



<http://www.diva-portal.org>

This is the published version of a paper presented at *CIRP Design Conference 2008*.

Citation for the original published paper:

Bertoni, M., Bordegoni, M., Johansson, C., Larsson, T. (2008)

Pilot specifications definition guidelines for the implementation of a KEE solution in the aeronautical domain.

In: Fred J. A. M. van Houten (ed.), *CIRP Design Conference 2008* Enschede, The Netherlands:  
Laboratory of Design, Production and Management, Faculty of Engineering Technology, Univ. of Twente

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:bth-11274>

# Pilot specifications definition guidelines for the implementation of a KEE solution in the aeronautical domain

M. Bertoni<sup>1</sup>, M. Bordegoni<sup>1</sup>, C. Johansson<sup>2</sup>, T. Larsson<sup>2</sup>

<sup>1</sup>Department of Mechanical Engineering, Politecnico di Milano, Italy

<sup>2</sup>Division of Functional Product Development, Luleå University of Technology, Sweden

## Abstract

Designing and implementing a Knowledge Management System (KMS) in a Virtual Enterprise is a labour intensive and risky task. Solution prototypes (Pilots) are usually built to verify system effectiveness prior to final implementation. The paper proposes a methodology to guide this Pilot specifications definition process. These guidelines support engineers and knowledge experts in collaboratively defining functionalities, services, software components and performance indicators of the prototype. The methodology has been conceived and applied within the European project VIVACE, to support the development of a Knowledge Enabled Engineering (KEE) system in the aeronautical domain.

## Keywords:

Knowledge Management System, Knowledge Enabled Engineering, Pilot

## 1 INTRODUCTION

Developing and implementing a Knowledge Management System (KMS) aiming at supporting a collaborative working paradigm in a Virtual Enterprise, is a complex and risky task. Therefore, it would be useful to check, before embarking on a full scale implementation, if the solution satisfies initial expectations [1]. Physical prototypes (Pilots) are usually built for a preliminary evaluation of the KMS [2][3], to take the concepts out of the realm of theory and to provide empirical information of what can reasonably be expected by the new technology/methodology. However, it is not always easy to understand how to scale down the final system to obtain reliable feedbacks from the trials. The main aim of this paper is to present the methodology developed within the European project VIVACE [4][5] to support this Pilot specifications definition activity, dealing with the design and implementation of a Knowledge Enabled Engineering (KEE) solution in the aeronautical domain.

## 2 VIVACE KNOWLEDGE ENABLED ENGINEERING

VIVACE stands for 'Value Improvement through a Virtual Aeronautical Collaborative Enterprise' and it is a €70M Integrated Project in the EC Sixth Framework Programme (FP6). The main goal of the project is to improve the aircraft design process developing "virtual products in a Virtual Enterprise", pushing the European aeronautical industry to become the number one in the world with a market share of 50% in 2020 [6]. In such a context, the VIVACE Knowledge Enabled Engineering (KEE) Work Package aims to define and exploit advanced methods and tools to help companies in improving their engineering processes by leveraging past design experience. The main aim of this paper is to describe part of the work done by the KEE team to develop and implement this working approach. The methodology presented here supports the definition of KEE Pilot systems specifications and related evaluation metrics. The methodology is conceived as general purpose and can be applied also outside the boundaries of VIVACE.

## 3 THE USE OF DEMONSTRATION PROTOTYPES FOR KMS EVALUATION

As stated by several authors [7]-[9], the value of Knowledge Management (KM) is difficult to pinpoint as well as the real effectiveness of KM practices and technologies. Dealing with KMS design in a complex and multi-faceted environment, such as the Virtual Enterprise, is particularly challenging. The final full-scale system implementation is a significant investment [10]. It deeply impacts on design teams ways of working and on product development process performances [11]. Unexpected failures can cause big losses in terms of time and money. For these reasons a preliminary measurement of the KMS effectiveness has to be performed to better target the implementation at the real user needs [12].

An efficient way to validate KMS technology is to build a demonstration prototype [2][3][11]. On one side, prototyping spotlights hidden barriers and constraints limiting the capability of the system to correctly support the design process. On the other side, it helps in verifying users' ability to deal effectively with the new technology. Moreover, data obtained by the simulation demonstrate to senior management and to domain experts the intelligent behaviour and economic value of the system [11].

