@ltu.se

Ola Isaksson
Volvo Aero Corporation
ola.isaksson@volvo.com

ABSTRACT

*The main aim of this paper is to present a methodology developed within the European Project VIVACE to guide the design and implementation of a Knowledge Enabled Engineering (KEE) system in a Virtual Enterprise. The proposed methodology tries to overcome some of the limitations which characterise traditional methods for Target Cascading, promoting a more collaborative and iterative approach to derive system specifications (in terms of advanced knowledge functionalities) from initial high-level targets. Social and behavioural aspects of Knowledge Management play a crucial role when many different users, knowledge experts, and process owners are involved in the Knowledge Management System (KMS) development. A well designed methodology is needed, therefore, to enhance communication and information sharing among design teams, to promote requirements merging and to take care both of the technological and behavioural aspects of the implementation. Initial business targets have been step-by-step decomposed into a set of sub-problems (*Service Requirements, Knowledge Issues, *and* Knowledge Challenges*) in the form of simple sentences in natural language. Then Quality Function Deployment (QFD) matrixes have been used to identify the set of functionalities to be implemented in the system, addressing the most important knowledge-related problems outlined in the cascading activity.*

KEYWORDS

Knowledge Enabled Engineering, QFD, Target Cascading, VIVACE project,

1. INTRODUCTION

In the continuous drive for competitiveness, knowledge has been recognised as a key enabler in improving quality, reducing costs and lead-time needed to develop new products. In order to achieve these ambitious objectives, a modern solution for Knowledge Management has to be conceived as a bridge between design teams, aiming at establishing and supporting a collaborative working paradigm in a Virtual Enterprise. The capability to set up successful communications and information sharing is particularly important for the aeronautical industry, which operates in a highly complex and dynamic relationship with a vast number of specialist companies, characterised by different skills, ways of working and organisational situation.

The design and implementation of a new Knowledge Management System (KMS) in such a multifaceted environment can be a complex and labour intensive task. In general terms, these systems are requested to provide a common answer, in the form of advanced knowledge management functionalities, to heterogeneous and partner specific knowledge-related issues. It can be very hard, however, to translate the desirable top level design targets, in a consistent and efficient manner, into appropriate specifications for the various sub-systems and components of the solution.

The main aim of this paper is, therefore, to present a methodology, to support the design and implementation of a new Knowledge Enabled Engineering (KEE) system in the product development area. The method aims to guide the

Knowledge Management System's (KMS') *Target Cascading* process, helping in translating initial high level business objectives into specific knowledge problems and system functionalities.

2. THE VIVACE PROJECT

The methodology here presented has been developed in the frame of a European 6th Framework Project called VIVACE (Value Improvement through a Virtual Aeronautical Collaborative Enterprise, http://www.vivaceproject.com). The main aim of the project is to support the design of a complete aircraft, including engines, by providing increased simulation capabilities throughout the product engineering life cycle, through the creation of a 'virtual product' in a 'Virtual Enterprise'. Final scope is to achieve a 5% cost reduction in aircraft development and contributions toward both a 30% lead time reduction and 50% cost reduction in engine development before 2020.

In such a context, KEE, intended as the exploitation of Knowledge Management within an engineering context, has been recognised as one of the most important key enablers for the development of such a virtual collaborative design environment. The paper here presented describes the approach used to define and develop the KEE system's advanced knowledge functionalities, starting from the analysis of the knowledge-related problems affecting the design process of some of the partners involved in VIVACE.

3. STATE OF THE ART AND RELATED ISSUES

KEE fundamentally means leveraging knowledge sources in order to enable engineers to complete their work quickly and correctly. Thus, KEE is about providing the right information to the engineers, at the right time, in the right format, in a collaborative environment which promotes learning within the organisation, across the supply chain and across the Virtual Enterprise.

The design and implementation of such a solution inside company's walls is still far from a piecemeal and easy task that organisations can undertake (Gallupe, B., 2001; Wong, K.Y. and Aspinwall, E., 2004). For a lot of reasons (Davenport, T.H. et al., 1997; Alavi, M. and Leidner, D.E., 1999; Liebowitz, J., 2001) introducing a new collaborative working paradigm is challenging, as proved by the fact that a significative proportion of these attempts usually fail (Storey, J. and Barnett, E., 2000).

Although many frameworks (McLure, M., 1998; Levett G.P. and Guenov M.D., 2000; Rubstein-Montano, R. et al, 2000; Sunassee, N. and Sewry, D., 2002; Hu, C.P.W.J.H. and Chen, P.H.H, 2002, Griman, A. et al, 2002, Wong K.Y. and Aspinwall, E., 2004) have been proposed to provide guidance on how to improve the successful incorporating of a new KM working paradigm into the organisation, the area still lacks of structured guidelines supporting system's specifications definition process. Most of these frameworks are not able, to go beyond the simple description of the KMS's "modus operandi". Suggesting generic principles about the way to carry out KM in an organisation (Wong, K.Y. and Aspinwall, E. 2004) represents just a part of the work needed to introduce a new KM approach, as the KEE, in an organisation. These high level concepts need to be "cascaded down" to lower levels of detail, in order to determine, both from a methodological and technological point of view, how the new solution has to be configured to really answer user needs and expectations.

Providing guidance to the KEE system's cascading process is the main scope of the work presented here. The driving hypothesis of this research work is that the Knowledge Management field still lacks clarity and commonly agreed guidelines to deal with KMS Target Cascading task. Traditional cascading methods, such as *Analytical Target Cascading* or *Collaborative Optimization* (Kim, H.M. et al., 2003; Allison, J. et al., 2005), although easily applicable for the development of mechanical artefacts, cannot be easily used in KMSs' design. Since the social and behavioural dimensions play a crucial role in the development and implementation of such systems, a mathematical formulation of the cascading problem has been judged not to be reasonably applicable in the knowledge domain. A successful implementation requires a multi-perspective analysis of the business process involved, and it is not just a matter of facts about how the system should be physically configured at a software level.

