ebbits

Enabling the business-based
Internet of Things and Services

(FP7 257852)

D 2.4 Initial Requirements Report

Document control page

Document file: D2.4_Initial Requirements Report_v1.docx
Document version: 1.0
Document owner: Andreas Zimmermann (Fraunhofer FIT)

Work package: WP 2 – Requirements Engineering and Validation
Task: T 2.2 – Initial Requirements Specification
Deliverable type: R

Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Published by the ebbits Consortium

Dissemination Level: Public

European Commission
Information Society and Media

SEVENTH FRAMEWORK PROGRAMME

Project co-funded by the European Commission within the 7th Framework Programme
Objective ICT-2009.1.3: Internet of Things and Enterprise environments

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Andreas Zimmermann (FIT)	2010-11-17	Initial Version
0.2	Andreas Zimmermann (FIT)	2010-11-18	Chapter 3 and 4 added
0.3	Andreas Zimmermann (FIT)	2010-11-19	Conclusion added
0.4	Andreas Zimmermann (FIT)	2010-11-19	Introduction and Executive Summary finished
0.5	Ferry Pramudianto (FIT) Amro Al-Akkad (FIT)	2010-11-20	Functional requirements
0.6	Amro Al-Akkad (FIT)	2010-11-24	Content and format was improved according to comments
0.7	Ferry Pramudianto (FIT)	2010-11-25	Requirements were improved according to comments
1.0		2010-11-26	Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Summary of comments
H. Udsen (In-Jet)	2010-11-23	Approved with various comments on content and format
Michael Jacobson (TNM)	2010-11-23	Approved with some correction of the functional and non-functional requirements.

Table of Content

1. Executive summary

The ebbits project aims to develop architecture, technologies and processes, which allow businesses to semantically integrate the Internet of Things into mainstream enterprise systems and support interoperable real-world, on-line end-to-end business applications. It will provide semantic resolution to the Internet of Things and hence present a new bridge between backend enterprise applications, people, services and the physical world, using information generated by tags, sensors, and other devices and performing actions on the real world. ebbits opens possibilities to offer a wide range of new business services based on orchestration of physical devices, software services, and people that are introduced as Internet of People, Thing, and Services (IoPTS).

The ebbits project has implemented a human-centred development process according to ISO 9241-210:2010. The specific challenges in developing a highly innovative technology raised the need to choose appropriate methods and implement the process in an adequate way.

Using scenarios is a powerful approach to keeping a human-centred focus throughout the application specification process, and to allow early user involvement. In iterative steps, the technical scenarios have been created, and will be validated and used to collect user feedback. The user workshops conducted were well structured and focused on context of use, benefits of support by the future ebbits platform, and user requirements.

The user requirements are based on empirical data, i.e. on real users' comments on the vision depicted by the ebbits project. The discussions within the user workshops were qualitative, explorative and meant to explore the context of use, as well as the users' values and concerns. In some cases, several users shared a concern, which can be taken as a confirmation of the issue. On the other hand, if only one user mentions an issue, this should not be dismissed merely because it was just a small number of users.

A major challenge was to aggregate the original user statements to a traceable set of more prescriptive user requirements. The Volere template proved to be useful at this step, since the results need to be documented in a way that can be communicated efficiently to the developers of the ebbits project. The condensation was done mainly by abstraction, i.e. by eliminating redundancy in clusters of equivalent statements and phrasing the essential meaning in one statement. Assigning a rationale and an appropriate classification was done in several group discussions within the Fraunhofer FIT project team.

Additionally the vision and technical scenarios themselves are one major way of communicating the context of use. Technical scenarios will become part of the specification of possible application of the ebbits platform. The impact of a set of validated technical scenarios on all project members is high. Also, scenarios are flexible and provide further inspiration, since it is vital to create visions and novel solutions.

The next major step within this human-centred development process is the determination of the impact of the requirements on each work package. In order to achieve this, the requirements need to be further condensed, prioritized and the level of quality needs to be improved. After the elimination of redundancies and a reconditioning for making the requirements referring to the ebbits platform functionality as clearly as possible, this process will result in a condensed list of functional user requirements. Thus, the total number of requirements will be reduced to a set of requirements within the scope of the ebbits project. Not all the requirements from the initial set will be addressed and implemented. The determined impact of the selected requirements on the work packages will initiate the creation of a software architecture specification draft and the technical work in the work packages.