### 3.1 Issues in Pilot design and implementation

Prototyping is widely applied to evaluate system performance, since it takes less time to build than a delivered system [13]. On the authors' minds, however, the Knowledge Management area still lacks clear and commonly agreed guidelines to guide the Pilot specifications and metrics definition process [14].

A deep literature review in the Knowledge Management domain [15]-[20] has shown that very little attention has been given to the investigation of methodologies supporting Pilot design and implementation in all its phases. Piloting is more than just a proof of concepts. It involves gathering requirements for the requested functionalities, setting the infrastructure and landscape, and technically configuring the solution. On one side the Pilot should be small enough to make feasible its

implementation and testing. On the other side it should provide reliable information on the system's behaviour, which means it should replicate exactly, in a smaller scale, final system services. To obtain the best trade-off, Pilot design needs to be supported by appropriate methods and tools in each step of the specification definition process.

#### Pilot selection

Pilot development takes the move from the selection of the sub-part of the design process to be tested. It is required to be meaningful for evaluation purposes and to give reliable feedback both for process owners and users. This choice deeply impacts on time and costs associated with the testing activity. Appropriate methods and tools should be applied to support the KMS design team with the identification of the process to be piloted, since this choice largely determines the effectiveness of the final tuning process.

#### Pilot features definition

Once the sub-part of the solution/process to be validated has been identified, the Pilot needs to be defined from a functional viewpoint. Since the Pilot is requested to address heterogeneous and partner-specific needs, methods and tools selected to support this specification definition task should be able to enhance communication and collaboration across several working groups.

#### Metrics set-up

Through piloting, the design team aims to verify whether the initial knowledge needs have been satisfied. A set of qualitative and quantitative indicators may be defined and measured, therefore, to provide information for the final tuning process. These indicators are required to be targeted with the scope of the implementation, to be consistent, reliable and easy to implement and measure.

#### Benchmarks identification

Information obtained by the trials needs to be correctly benchmarked to assess the effectiveness of the new working approach [21]. As-Is process performances are usually very difficult to retrieve and use for this purpose. Therefore, data obtained by the trials are usually compared with a set of pre-defined target values, which need to be carefully established, in order not to cause (in the case they are considered too ambitious) demotivation or lack of interest towards the KMS [22].

#### Social and behavioural issues

A successful implementation of a KMS is not just a matter of how the system is realised at a technical level, but is also deeply linked to behavioural and social aspects of Knowledge Management [23]-[25]. Relationships between individuals and teams can both enable or inhibit the effectiveness of Knowledge management. Encouraging users in overcoming their communication barriers and applying the new collaborative working paradigm in their daily work is one of the main purposes of every KM effort.

## 4 A METHODOLOGY TO DEVELOP PILOT IMPLEMENTATION SPECIFICATIONS

The KEE system developed in the frame of VIVACE aims to support a collaborative working paradigm in the Virtual Enterprise. It is conceived as a bridge between design teams - able to provide a common answer, in the form of a common KEE Platform, to heterogeneous and partner-specific knowledge-related issues.

Moving from the considerations outlined in the previous section, the Pilot design process is decomposed into 2 levels. The first Pilot, named *Pilot Level 1*, focuses on the evaluation of the KEE software related capabilities. The

second Pilot, named *Pilot Level 2*, is implemented to test behavioural and social aspects of Knowledge Management inside the enterprise.

## 5 PILOT LEVEL 1 DESIGN APPROACH

The methodology applied to develop Pilot Level 1 transcends several steps, as outlined in Figure 1.