What a methodology for Target Cascading in this area is asked to do, instead, is to enable communication and mutual understanding among all the people participating to the system design process and to promote stakeholders involvement in decision-making. The methodology should be able to support the KMS design team in the definition of a common solution for Knowledge Management able to address heterogeneous and partner-specific knowledge-related problems.

3.1. Issues in requirements identification

"The hardest single part of building a system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements… No other part of the work so cripples the resulting system if done wrong. No other part is as difficult to rectify later" (Brooks, F., 1987)

As reported by Lubars, M. et al. (1993), a very small percent of a project's duration is spent in the identification of users' needs, although it costs between five and ten times more to repair errors during implementation than during preliminary design steps. For this reason, the requirements elicitation activity plays a crucial role in the system development process.

The importance of establishing explicit links between the business objectives and the technicalities of the system under development is widely recognised. Plenty of field studies (see, for instance, Macaulay, L., 1996) show that mis-identification of requirements during the conceptual design steps is one of the most significant sources of users dissatisfaction with delivered systems. The growing competitive importance of the KMS applications indicates, therefore, the need for both a better understanding of the real companies' needs and for a better derivation of information system requirements from this understanding (Bubenko, J.A., 1995).

People involvement has so to be considered the key-point of KMS development, especially when dealing with large implementations in complex and multi-faceted environments. User participation and contribution in the process largely determines the success of the reengineering effort. A solution unable to address real users' needs, will meet resistance strong enough to impede a complete adoption of the new working paradigm (Caron, M. et al., 1994, Kim, C.H. et al., 2001).

Broadly speaking, dealing with the conceptual design of a KMS means enhancing collaboration and interoperability among all the people involved in the specifications definition activity. Communication difficulties and lack of information sharing translate directly into misunderstandings for technology application design and development. From a business perspective, this situation often results in less than optimum design. Business processes can be overlooked and system features might be unresponsive to the business environment or incomplete for regular operations. From a technological perspective, lack of understanding about computer potential can result in disappointment based on unrealistically high expectations for system excellence (Damian, D.E.H., 2000).

4. WORKING METHOD

A typical cascading approach foresees that design targets would be initially defined at the top level of the multi-level design formulation and subsequently "cascaded down" to lower levels in order to derive a system description from a technical perspective. What the Target Cascading methodology is asked to do is, therefore, to guide the KEE development team in decomposing iteratively the starting problem in a set of independent sub-problems, taking care that system, sub-systems, and components depicted at each step would be consistent with the generic initial objectives.

The development of such a methodology in the frame of VIVACE has been driven by the need to:

- Enhance communication and mutual interaction between all potential system stakeholders (users, managers, system developers) upon specifications that really represent requirements in an unambiguous, consistent, and complete manner.
- Promote knowledge elicitation and sharing, facilitate discussion and improving a common understanding of the problem domain, in order to improve the capability to find commonalities across Use Cases and to merge requirements.
- Allow a comprehensive and structured development of system specifications taking care both of technological and human aspects of knowledge management, since any approach which doesn't take into account these two aspects has a high probability of failure (Soliman, F., 2000; Wong, K.Y., 2005).

4.1. Enhancing communication through the use of natural language

The necessity to enhance motivation and commitment of knowledge workers, professionals, and managers during the KMS deign process forced the KEE development team in finding an alternative to the traditional Target Cascading approaches proposed in the literature. It means that, instead of relying on a rigid and mathematical optimisation method, the work has been oriented towards the definition of a more qualitative and collaborative oriented approach, aiming to leverage people's capability to formalise and communicate their personal expectations about the new system. Knowledge-related issues and challenges have been addressed and solved making use of a natural language, through a step-by-step decomposition of the initial problem into lower level and more specialised matters. The Target Cascading problem has been formalised and represented at each step making use of simple sentences in narrative form. Initial assumptions have been translated into a set of more specific and functional-oriented sentences, with a different level of granularity. This discursive approach, has been adopted to cope with a more understandable and flexible formalisation of the cascading problem, in order to grant a continuous stakeholders interaction during the elicitation task, to promote innovative ideas and to get faster agreement when dealing with the decision making process.

4.2. Eliciting and merging requirements: The Use Case approach

Typically, at the beginning of the requirement elicitation activity, the knowledge about the system is coarse. Some features of the system are obvious, whereas for others only vague imaginations exist and the vision about the "To-Be" system is opaque. Moreover, when dealing with a re-engineering task which involves different company functions, people and processes, a common form of expression is needed to organise, compare and contextualise systems requirements. In order to overcome these difficulties and to facilitate the elicitation and merging process as well as the validation activity, the VIVACE community promoted the adoption of a Use Cases approach, since it was considered the best way to overcome such difficulties and to provide a common basis for the development of a collaboration environment.