The first iteration of the development process is finished by testing and evaluating the prototypes. The test environment will be created according to the technical scenarios that built the foundation for the first iteration. The evaluation of the prototype yields further requirements. Besides new aspects, these requirements can be considered as a refinement of the initial set of requirements presented in this document. From our experience in similar projects, the involved project partners in

WP2 expect that the refined set of user requirements will be more complete, more concrete, and more reliable.

2. Introduction

The ebbits project aims to develop architecture, technologies and processes, which allow businesses to semantically integrate the Internet of Things into mainstream enterprise systems and support interoperable real-world, on-line end-to-end business applications. It will provide semantic resolution to the Internet of Things and hence present a new bridge between backend enterprise applications, people, services and the physical world, using information generated by tags, sensors, and other devices and performing actions on the real world. ebbits opens possibilities to offer a wide range of new business services based on orchestration of physical devices, software services, and people that ebbits project introduced as Internet of People, Thing, and Services (IoPTS).

2.1 Purpose, context and scope of this deliverable

The purpose of this deliverable is to give a systematic formalization of an initial set of relevant stakeholder requirements and sub-system requirements. These requirements will guide the developments in the technical work packages, and therefore, this deliverable will be the first common source of user requirements for the ebbits consortium. The list of requirement in this document reflects the work performed in Task 2.2 – Initial Requirements Specification and emerged from two workshops conducted with representatives from the two application domains of the ebbits project: agriculture and manufacturing.

2.2 Deliverable Organization

This deliverable is organized as follows:

- Chapter 3 describes the methods and principles applied for the user-centred development of software in general.
- Chapter 4 instantiates these methods and principles for the specific properties of the ebbits project.
- Chapter 5 lists the initial set of functional and non-functional requirements for a future ebbits platform.
- Chapter 6 provides a conclusion regarding the first three months of the requirements engineering process.
- Appendix A illustrates the complete initial requirements in a table that follows the Volere template.

3. Methods and Principles of Human-Centred Development

Requirements are descriptions of how the system should behave, application domain information, constraints on the system's operation, or specifications of a system's property or attribute. This deliverable is the result of the process of requirements engineering that the ebbits project has started. Requirements engineering is a continuous iterative process driven by an adopted user-centred design (UCD) approach and not a stage or phase in the way. An incomplete requirements analysis tends to lead to problem descriptions later in the system development. This is why it is important to continue the user-centred design process outlined in this document. As a consequence, this document should be considered as a first initial version of the requirements that will be the basis for updated and changed requirement reports as new requirements arise or outdated disappear in the iterations of the project.

The general approach to requirements gathering involves following activities in the ebbits project:

- *Elicitation. Discovering, extracting and learning needs of stakeholders. It includes a domain analysis that helps to identify problems and deficiencies in existing systems, opportunities and general objectives. Scenarios are part of this activity.*
- *Modelling. Creating models and requirements, looking for good understanding of them and trying to avoid incompleteness and inconsistency.*
- *Negotiation and agreement. To establish priorities and to determine the subset of requirements that will be included for the next phase.*
- *Specification. Requirements expressed in a more precise way, sometimes as a documentation of the external behaviour of the system.*
- *Verification/Validation. Determining the consistency, completeness and suitability of the requirements. It could be done by means of static testing (using regular reviews, walkthroughs or other techniques) and prototyping.*
- *Evolution and management. The requirements are modified to include corrections and to answer to environmental changes or new objectives. It is important to ensure that requirement changes do not produce a large impact on other requirements. Requirement management means to face those modifications properly, to plan requirement identification and to ensure traceability (source, requirements and design traceability).*

It is important to underline that most of those activities are performed in parallel.

3.1 ISO 9241-210 Standard

The ISO 9241-210(ISO, 2010) "Ergonomics of human-system interaction" gives guidance on human-centred design activities throughout the life cycle of computer-based interactive systems.

Essential principles in this process are

- *Multi-disciplinary design*
- *Iteration of design solutions*
- *Appropriate allocation of function between developer-user and technology*
- *Active involvement of users and a clear understanding of user and tasks requirements*

The multi-disciplinary design is given by the expertise in ebbits, which includes psychologists, computer scientists, and usability engineers and designers. The iteration of solutions is implemented in the ebbits work plan.