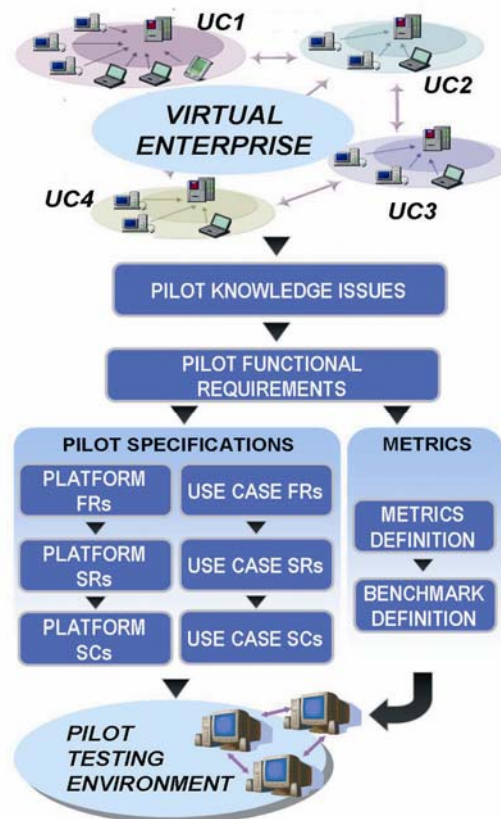


Figure 1: Pilot Level 1 specifications definition approach.

Initially, Use Cases (UCs) have been used to select the most meaningful part of the system to be piloted. The Scenario description provided a means to describe, for each UC, a first set of Pilot Knowledge Issues, then translated in form of specific Pilot Functional Requirements (FRs). FRs have been then refined and grouped in form of Platform Functional Requirements (PFRs) addressing common knowledge problems. Use Case-specific FRs, not addressed by any PFRs, have been cascaded down in parallel in order to identify a set of extension modules to be integrated with the common Pilot. PFRs have been then re-elaborated in form of Platform Service Requirements (PSRs), intended as a list of services providing the functionalities requested by the users. PSRs have been then mapped into a set of Solution Components (SCs), to define at a technical level the Pilot software architecture. In parallel, the attention has been oriented towards the definition of a set of metrics indicators and related benchmarks to be used for evaluation purpose.

### 5.1 Use Case Scenario for Pilot selection

First step of the Pilot specification activity is the selection of a sub-part of the Use Case Scenario (business cases showing knowledge management problems) relevant for the development and testing of the new solution. *Scenario* represents a possible way to use the system to accomplish some function the user desires [26]. This method has been applied since it is widely considered as one of the best known and most employed requirements elicitation techniques in the industry [27][28].

A preliminary As-Is analysis of these Scenarios has been performed to outline where most important knowledge management problems occurred in the design process. These descriptions helped in identifying the sub-part of the Use Case associated with the highest level knowledge needs. Pilots selected at this step mainly focused on knowledge identification and sharing aspects. Then, a Should-Be model has been built for each Use Case to make explicit future system developments and to collect ideas about the way the As-Is process could be improved.

Scenario representations provided a means to identify, clarify and classify Pilot knowledge issues. UML and IDEF modelling techniques have been used to support knowledge elicitation, formalisation and sharing. This user-oriented approach allowed improved common understanding on the problem domain, enhanced collaboration in Pilot design and promoted creativity among system's stakeholders.

## 5.2 Definition of Pilot level 1 specifications

Since a prototype usually addresses only a small part of the needs expressed by the users, a big effort is required to translate initial knowledge-related issues into the set of solution components the Pilot will include once implemented. This activity foresees several steps, presented in the following sections.

### Making explicit and formalising Pilot Functional Requirements

The results of Should-Be process analysis have been formalised in the form of Knowledge-related Challenges (K-Challenges), then translated in Pilot Functional Requirements, expressing at a technical level how these challenges could be addressed by the Pilot functionalities.

FRs express services that the system must be able to perform without taking physical constraints into consideration. They can be considered as a sort of draft To-Be model of the process to be piloted. During the first iteration, FRs showed to be redundant, overlapping, and to differ in terms of formalisation and level of granularity. To better explain their meaning and purpose and to facilitate discussions between the partners, each Use Case responsible has been asked to formalise his Pilot To-Be vision in form of a *mock-up*. Mock-ups represented a first attempt to build a prototype of the final solution; they consist of a sequence of slides, representing Pilot's interface screenshots, showing the way users can interact with the KEE system to solve their specific knowledge problems (Figure 2).