Use Cases are considered by several authors as one of the best known and most widely employed requirements elicitation techniques in the industry (Lee, J. and Xue, N.L., 1999; Kulak, D. and Guiney, E., 2004). These descriptions have been based on the concept of *Scenarios,* which represent a possible way to use a system to accomplish some function the user desires (Hsia, P. et al., 1994). The Scenario approach has been considered by the KEE team the best way to identify, clarify and classify system requirements, and to build a common and shared vision of the future system. Scenario based representations provide, in fact, several benefits:

- *Simplicity*. Making use of the Scenario approach, the behavior of a large and complex system can be stated as a collection of independently and incrementally developed Use Cases (Hsia, P. et al., 1994), narrowing aspects of the system usage one at a time (Regnell, B. et al, 1995). Scalability allows considerable reduction of the complexity of requirements determination (Lee, W.J. et al, 1998).
- *User oriented approach*. Use Cases describe system functionalities as specific flows of events, represented directly from the user's point of view (Hsia, P. et al., 1994). Especially when dealing with the conceptual design of a new KMS, the adoption of such a perspective allows users in getting a faster agreement on system's features and characteristics (Lee, J. and Xue, N.L., 1999; Pohl, K. 1993).
- *Focus on early development stages*. Plenty of methods and techniques for system development exist today, but practically none of them address, in a structured way, the requirements generation stages, as well as the problem of moving from the vague and informal high-level needs to the formal domain (Bubenko, J.A., 1995). Moreover, existing methods are not adequate for explicitly capturing and representing, the "business and organisational knowledge" which is needed to properly guide the re-engineering effort. The adoption of a Scenario approach addresses these needs, enhancing the comprehension of the problem domain also from a business and organisational point of view.

The scenario approach has been chosen, at the end, as a common basis of expression both for the "customer" (companies, managers, or process owners), who determines the desired functions and capabilities, and the person who operates or uses the system (end user, administrator, casual user, etc.).

4.3. Developing a holistic solution: a framework for requirements classification

As already stated before, the development of a KMS is not just a technological issue, but it must take care also of people, products and processes involved in the process. In order to address initial High Level Objectives and to develop a successful working paradigm based on the collaboration between users, several aspects, mainly related to the behavioural and social aspects of Knowledge Management, need to be carefully considered. In order to ensure this multi-perspective approach during system design, a general framework has been proposed in order to classify heterogeneous requirements, functionalities and candidate technological solutions. The framework includes 4 different dimensions, named, respectively, *Application*, *Model*, *Organisation* and *Workers*.

Knowledge Enabled Applications (KEA). KM requires an adequate IT infrastructure in order to support in the best way the collaborative working paradigm. The development of an effective KEE system, which means a solution promoting accessibility and re-use of information, can be achieved by increasing software interoperability, which is separating knowledge models from applications and overcoming limitations related to the use of proprietary computing languages.

Knowledge Enabled Models (KEM). IT solutions, typically, support an approach that documents only the final result. Capturing the design rationale at the basis of the decision making activity, which is modelling the tacit and explicit knowledge resident on experts minds, is one of the key objectives of the collaborative working paradigm promoted by the KEE team. Mechanisms are required, therefore, to record this design rationale and to provide a “path of understanding” of design decisions.

Knowledge Enabled Organisation (KEO). KM is not only a technical concern but includes social aspects. Understanding the way knowledge circulates around the Virtual Enterprise, which means having a lifecycle perspective, leads to mechanisms for making this happen effectively. Only by resolving behavioural legacies, is it possible to develop an effective knowledge sharing community.

Knowledge Enabled Workers (KEW). KM is a labour-intensive task and there is an urgent need to assist workers in managing their knowledge effectively; that is to help individuals and teams in assimilating, creating, and recording vast amounts of cross-company and ad-hoc knowledge. The KEE solution aims to empower people with a global view by capturing and delivering the right knowledge at the right time. This will help innovation, freeing experts' time to concentrate on creative activities that do not “re-invent the wheel”.

5. THE KEE CASCADING APPROACH

The methodology presented in the following section aims at supporting the KEE development team in translating high-level system requirements (defined from a VIVACE Global perspective) into a set of functionalities the new solution should include once implemented. The approach chosen foresees several steps, as represented in Figure 1.

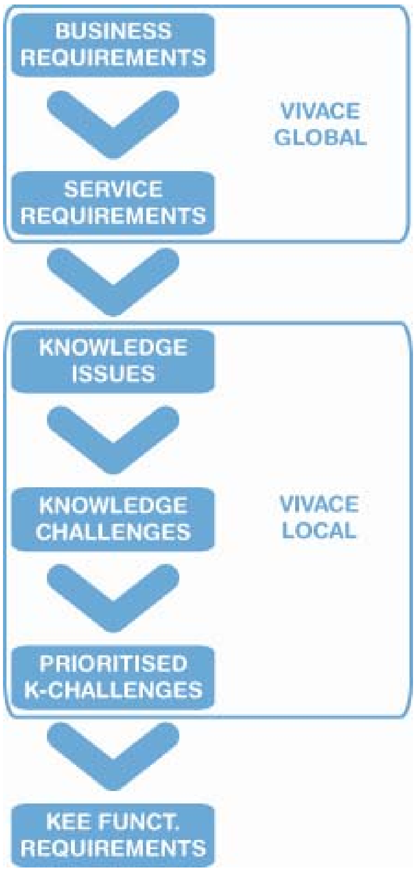

Figure 1 *Target Cascading* methodology

Step-by step, initial system requirements have been cascaded down into local and more specific knowledge-related issues and challenges. After several iteration steps it was possible to obtain the KEE system specifications description from a functional perspective. This information may be used, then, to define the system from a *Service* viewpoint and to derive a preliminary representation of the system’s architecture, to be validated and refined prior to the final implementation. The description of the low-level *Service Requirements* specification process, however, is not within the main scope of this paper, which is focused on the preliminary design steps of the KEE system development.