Human-centred approach implies an iterative life cycle in a project. A system is perceived as a socio-technical system, i.e. the novel technology is a fit between a technical system and its usage. Scenarios are part of the system specification; they explicitly deal with the usage of a technical system, the context of use, and the allocation of function between the technical system and human users. Later, when a prototype is available, users can try it out and gain personal experience with the system. Iterative cycles allow advancing from specification to implemented prototypes, from experience and evaluation to improved specifications and improved prototypes. In ebbits four cycles are planned for the project lifetime, aiming at validated prototype specifications, including concepts of usage.

The active involvement of users and clear understanding of user and task requirements is a challenge in projects like ebbits for three reasons:

First, the user groups in the case of ebbits are only indirectly addressed by the vision scenarios.

Second, the potential user groups are not known a priori but need to be identified according to the vision scenarios; this needs to be revised as the objectives of the visions evolve, there may be various groups of potentially affected users.

Third, the visions of ebbits are far-sighted and not close to users' current experiences; therefore users may not be confident and precise about their needs concerning this future middleware.

One of the core tasks of user-centred design is to negotiate and facilitate the communication across the well-known user-developer gap while acknowledging the different forms of expression and different requirements on each side. The literature has a lot of examples demonstrating that end users have to bridge the large gap in understanding especially in projects that apply a waterfall model. Clark, Lobsitz & Shields, (1989) show that evolutionary or iterative approaches drastically reduce this gap.

The user-centred design process reflects an iterative process with no sharp start and end points: eliciting the 'context of use' requires intensive user involvement continuously for the whole duration of the process, and the requirements elicitation likewise extends well into the design proposal phase. There are four essential human-centred activities recommended by the ISO standard (ISO-9241-210):

1. *to understand and specify the context of use*
2. *to specify the organizational and user requirements*
3. *to produce design solutions*
4. *to evaluate design regarding requirements*

The human-centred design approach implies an iterative life cycle in a project. Iterative cycles allow advancing from specification to implemented prototypes, from experience and evaluation to improved specifications and improved prototypes. A system is perceived as a socio-technical system, i.e. the novel technology is a fit between a technical system and its usage (Emery &Trist., 1960). The design proposals are based on the current understanding of the context of use. These proposals provide an idea on how to meet identified or assumed requirements. The evaluations of the design proposals yield a richer understanding of the context of use and new or modified requirements and thus guide the evolutionary improvement of the design.

3.2 The Volere Schema

The ISO standard does not prescribe specific methods to achieve these goals; they are to be chosen according to the current state of the art and what is appropriate under individual project circumstances. Based on practical experiences from other R&D projects, we have devised a scenario-based approach, combined with user workshops and expert analysis, based on the structure proposed by Robertson & Robertson (1999) for mastering requirements.

The Volere process recommended by Robertson & Robertson ensures that all important aspects of requirements are carefully addressed and that the methods applied have proven their value in practical work. The details of the applied process within ebbits are explained in the subsequent Chapter 4 and the important aspects of the requirement description according the Volere schema are addressed in Section 4.5. The distinction between global constraints affecting the project, functional requirements and non-functional requirements, with a fine-grained distinction of different types, as well as the categorisations of the Volere template, with the need to define fit criteria and a rationale for each requirement and the evaluation of customer satisfaction and dissatisfaction whether or not the requirement is implemented has proven to be of great practical value. The philosophy of Robertson & Robertson is very much in line with ISO 9241-210 and allows a structured processing of the requirements assuring that they remain always applicable and testable.

3.3 Sources for the Derivation of Requirements

The requirement derivation process has to be founded on specific sources for information. The two classical sources are scenarios and field studies, which consist of interviews and ethnographical methods like participatory observations of the domain context and experimental testing of existing solutions.

3.3.1 Scenario Discussion

Scenarios have proven their potential to communicate project goals and design solutions among all stakeholders and are widely used to discover and understand users' goals and system requirements. Scenarios can be used at all stages of a project, and for various purposes. In particular, scenarios are the first tangible artefact a project can possibly produce, and are therefore suited to start user involvement very early. Scenarios can capture and illustrate features of a system, modes of its usage, and the benefits for users. Scenarios can be written at several levels of detail, they can tell about the current as well as future states of a socio-technical system. Scenarios can focus on normal usage, but can also be used to explore critical cases, limitations or even catastrophes. Scenarios are useful to support discussion among project team members as well as with prospective developer-users. Scenarios can also be seen as part of the documentation and specification of a system. There is huge amount of literature concerning scenario-based approaches (see: Carroll (2000), Sutcliffe (2003), Weidenhaupt et al. (1998) and Dzida et al. (1999)).