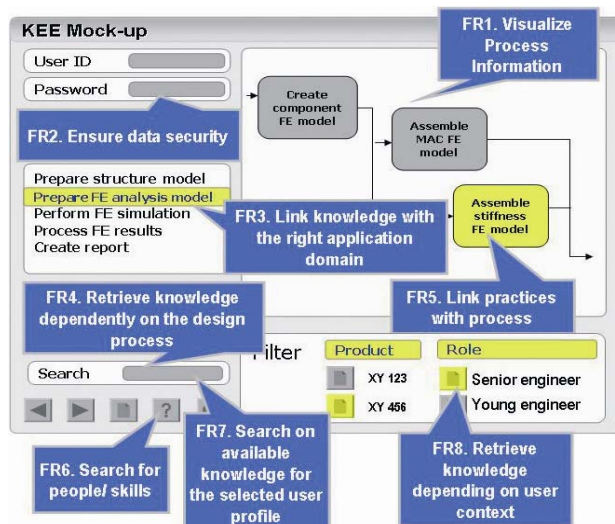


Figure 2: Example of Pilot Mock-up

The mock-up shown in Figure 2 gives an intuitive and easily understandable description of how a single Pilot would be configured if implemented. Mock-ups greatly enhanced the capability of the design team to detect commonalities across UCs, preparing the ground for the development of the common Platform Pilot. IDEF and UML models have been also used, together with mock-ups, to go deeper in detail with the Pilots description.

### FRs Harmonisation

The capability to address multiple and heterogeneous knowledge-related issues is crucial for any system aiming to link design teams with different competencies and responsibilities. Reaching an agreement on common Platform functionalities can be particularly hard in such a context, considering that each working team is most likely to be concerned about those behaviours of interest for their specific activity. However, these issues are rarely truly independent in practice, although they can seem to be poorly interrelated and linked together. Therefore, FRs previously identified have been reworked and synthesized to obtain so called Platform Functional Requirements (PFRs), representing the set of most important Platform functionalities to be piloted.

In order to ensure the Platform will cover all the relevant aspects of the knowledge lifecycle. PFRs have been collaboratively defined and categorised making use of the *Knowledge LifeCycle* (KLC) framework (Figure 3).

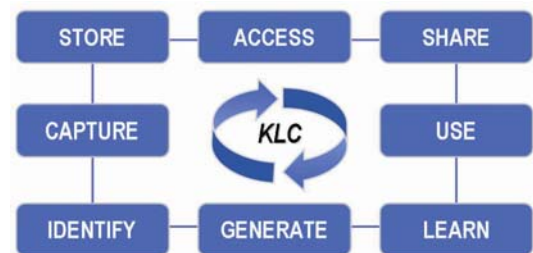


Figure 3: Knowledge LifeCycle

The KLC framework represents 8 steps in the knowledge lifecycle requiring specific Knowledge Management capabilities. It may be used as a base to classify methods, technologies and components supporting KM. Once PFRs have been collaboratively defined by partners at each step of the framework, it was decided to consider relevant for piloting purposes only those PFRs matched with 3 or more FRs coming from different Use Cases. Each PFR has been presented by means of a description, expected input and output, links to other PFRs and a general example of its application. To make clear the meaning of PFRs, a second Mock-up has been developed and used for discussion purpose.

### Definition of Platform Specifications

PFRs represent the Pilot just from a knowledge perspective. They do not include the basic management functionalities that are expected from any modern software application. Platform Service Requirements, defined collaboratively by system experts and process owners, collect a set of common software services needed to support the KEE capabilities in a real working environment. PSRs have been developed and grouped in nine different categories, named: *Access, Administration Services, Context, K-Elements/K-Sources Management, Methodology and Organisation, Search, Security, System Integration and User Interface*.

The list of PSRs has been mapped on PFRs using Quality Function Deployment (QFD) matrices. PSRs selected, the ones covering one or more PFRs, have been then mapped, always making use of QFD, on a list

of Solution Components (SCs) to define at a software level how the Pilot would be physically implemented.