5.1. Use Case selection

The system's specification definition activity takes the move from the selection of a set of heterogeneous Use Cases considered relevant by VIVACE industrial partners for the KEE system development. These Scenarios express specific situations, related to the Use Case containing them, where there are opportunities for business improvement. Each Scenario has an associated process, with inputs, outputs and resources needed to perform a certain activity.

To be suitable as a KEE Use Case, the description and its corresponding Scenarios need to contain enough information to provide an understanding of the situation with respect to the knowledge-related issues. Moreover, these processes are asked to cover all the four dimensions of the framework presented in the previous section, which means they would be able to address aspects related to the KEO, KEM, KEW and KEA domains.

Four different Use Cases have been selected by the VIVACE partners participating to the KEE design (from Avio, BAE Systems, EADS and Volvo Aero):

- The increasing focus on the Virtual Enterprise context adds a number of questions related to intellectual properties, security, organisational and cultural issues. In order to address these issues, two different Use Cases have been selected, the *7Day Proposal (7Day)* Use Case, proposed by Volvo Aero, and the Knowledge Enabled Wing Engineer (KEWE) Use Case, proposed by Airbus, Assystem UK and BAE Systems (Sellini et al., 2006).
- The need to integrate in a better way different "sources of knowledge" and applications has been addressed in the *Multi-disciplinary Data Model* (*MDM)* Use Case proposed by Airbus, together with EADS. *MDM* aims at providing an infrastructure for enabling a consistent and efficient management of both process and product data with associated knowledge and decision.
- The *Turbine Rotor Design* (*TRD*) Use Case proposed by AVIO outlines, instead, the need of a more efficient use of experience and knowledge within design teams and design processes.

In order to move towards the definition of a common solution able to merge and address several heterogeneous requirements, a fifth Scenario, called *General Use Case (GUC)*, has been identified during the initial iteration steps. It served three purposes; first it provided other work packages with a concrete illustration of what Use Case information was needed by the work package. Second, it gave a means to capture KEE concerns not present in other VIVACE Use Cases. Third, it provided an evolving description used to collate all Use Case material into a single Use Case description.

Representing and sharing Use Case knowledge

All Use Case Scenarios selected have an associated process containing an abstract representation of actors, objects, relationships and information flows. In order to improve commonalities' detection, process owners have been encouraged to use an appropriate method for knowledge formalisation (in terms of As-Is activity flows' description, requirements and knowledge-related problems).

Considering the specific problem domain in which the KEE team was involved, the degree of formalisation of the problem had to be carefully balanced. The notation has been required, on one side, to be flexible enough to cope with the fact that some users are much more interested in a quick, intuitive and concise representation more than in a detailed description of process behaviour. On the other side, it needed to be formal enough to allow for the detection of inconsistencies among the partial information provided and not to lose the understandability of the requirements. Verbosity does not imply clarity of understanding and, in order to reduce the ambiguities often present in natural language, the KEE team was asked to introduce more technical information in the Use Case processes description.

Use Case scenarios have been therefore described making use of IDEF0 representations (Figure 2). The use of a graphical notation, allows expressing process behaviour in an easy understandable way and to express requirements as a complement to natural language.

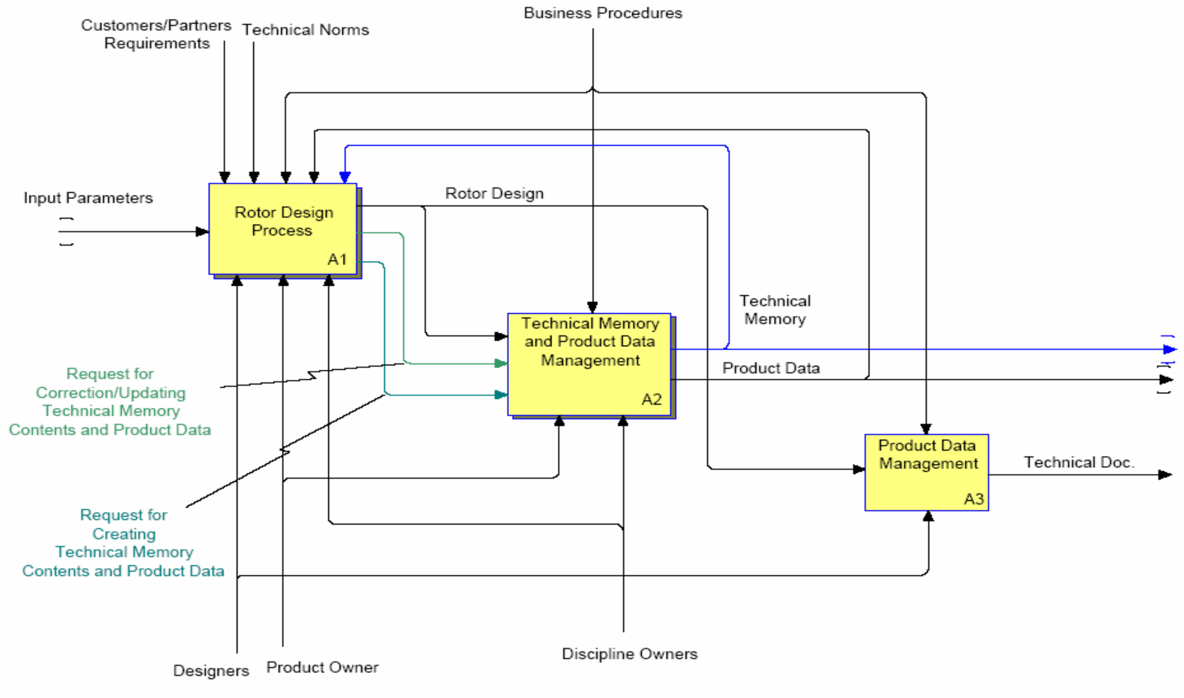

Figure 2: Example of IDEF0 diagram for Use Case description

The use of these models provides several benefits: users can grasp the meaning of graphical notation much easier than "technical language", process flows become more intuitive and the content can be formalised to such a degree that, sometimes, it can be processed by a computer.