From the scenarios and storylines, a systematic formalisation of all relevant user requirements and subsystems requirements will be derived.

The basis will be user-centric requirements originating from the ecosystems of heterogeneous stakeholders. These include functional requirements, energy requirements, business requirements and trust, privacy and security requirements. The non-functional societal requirements will include requirements related to ethics, inclusion and data protection, quality of use, professional liability, IPR issues, legal and regulatory needs, etc.

3.3.2 User Workshops

The workshops aimed at acquiring feedback regarding new ideas and providing invaluable information about the potential market acceptance of the product or the idea. The workshops established an interactive group setting, where participants are asked questions by a moderator and are free to discuss with other group members. They enable the collection of information about the working environment, which includes a survey of the existing IT infrastructure, and thus, the identification of preliminary requirements and restrictions from the ability to interface with these systems.

Especially if a workshop is set up to discuss new not yet realized ideas or setting, scenarios can be introduced to make it possible for the participants to imagine the new system. The scenarios help capturing and illustrating the features of the new system, its modes of usage, and its benefits. The result of focus groups in a requirement gathering process is a set of statements that have to be converted into requirements. The details of the applied process within ebbits are explained in the subsequent chapters. The outcome of these workshops can be documented by taking direct notes.

3.3.3 Focus Groups

A focus group is a qualitative research method using a group of experts, in this case developers, to find out their attitude towards a new idea, a product, service, or concept. Focus groups are an important tool for acquiring feedback regarding new ideas. In particular, focus groups allow discussing, and testing of a new product before it is made available to the public. This can provide invaluable information about the potential market acceptance of the product or the idea.

A focus group establishes an interactive group setting, where participants are asked questions by one or two moderators and are free to discuss with other group members. Apart from the discussion in the focus groups the participants can be observed. Observations tend to reveal more details compared to interviews. They enable the collection of information about the working environment, which includes a survey of the existing IT infrastructure, and thus, the identification of preliminary requirements and restrictions that exist for interfacing with these systems.

The combination of observation and interviews has proven its potential to deliver the best insight into understanding the processes among all stakeholders. Observation is of great value especially in the initial stages of a project. It is often much easier to observe users in their domain context and their activities than to ask them in interviews.

Interviews can be used at all stages of a project, and for various purposes. During their work it is more convenient for users to explain what they are actually doing and why. In order to create a more comfortable and conversational situation, it is common practice to perform semi-structured interviews. These interviews loosely follow specific guidelines and a list of questions, but leave room for a spontaneous adaptation of the progress and development of the conversation.

The documentation of this information is done either by taking direct notes or, if all details must be captured, by videotaping the participants while using of the system and capturing the environment by taking pictures or painting images.

In general, focus groups support the collection of ideas about how to optimize the use of the new technology. Focus groups are low in cost, one can get results relatively quickly, and they can increase the sample size of a report by discussing with several people at once.

The initial requirements that we obtained are the result of two one-day focus group workshops that we have conducted in TNM and COMAU premises.

4. User-Centred Design Procedure in the ebbits Project

An essential property of the UCD approach is that it has to be adapted to the specific requirements of the individual project. This chapter gives an overview on how the standard procedure has been instantiated and adapted to the ebbits project.

4.1 Initial vision scenario

A scenario is an acknowledged way of communicating the vision of a particular system, as well as to explain and document requirements. Deliverable 2.1 "Scenarios for Usage of the ebbits Platform" documents the work undertaken in task T2.1 "Scenario Thinking" and provides top-level user requirements in the form of vision scenarios of future use of the ebbits platform in the two selected domains. The next step produces technically oriented scenarios focusing on the deployment and use of the ebbits platform. These scenarios address technical questions referring to the platform and its components. The deliverable also summarises the future developments foreseen by the experts and the significance this will have for the development of the two ebbits applications in 2015 and beyond. The future scenarios describe end user activities as well as application functionalities, and thus, bridge the gap to the formulation of technical user requirements.