#### *Definition of specific extension modules*

A similar approach has been followed to identify a set of extension modules to be integrated with the common Platform Pilot. These modules have been developed specifically to cover those FRs not addressed by any PFRs, but needed to answer important partner-specific issues. The KEE design team has been asked, therefore, to identify, for each UC, a set of specific Knowledge Elements (defined as elementary pieces of knowledge) and Knowledge Sources (containers of K-Element) to be integrated within the Platform. SCs defined at the end of this phase in some cases already existed within the companies, while sometimes they needed to be implemented specifically for the trials.

Service Requirements, Solution Components and the detailed list of Knowledge Sources to be integrated within the testing application constitute the main output of the Pilot implementation specification activity. This information has been used in the frame of the KEE development to physically build the testing environment and to design interconnections between heterogeneous software components and data warehouses.

### **5.3 Metrics definition**

In parallel with the definition of the Pilot technical details, the KEE team focused also on the identification of a list of performance indicators to be measured during the test. In general terms, it is difficult to identify one "right" set of measures to give reliable and intuitive feedbacks on system effectiveness [21]. The complex nature of measurement in KM has resulted in a plethora of definitions [7] and this lack of standards leads to proliferation of measures and difficulty in comparison. [8]. Several general frameworks actually exist to evaluate KMS success, such as the Balanced Scorecard [29] the APQC method [30] the Skandia Navigator [31], the Intangible Assets Monitor [32], the IC index [33], the KP3 methodology [3], together with a number of more specific tools and methods [34]-[36].

These methods, however, have been mainly developed to support a post-implementation analysis, focusing on the measurement of the intellectual capital of an enterprise. Moreover, they are strongly oriented towards the evaluation of business performance, resulting in being too vague and too focused on the strategic companies' objectives to be applied in the Pilot evaluation phase [37][38]. One on side, in the first steps of the implementation, the knowledge about the system is coarse and business estimates are very few reliable. On the other side, they do not suggest how to operate at a technical level to fix knowledge problems [7]. Metrics should include, at this step, heterogeneous types and classes of performance indicators to effectively communicate with all the key stakeholders [35].

To cope with these problems, the KEE design team proposes a new framework to guide the metrics selection task when dealing with the validation and testing of a KMS prototype. Together with this framework, a list of specific indicators reflecting the system's usage and efficiency have been designed and implemented to obtain qualitative and quantitative data about Pilot behaviour. These measurements are also linked with a set of guidelines suggesting how to fix a proper benchmark to carry out the validation task.

#### *Classes of indicators*

Three classes of indicators have been defined in conjunction with Use Case owners and final users to give

high quality feedbacks both at technological and methodological level (Cost of Implementation, Achievement of K-Challenges and New process advantages). Each indicator is identified by a name, a brief description, the unit of measurement (i.e. time, cost, frequency, etc.) and its specific domain of application.

*Cost of Implementation.* First group of indicators aims to directly estimate business outcomes prior to final implementation. A preliminary assessment of the KEE system impact on business performances may help, in fact, in determining the long-term viability of the initiative [22][39]. Better knowledge usage impacts on direct costs (material and labour costs), on overheads and on financial costs, producing in the long term a more efficient use of product development resources, decreasing product development time and production total costs.

*Achievement of K-Challenges.* These indicators directly show the success of implementing the KMS inside the companies. It is important to focus on factors that affect the ability to achieve strategic objectives [35], measuring the level on which initial K-Challenges are achieved. *Frequency of use, Degree of usability and Degree of accessibility* are three of the main parameters related to the capability of the Platform to share knowledge among all process stakeholders. Moreover, also the way in which K-Elements are formalised and updated in the databases and the way in which different K-Sources interact together is interesting to be measured and evaluated.

*New Process Advantages.* Evaluating the system just from a business perspective can be misleading and counter productive [38]. Metrics need to be expanded to capture the impact of the new knowledge management approach on organisational performances [40]. A different set of indicators, addressing time and quality issues has been elaborated, in order to show how product development process improves due to the new KEE solution. The effective use of information can substantially reduce the "time to market" for new products and improve its quality, reducing, for instance, the number of inconsistent results or the number of non-conformities found during a simulation process.