In conclusion, since the idea of Use Cases is simple and their descriptions are based on natural concepts that can be found in the problem domain, the customers and the end users can actively participate in requirements analysis. In consequence, the developers can learn more about the potential users, their actual needs, and their typical behaviour.

5.2. Deriving *Business Requirements*

The end of Use Case analysis is a document that consists of sets of correct, complete, and consistent scenarios representations. This document is used as a guide for the definition and the collection of system's *Business Requirements* (*BRs*). A Knowledge Management initiative has to be clearly aligned with the business objectives, otherwise it will fail to accomplish goals that are visible to the organisation (Rubenstein-Montano, R., 2000). The KEE implementation has to be considered just a part of a high level plan that aims at supplying the organisation with the knowledge resources that it needs to carry out its vision and goals. As a result, the methods and tools for Knowledge Management have to be closely aligned to the overall company's strategy and must produce a tangible result to the organisation as a whole. Identification and addressing the correct BRs becomes crucial for the organisation (Hu C.P.W.J.H. and Chen P.H.H, 2002).

This elicitation activity aims, therefore, to translate the general knowledge-related statements coming from the Use Cases analysis into business-related problems. Since an effective elicitation is important because the system may not be acceptable unless it would satisfy company's management, a close interaction with customers, system users and owners is needed to identify those elementary actions on business processes, enabling the achievement of high level objectives (Sunassee, N. et al., 2002).

In order to provide a common basis of expression between all the participants of the KEE development activity, it was decided to formalise business requirements in form of:

[Something which is possible to measure] in a [business context]

A typical example of BR is represented by the following statement:

- *Reducing decision making process lead time in structure engineering simulations throughout the aircraft development process.*

This form of expression underlines the need to indicate or suggest improvements in a measurable form. BRs at this level, are strongly process oriented and are asked to show how a specific Use Case is able to address VIVACE global purposes. They justify why the VIVACE solution has been built and underline benefits for the customers and the business.

5.3. Deriving high-level *Service Requirements*

Since BRs don't provide any information about how the system has to be physically implemented, the KEE design team has been asked to consider what to include in the final solution in terms advanced knowledge capabilities. A further step in the KEE system specification process is, therefore, the definition of *Service Requirements* (*SRs*), describing from a high-level perspective what kind of services the KEE solution should be able to provide in order to achieve high-level targets. Outputs of this activity is a list of SRs, organised according to the framework proposed in the previous section, expressed in form of:

To be able to [verb] on a [product/process] aspect within a [sub-context]

BRs presented in the previous example have been observed from 2 separate perspectives:

- *To be able to set appropriate links between data, processes or decisions and the corresponding applied knowledge within the design process.*
- *To be able to set appropriate links between data, processes or decisions and the corresponding applicable knowledge within the design process.*

These SRs can be related to more than one knowledge category (*Knowledge Enabled* A*pplication/ Model/ Organisation/ Workers*). In this specific case, since the two service requirements analysed are quite generic, they refer both to the *Application* (context sensitive applications), *Model* (need to integrate design rationale to define applicability), *Organisation* (applicability embeds a lifecycle perspective) and *Workers* (delivering to the user the right knowledge) dimensions.

5.4. Deriving *Knowledge Issues*

The focus has been then oriented towards the identification of those specific knowledge-related problems (*K-Issues*) that needed to be solved in order to develop and deploy SRs. K-issues go deeper in detail with the problem description and represent a first attempt to shape the advanced knowledge-related functionalities to be implemented in the KEE system. K-Issues have been specified in form of:

We need to [Knowledge related need of action]

Also in this case there is no fixed mapping between SRs and K-Issues. However, as a general principle, since the aim is to better specify the key aspects and implications of a service, several K-Issues usually refer to the same SR. The cascading activity, moreover, is not performed making use of a mathematical formulation of the problem. K-Issues have been elaborated by means of discussions in a collaborative and iterative way by all the process stakeholders participating to the Work Package.

SRs presented in the example have been so re-elaborated in the following statements:

- *We have to structure our knowledge and to manage it along its lifecycle in order set consistent links with data, processes and decisions*
- *We have to provide means to retrieve applicable knowledge to processes, products and simulation models.*

5.5. Identifying *Knowledge Challenges*

K-Issues are shaped as generic and process-oriented purposes which need to be decomposed into a set of more specific and knowledge oriented sub-questions, named *Knowledge Challenges*. K-Challenges have been conceived as a compact and self explanatory description of specific knowledge matters and represent the maximum level of granularity a problem can be stated in the frame of KEE methodology. Three main elements are at the basis of K-Challenges specification: the *Knowledge Elements*, *Functions* and *Context*. A K-Challenge is therefore formulated in terms of what *Knowledge Elements* (intended as both explicit and tacit elementary pieces of knowledge, such as documents, rules or best practices) are needed to perform certain *Functions* in a specific *Context.* The question, therefore, takes this form:

How to [function] on [K-Elements] within [Context]?

Typically, K-Issues and K-Challenges are characterised by a 1:n correlation. Once more, there is no rigid mapping between these two entities. The transformation process has been conducted jointly by work package members, who have been asked, on the basis of their previous experience, to collaboratively refine and specify with a higher level of detail the list of K-Issues outlined in the previous phase.

Table 1 presents a set of K-Challenges derived from K-Issues presented in the previous section.