### **5.4 Benchmark definition**

The KEE design team developed a specific set of benchmarks to assess Pilot effectiveness. They are not intended to be ideal system targets. On one hand it is not easy to determine what the "optimum solution" would look like, while, on the other hand too high expectations can often result in demotivation amongst systems stakeholders. The benchmarks proposed are not supposed to be optimal, but a sort of trade-off between an ideal target and a more realistic and practical provision. Their definition have been collaboratively led by Use Case and KEE experts according to their previous experiences in the field, to grant a more direct and intuitive comparison of system performances.

## **6 PILOT LEVEL 2 SPECIFICATION APPROACH**

Improving Knowledge Management effectiveness is not just a technological issue. It is also important to educate users how to master the technology [10]. For this reason, the KEE design team adopted a multi-aspect approach by defining two levels of the Knowledge Enabled Solution to be implemented and tested. This second prototype, named Pilot Level 2, focus on the behavioural and methodological issues that arise from the necessary change in working practices as a result of implementing the KES Platform within an organisation.



A set of guidelines have been defined to suggest to users how to overcome their knowledge related problems within the company. They consist of good practice documents, lessons learnt, templates or perhaps validation of evidence of case studies from other organisations. Pilot Level 2 does not address any specific Use Case and the trials have been abstracted away from a specific process, to enable the guidelines to be generic.

### 6.1 Pilot Level 2 selection

Pilot Level 2 specification activity took the move from the identification of the high level VIVACE objectives strictly linked to specific knowledge matters, to tie the implementation into key business drivers for the organisation and to persuade users to change their working practices. Business requirements have been then cascaded down to specify important knowledge issues to be addressed at cultural and behavioural level. They mainly refer related to the possibility to increase cross-project and cross-departmental information sharing, to improve re-use of existing knowledge and to ensure intellectual property.

### 6.2 As-Is analysis of the process

An extensive As-Is analysis of working practices has been performed to capture user requirements (through interviews) and to outline most important knowledge aspects to be tested during the trial. Brainstorming sessions have been set up to help the KEE team to analyse how design teams worked with already in use technologies. This information has been used as a baseline to set up the Pilot Level 2 environment.

### 6.3 Pilot Level 2 specifications definition

The features undergoing Pilot Level 2 validation and testing have been grouped in four different categories:

*Methodologies for Lessons Learnt.* The aim of this prototype is to test the capability to capture and communicate *Lessons Learnt* inside the company, addressing at the same time intellectual property management issues.

*Gate Maturity Assessment* trials provide a summary and recommendations on how to apply maturity techniques in a Stage-Gate process [41], in order to avoid costing iterations and to reduce design process lead-time.

*Team Relationship Assessment* trials deal with the analysis and validation of relationships and trust in collaboration between design teams. The main aim is to provide useful information on how to deploy these techniques in the context of a Virtual Enterprise.

*Knowledge Sharing* trials investigate how knowledge is captured, stored, accessed, exchanged, used, and re-used through the Virtual Enterprise, and how social and organisational aspects enable or inhibit effective collaborative working.

### 6.4 Pilot Level 2 metrics definition

Pilot Level 2 metrics have to be designed with the scope to understand whether the trials have been beneficial and contribute to the KEE design overall goals. Guidelines effectiveness has been measured in terms of how they impact on the "collaborative technology" approach, measured on the basis of the As-Is design process performances. In order to maintain homogeneity with Pilot Level 1, indicators have been categorised into the three macro classes previously introduced.

On one side, users have been asked to assess whether outcomes and working processes were better, worse, or unchanged from their previous method of conducting business. These surveys and interviews helped the KEE

design team to understand how the Pilot effectively performed against user requirements.

Quantitative measurements have been collected via network analysis and formal questionnaires to provide a "snapshot" of how the team operated and interacted during the tests. Time and quality indicators, compared with old system performances, underlined main improvements in the design process - i.e. number of design options investigated, thoroughness of the investigation of options and lead time reduction.

## 7 RESULTS

The final result of the Pilot specification definition activity is a set of tools and guidelines that enables Knowledge Management in a Virtual Enterprise environment. Two different demonstration prototypes have been developed. Pilot Level 1 concerns the application of a KEE system for the management of technical knowledge during the robust multi-disciplinary design of a turbine rotor disks. The prototype has been developed to support a modern Stage-Gate process and it enables engineers to search for applicable knowledge needed to accomplish their specific task. Pilot Level 2 is associated with a major product change for the re-design of a winglet. It focuses mainly on knowledge sharing aspects, showing how knowledge is captured, stored, accessed, exchanged, used and re-used throughout the Virtual Enterprise. Guidelines have been validated internally by two of the VIVACE partners, in order to obtain less subjective data from the tests.

## 8 CONCLUSIONS AND FUTURE WORK

The methodology presented in this paper proved "on the field" to be targeting the scope of VIVACE. It showed it was effective in improving communication among system stakeholders and in merging heterogeneous knowledge-related issues during the definition of Pilot technical details. However, further efforts are needed to assess the consistency of the methodology. First, it is difficult to understand how much the final Pilot configuration really reflects user requirements. Knowledge-related issues can evolve during specification definition process and an iterative mechanism could be helpful to continuously update system requirements during Pilot development.

Then, data reliability should be better analysed to verify how information obtained by the trials is influenced by the particular testing conditions. Use-cases usually don't exist in real life in the way that they have been defined. Therefore, the Pilot is often not tested in the same To-Be environment that the solution was developed for. Providing reliable feedbacks, especially from a business point of view is particularly challenging in such a context.

Streamlining and validating the Pilot design process is one of the main scopes of future researches in this field. It would be useful, moreover, to develop a set of guidelines helping the process stakeholders in correctly interpreting data obtained by the trials, to guide decision making during the tuning process.

## 9 REFERENCES

- [1] Levett, G.P., Guenov, M.D., 2000, A methodology for knowledge management implementation, *Journal of Knowledge Management*, 4/3:258-270.
- [2] Presley, A., Liles, D., 2000, R&D Validation planning: a methodology to link technical validations to benefits measurement, *R&D Management*, 30/1:55-65.