Table 1 Example of *Knowledge Challenges*

How can we manage knowledge about structure engineering simulation in an external knowledge repository?
How do we manage knowledge in configuration?
How can we provide tools to edit links between knowledge used, processes and decisions in a collaborative design environment?
How can we capture/define knowledge applicability within the product development process?
How can we define a query mechanism to retrieve applicable Knowledge within the design process?

5.6. Prioritising *Knowledge Challenges*

Due to the large number of K-Challenges that can be generated from the Use Cases, the KEE team has been asked to prioritise and select them in order to direct the re-engineering effort appropriately, since addressing each single statement has been considered infeasible due to budget and time constraints. Refining and ranking this list also provided a means to better specify and merge different requirements in order to avoid overlapping and find commonalities across Use Cases.

The method promoted for K-Challenges selection is based on decision theory, which provides a systematic way to choose between a set of alternatives. The decision tree approach has been, therefore, utilised to hierarchically structure this decision process. Each K-Challenge has been considered as a branch of the tree, and for each branch Use Case owners, the one closest to understanding the business needs, have been asked to estimate the associated *Reward*, *Cost* and *Likelihood*. *Reward* refers to the gain received by the company if the issue can be solved, *Cost* relates to the monetary resources needed to implement a solution able to address properly the Challenges, while *Likelihood* expresses the probability of deployment of such a solution The KEE team decided to adopt a *High*, *Medium* and *Low* rating as evaluation criteria. However, during the preliminary design steps it's not possible to assess with precision the values associated with these parameters. Several iteration loops have been set up, therefore, to refine the ranking on the basis of the list of possible solutions identified developing the KEE system from a technological point of view.

5.7. Defining KEE's *Platform Functional Requirements*

Guiding the design team in translating the low-level needs expressed by the K-Challenges into a set of technical and methodological solutions is one of the more critical aspects of the KMS cascading process.

To cope with this issue, each partner has been asked, at this stage, to define a list of functionalities addressing their knowledge-related problems. These *Functional Requirements* (*FRs*) are intended as possible answers to specific company's knowledge management requests. FRs express, in terms that are meaningful to the users and without taking physical constraints into consideration, the set of functionalities the KEE system must be able to provide at the end of the implementation.

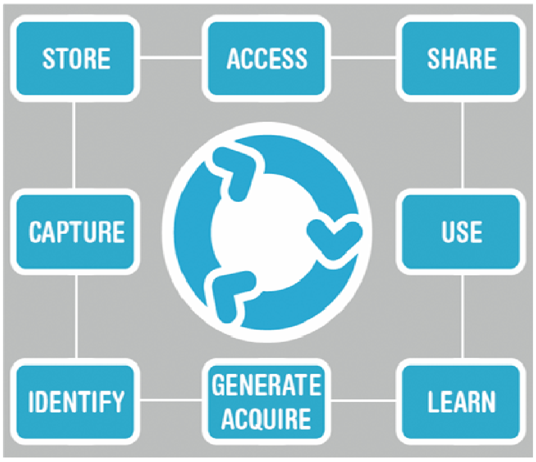

Figure 3 *Knowledge Lifecycle*

Since K-Challenges are usually characterized by various maturity and scopes, it may be difficult for the design team to gather them into a unique functional definition of the system. At this scope, the Functional Requirements structure has been defined basing on the Knowledge Lifecycle (KLC) model (Figure 3). The KLC is intended as a generic housing for any knowledge-related problems. It has the purpose to classify the Knowledge Management process in terms of methods, technologies and components. Every single step of the KLC framework has particular characteristics, includes different sub-steps and requires specific knowledge management capabilities. The model guides the KEE design team in sorting and merging heterogeneous partner requirements.

The KLC has been first applied to categorize all those partner-specific knowledge management solutions of interest for the development of the KEE system. FRs have been refined and grouped to identify a first set of common KEE Platform functionalities. This *Harmonisation* activity, lead together by process owners and knowledge experts, relies on the major hypothesis that some similarities could be found across Use Cases. Several iteration steps converged to the definition of a limited set of commonly agreed *Platform Functional Requirements* (*PFRs*). As generic rule, a PFR was considered of interest if linked at least with 3 different Use Cases. The Harmonisation task represents one of the main challenges (and achievements) in the frame of the KEE Work Package. The set of PFRs selected synthesises requirements with different maturity and scopes and describes features and characteristics of a common KEE solution able to handle knowledge in a Virtual Enterprise.

An example of Platform Functional Requirements, classified on the basis of the particular LifeCycle step to which they are associated, is given in Table 2.

Table 2 Example of *Platform Functional Requirements*

Identify the User's Context	I
Identify applicable K-elements for a Context	I
Capture context properties of Knowledge Elements	C
Capture Context of Use of Knowledge Elements	C
Search for K-Elements	A
Retrieve K-Elements	A
Allow early sharing of knowledge	SH
Store K-Elements	ST

A Quality Function Deployment (QFD) approach has been applied to map initial K-Challenges with the PFRs list. QFD was originally developed by Drs. Yoji Akao and Shigeru Mizuno in the early 1960s (Akao, Y., 2004). Quality Function Deployment is an overall methodology that begins in the design process and attempts to map the customer-defined expectation and definition of quality into the processes and parameters that will fulfil them. Table 3 proposes a graphical example of this mapping process. PFRs characterised by the highest absolute and relative importance (stated on the basis of the number of K-Challenges answered and on the K-Challenges ranking reported in the 2nd column) represent the set of core functionalities crucial for final system's implementation. In order to better evaluate the impact of a PFR on a specific K-Challenge, it was decided to avoid the simple binary check approach (0-1) and to slightly modify the *Analytic Hierarchy Process method*, or AHP (Saaty, T.L., 1990) based on 0-1-3-9 or 0-1-5-9 scale, by adopting a 0-1-3-5 scale to denote weak, medium and strong relationships between pairs of customer and technical design requirements.

Table 3 Example of application of the QFD method

		Identify the User's Context	Identify applicable K-elements for a Context	Capture context properties of Knowledge Elements	Capture Context of Use of Knowledge Elements	Search for K-Elements	Retrieve K-Elements	Allow early sharing of knowledge	Store K-Elements
How can we manage knowledge about structure engineering simulation in an external knowledge repository?	5	0	3	0	0	0	5	0	3
How do we manage knowledge in configuration?	5	0	3	0	3	0	0	0	0
How can we provide tools to edit links between knowledge used, processes and decisions in a collaborative design environment?	3	0	0	1	0	0	0	0	0
How can we capture/define knowledge applicability within the product development process?	5	3	5	0	0	3	0	0	0
How can we define a query mechanism to retrieve applicable Knowledge within the design process?	5	3	5	0	0	5	1	0	0
Absolute Importance		30	80	3	15	40	30	0	15
Relative Importance		14%	38%	1,4%	7%	19%	14%	0%	7%

Applying the QFD approach it was possible to define the complete list of functionalities to be replicated in the final system. Also in this case, however, several iteration loops are needed in order to refine the PFRs' definition and classification inside the matrix.

6. RESULTS

The application of the methodology in the frame of VIVACE brought, at the end, to the definition of an innovative Context-based Search Platform, enabling the user to perform multi-source searches for knowledge applicable to his or her engineering context.

The Platform integrates typical search engine functionalities with advanced context-based methods. The system is intended as an intelligent knowledge assistant that automatically provides the engineer with the contextualised *Knowledge Elements* (elementary pieces of knowledge, like *Practices*, *Norms* or *Drawings*) that he or she really needs in his/her design activity. *Context* can be defined as any information that can be used to characterise the situation of a user. Within the Platform, an engineer is described by a set of six context dimensions: Gate, Product, Discipline, Process, Project and Role.

The system enables the users to pre-define their company's context model instance and then to search for knowledge through a full-text search directly into the different knowledge sources used within the Virtual Enterprise. The information is retrieved on the basis of the user profile and the Platform automatically learns, through user feedback capture, the knowledge applicable to a specific engineering context

The software application has been designed following a "service-oriented architecture", and divided into three sub-systems developed using state-of-the-art technologies. Each sub-system (layer) is independent, and communication with other layers is guaranteed through internal web services. This choice gives the Platform great flexibility in terms of scalability and deployment scenarios.

The Context-based Search Platform has been tested through piloting within partner companies to validate the new concepts and functionalities in real company situations. User feedbacks have been collected by means of interviews and questionnaires. The results were documented in a pilot report, where they were measured against previously defined quantitative and qualitative indicators, to evaluate system effectiveness. These trials demonstrate the real benefits linked to the use of KEE solutions in a To-Be design scenario as compared to the initial As-Is situation.

7. CONCLUSIONS

The main aim of this paper is to present the methodology applied by the VIVACE KEE Work Package to develop the functional specifications of a new KEE system aiming to introduce a collaborative working approach in the aeronautical domain. After several iteration steps and further work from the technological viewpoint, the KEE application is almost completely defined in its general architecture and a Pilot project has been launched to test its feasibility. The methodology proved "on the field" to hit the target, ordering the translation of initial strategic issues and objectives into system functionalities. The collaborative and iterative approach chosen for Target Cascading showed to be successful in order to enhance user interaction and to gain a deeper common understanding on system specifications. The choice to deal with Use Case scenarios and to make use of a narrative approach together with graphical representations effectively demonstrated enhanced common understanding on issues and solutions. It facilitated process stakeholders in taking part to the As-Is and To-Be description of the system, building at the same time cross-functional collaboration across a working team and to set-up a common ground for the subsequent development steps.

8. DISCUSSION AND RECOMMENDATIONS

The methodology here described presents, however, some weak points. One of the problems emerging from the field relates to the fact that original phrasing can evolve, during and after the cascading process. Although the way the methodology is conceived promotes iterative mechanisms, a more structured and rigorous approach is needed in order to continuously update system requirements and to better address evolving user's needs.

Moreover, although the QFD method used for the PFRs determination showed facilitation of the concurrent capture of problem and solution domain knowledge without requiring complex formalisms, it seems not to be perfectly targeted for the specific problem under analysis. The downside of this approach is that it "only" allows evaluation of existing information, and it can be hard with this technique to identify new components or innovative architectures. It's difficult, moreover, to select the right scale to adopt to evaluate the functionalities against the associated challenges. The numbers in the matrix represent indications rather than facts. The mapping process is far from a mathematical formula, and it is strongly dependent upon experts' knowledge.

Finally, some problems arise when trying to evaluate the "performances" of the cascading activity for a comparison means. How to measure the methodology's effectiveness? Which metrics should be used? How to evaluate stakeholders' involvement and satisfaction? Further researches are needed to deal with these problems and to define a coherent and reliable metrics to be used to judge the approach a success.

REFERENCES

Akao, Y., (2004), "Quality Function Deployment: Integrating Customer Requirements into Product Design", ed. by Productivity Press, New York, NY.

Alavi, M. and Leidner, D.E., (1999), "Knowledge Management Systems: Issues, Challenges, and Benefits", Communication of the Association for Information Systems (AIS), Vol. 1, pp. 1-28.