- [3] Ahn, J.H., Chang, S.G., 2002, Valuation of knowledge: a business performance-oriented methodology. Proc. of the 35th Hawaii Int. Conference on System Sciences, IEEE Press, Los Alamitos CA, 2619–2626.
- [4] VIVACE Project official website: <http://www.vivaceproject.com>.
- [5] Bordegoni, M., 2006, Specification for Pilots-Version 2, VIVACE Project Official Deliverable.
- [6] Advisory Council for Aeronautics Research in Europe (ACARE) official website: <http://www.acare4europe.org/>.
- [7] Ahmed, P. K., Lim, K. K., Zairi, M., 1999, Measurement practice for Knowledge Management, *Journal of Workplace Learning*, 11/8:304-311.
- [8] Kankanhalli, A., Tan, C.Y., 2004, A Review of Metrics for Knowledge Management System and Knowledge Management Initiatives, Proc. of the 36<sup>th</sup> Hawaii International Conference on Systems Sciences, Big Island, Hawaii.
- [9] Wei, C.-P., Hu, J.-H., Chen, P.H.-H., 2002, Design and evaluation of a Knowledge Management System, *IEEE Software*, 19/3:56-59.
- [10] Cupello, J.M., Mishevich, D.J., 1988, Managing prototype knowledge/expert system projects, *Communications of the ACM*, 31/5:534–550.
- [11] Gallupe, B., 2001, Knowledge Management Systems: surveying the landscape, *International Journal of Management Reviews*, 3/1:61-77.
- [12] Jennex, E.M., Olfman, L., 2005, Assessing Knowledge Management Success, *International Journal of Knowledge Management*, 1/2:33-49.
- [13] Hewett, J., Sasson, R., 1986, *Expert Systems Volume I*, Ovum Ltd., London, UK.
- [14] Ramesh, B., Tiwana, A., 1999, Supporting collaborative knowledge management in new product development teams, *Decision Support Systems*, 27/2:213–235.
- [15] Davenport, T.H., De Long, D.W., Beers, M.C., 1997, Building Successful Knowledge Management Projects, *Sloan Management Review*, 39/2:43–57.
- [16] Wiig, K., de Hoog, R., Van der Spek, R., 1997, Supporting Knowledge Management: a Selection of Methods and Techniques, *Expert Systems With Applications*, 13:15-27.
- [17] McLure, M., 1998, A Framework for successful Knowledge Management Implementation, *Proceedings of 4th American Conference on Information Systems, USA*, 635-637.
- [18] Gersting, A., Gordon, C., Ives, B., 1999, Implementing Knowledge Management: Navigating the Organizational Journey, *Journal of Knowledge Management*.
- [19] Grimán, A., Rojas, T., Pérez, M., 2002, Methodological Approach for Developing a KMS: A Case Study, *Proceedings of the 10<sup>th</sup> American Conference on Information Systems, New York, NY*.
- [20] Wong, K.Y., Aspinwall, E., 2004, Knowledge Management Implementation Frameworks: a Review, *Knowledge and Process Management*, 11/2:93-104.
- [21] Fui-Hoon Nah, F., Lee, J., Nah, F.F.H., Lau, J.L.S., 2001, Critical factors for successful implementation of enterprise systems. *Business Process Management Journal*, 7/3:285-296.
- [22] Sunassee, N., Sewry, D., 2002, A Theoretical Framework for Knowledge Management Implementation, Proc. of the SAICSIT'02 Conference, 30:235-245.
- [23] Alavi, M., 1999, Knowledge Management Systems: Issues, challenges and benefits, *Communications of the AIS*, 1/2:1-28.
- [24] Hall, G., Rosenthal, J., Wade, J., 1993, How to make reengineering really work, *Harvard Business Review*, 71/6:119-131.
- [25] Stoddard, D.B., Jarvenpaa, S.L., 1995, Business process redesign: tactics for managing radical change, *Journal of Management Information Systems*, 12/1:81-107.
- [26] Hsia, P., Samuel, J., Gao, J., Kung, D., Toyoshima, T., 1994, Formal Approach Scenario Analysis, *IEEE Software*, 11/2:33-37.
- [27] Lee, W.J., Xue, N.L., 1999, Analyzing User Requirements by Use Cases: A Goal-Driven Approach; *IEEE Software*, 16/4:92-101.
- [28] Kulak, D., Guiney, E., 2004, *Use Cases: Requirements in Context*, Addison-Wesley, Reading, MA.
- [29] Kaplan, R.S., Norton, D.P. 1992, The Balanced Scorecard. Measures that drive performance, *Harvard Business Review*, 1:71-79.
- [30] Lopez, K., 2001, How to Measure the Value of Knowledge Management. Appropriate metrics for each stage of KM implementation, *Knowledge Management Review*.
- [31] Edvinsson, L., 1997, Developing intellectual capital at Skandia, *Long Range Planning*, 30/3:266-373.
- [32] Sveiby, K.-E., 1997, The Intangible Assets Monitor, *Journal of Human Resource Costing and Accounting*, 2/1:73-97.
- [33] Roos, J., Roos, G., Dragonetti, N.C., Edvinsson, L., 1997, *Intellectual Capital: Navigating in the New Business Landscape*, Macmillan, Basingtoke, UK.
- [34] Roy, R., Del Rey, F.M., Van Wegen, B. Steele A., 2000, A Framework to Create Performance Indicators in Knowledge Management, Proc. of PAKM2000 Conference, Basel, Switzerland.
- [35] US Navy, 2001, Metrics guide for knowledge management initiatives, a United States Department of Navy internal report.
- [36] De Gooijer, J., 2000, Designing a Knowledge Management Performance Framework, *Journal of Knowledge Management*, 4/4:303-310.
- [37] Andersen, A., 1996, The Knowledge Management Assessment Tool: External Benchmarking Version, an American Productivity and Quality Center (APQC) white paper.
- [38] Coult, G., Smith, G., 2000, Knowledge Management: preparation and measurement - Part 1, *Managing Information*, 7/9:65–66.
- [39] Vestal, W., 2002, Measuring Knowledge Management, An American Productivity and Quality Center (APQC) white paper.
- [40] Bukowitz, W.R., Williams, R.L., 2000, *The Knowledge Management fieldbook - Revised Edition*, Prentice-Hall, London, UK.
- [41] Cooper, R.G., 2001, *Winning at New Products: Accelerating the Process from Idea to Launch*, Perseus books, Cambridge, MA.