Allison, J., Kokkolaras, M., Zawislak, M. and Papalambros, P., (2005), "On the Use of Analytical Target Cascading and Collaborative Optimization for Complex System Design", Proceedings of the 6th World Conference on Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil.

Brooks, F., (1987), "No Silver Bullet: Essence and Accidents of Software Engineering", Computer, Vol. 20-4, pp. 10-19.

Bubenko, J.A., (1995), "Challenges in Requirements Engineering", Proceedings of 2nd International Symposium on Requirements Engineering, York, England, pp. 160-162.

Caron, M., Jarvenpaa, S.L. and Stoddard, D.B., (1994), "Business Reengineering at CIGNA Corporation: Experiences and Lessons Learned from the First Five Years", MIS Quarterly, Vol.18, pp. 233-250.

Damian, D.E.H., (2000), "Challenges in Requirements Engineering", Department of Computer Science, University of Calgary.

Davenport, T.H., De Long, D.W. and Beers, M.C., (1997), "Building Successful Knowledge Management Projects: Managing the Knowledge of the Organization", Center for Business Innovation Working Paper, ed. by Ernst & Young LLP.

Gallupe, B., (2001), "Knowledge Management Systems: Surveying the Landscape", International Journal of Management Reviews, Vol. 3-1, pp. 61–77.

Grimán, A., Rojas, T., Pérez, M., (2002), "Methodological Approach for Developing a KMS: A Case Study"; CLEI Electron Journal, Vol. 5.

Hsia, P., Samuel, J., Gao, J., Kung, D. and Toyoshima, T., (1994), "Formal Approach Scenario Analysis", IEEE Software, Vol. 11-2, pp. 33-41.

Hu C.P.W.J.H., Chen P.H.H., (2002), "Design and Evaluation of a Knowledge Management System", Software, IEEE, Vol. 19-3, pp. 56-59.

Kim, C.H., Weston, R. and Woo, H.S., (2001), "Development of an Integrated Methodology for Enterprise Engineering"; International Journal of Computer Integrated Manufacturing, Vol. 14, pp. 473–488.

Kim, H.M., Michelena, N.F., Papalambros, P.Y. and Jiang, T., (2003), "Target Cascading in Optimal System Design", Journal of Mechanical Design, Vol. 125-3, pp. 474-480.

Kulak, D. and Guiney, E., (2004), "Use Cases: Requirements in Context", ed. by Addison-Wesley Professional, Boston, MA.

Lee, W.J., Cha, S.D. and Kwon, Y.R., (1998), "Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering", IEEE Transactions on Software Engineering, Vol. 24-12, pp. 1115-1130.

Lee, W.J. and Xue, N.L., (1999), "Analyzing User Requirements by Use Cases: A Goal-Driven Approach", IEEE Software, Vol. 16-4, pp. 92-101.

Levett, G. and Guenov, M.D., (2000), "A Methodology for Knowledge Management Implementation", Journal of Knowledge Management, Vol. 4-3, pp. 258-270.

Liebowitz, J., (2001), "Knowledge Management and its Link to Artificial Intelligence", Expert Systems with Applications, Vol. 20-1, pp. 1-6.

Lubars, M., Potts, C. and Richter, C., (1993), "A Review of the State of the Practice in Requirements Modelling", Proceedings of 1st IEEE International Symposium on Requirements Engineering, San Diego, CA.

Macaulay, L., (1996), "Requirements Engineering", ed. Springer-Verlag, London, UK.

McLure, M., (1998), "A framework for successful Knowledge Management Implementation", Proceedings of 4th American Conference on Information Systems, Huston, TX, pp. 635-637.

Pohl, K., (1993), "The Three Dimensions of Requirements Engineering", Proceedings of the 5th International Conference on Advanced Information Systems Engineering, Paris, France, pp. 275-292.

Regnell, B., Kimbler, K. and Wesslén, A., (1995), "Improving the Use Case Driven Approach to

Requirements Engineering", Proceedings of the 2nd International Symposium on Requirements Engineering, York, England, pp. 40-44.

Rubenstein-Montano, R., Liebowitz, J., Buchwalter, J., and McGraw, D., (2000), "A Systems Thinking Framework for Knowledge Management", Decision Support Systems, Vol. 31-1, pp. 5-16.

Saaty, T.L., (1990), "How to make a decision: The Analytic Hierarchy Process.", European Journal of Operational Research, Vol. 48-1, pp. 9-26.

Sellini, F., Cloonan, J., Carver, E. and Williams, P., (2006), "Collaboration across the Extended Enterprise: Barrier or opportunity to develop your Knowledge Assets?", Proceedings of the Tools and Methods of Competitive Engineering Symposia (TMCE 2006), Ljubljana, Slovenia.

Soliman, F., (2000), "Strategies for Implementing Knowledge Management: Role of Human Resources Management", Journal of Knowledge Management, Vol. 4-4, pp. 337-345.

Storey, J. and Barnett, E., (2000), "Knowledge Management Initiatives: Learning from Failure", Journal of Knowledge Management, Vol. 4-2, pp. 145-156.

Sunassee, N. and Sewry, D., (2002), "A Theoretical Framework for Knowledge Management Implementation", Proceedings of the 2002 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement Through Technology (SAICSIT'02), Vol. 30, pp. 235-245.

Wong, K.Y., (2005), "Critical Success Factors for Implementing Knowledge Management in Small and Medium Enterprises", Industrial Management & Data Systems, Vol. 105-3, pp. 261-279.

Wong, K.Y. and Aspinwall, E., (2004), "Knowledge Management Implementation Frameworks: a Review", Knowledge and Process Management. Vol. 11-2, pp. 93-104.