Creating scenarios of end user behaviour and interaction with platform functionality is an extremely useful instrument for identifying key technological, security, socioeconomic and business drivers for future end user requirements. The scenarios provide a vision framework for the subsequent iterative requirement engineering phase.

4.2 Derivation of Technical Scenarios

From the main vision scenario more technical scenarios have been derived that could be used to elicit requirements for the future ebbits platform. Following the UCD cycle, these scenarios will be made more specific in a next step. These technical scenarios were tentative, trying to capture the context of use for a certain user role and to illustrate how the ebbits platform might support them. Such technical context scenarios illustrate the benefits and functionality of a system for certain user groups with their typical tasks and goals (see Dzida, 1999). They describe the users' view of the usage of a system within the current context of work and the envisaged improvement of tasks. Scenarios normally do not explicate details of interaction, which is left for a later stage when mock-ups are available. These technical scenarios were elaborated for different work of context aspects and specific to the problems and questions within all technical work packages.

It is important to note that the technical scenarios were meant as means for discussion with users! The scenarios do not contain requirements, but help the users and experts generate the requirements.

4.3 User Workshops

A combined Vision Scenario/User workshop for the automotive manufacturing domain took place in Torino, Italy, on October 19, 2010. The workshop was hosted by COMAU and moderated by IN-JET. The workshop covered two main manufacturing areas, Power Train Machining & Assembly (PWT) and Body Welding & Assembly (BWA). For each area presentations describing the business and the controls architecture were given, during and after which the participants had the opportunity to ask and answer questions. The partners also shared their visions and ideas for the future use of the ebbits platform, thus providing the basis for the scenarios in D2-1 Scenarios for Usage of the ebbits Platform. These visions combined with the conclusions from the Q&A sessions provided the Developer Partners with the foundation for eliciting the technical requirements.

Another combined workshop for the food manufacturing domain took place in Copenhagen, Denmark, on October 1, 2010. The workshop was hosted by TNM and moderated by IN-JET. The workshop covered four main topics, feedstuff manufacturing, livestock farming, slaughtering and retailing. For each area presentations describing the business and the controls architecture were given, during and after which the participants had the opportunity to ask and answer questions. With particular emphasis on traceability the external experts provided us with their visions for a future ebbits platform. As for the automotive manufacturing workshop, these visions together with the Q&A sessions gave the Developer Partners the foundation for eliciting the technical requirements.

The purpose of the user workshops was to get insight into roles, processes, exceptions, problems and stakeholders involved in the operation of their technology. Apart from questions concerning context of use, the major part of the discussion focused on the features of the future ebbits platform. Additionally, we wanted to gather requirements in a first step and to be able to elaborate the technical scenarios. During the interview and discussion phase, the users were asked what support they would want to have and what their requirements would be in order to construct own systems that best fits their needs. This way, the users could express ideas as to how the ebbits platform should look.

4.4 Requirements derivation

The main task after the completion of workshops and interviews was the consolidation of the information gathered from the discussions. The output of the discussions and interviews are user statements. The analysis of the original users' statements in the respective workshops led to the elicitation of requirements at different levels of detail and their aggregation in a structured way.

First, all user statements were extracted from the discussion protocols and statements referring to the more global constraints were separated. These statements, though not referring to the ebbits project, contain valuable information about context of use and have been archived.

The second step was the first classification of the remaining statements relating explicitly or implicitly to the support the ebbits platform might provide. Such user feedback to technical scenarios may relate to various aspects of the system and its use, and have been classified according the Volere scheme (see Robertson and Robertson, 1999).

Functional requirements give the specification of the product's functionality, derived from the fundamental purpose of the product, whereas non-functional requirements are the properties of the product, the qualities and characteristics that make the product attractive, usable, fast or reliable. Non-functional requirements can be grouped according to following subcategories:

- *Look and feel requirements (intended appearance for end users)*
- *Usability requirements (based on the intended end users)*
- *Performance requirements (how fast, big, accurate, safe, reliable...)*
- *Operational requirements (what is the intended operating environment?)*
- *Maintainability and portability requirements (how changeable it must be)*
- *Security requirements (security, confidentiality and integrity)*
- *Cultural and political requirements (human factors)*
- *Legal requirements (conformance to applicable laws)*

Of course look and feel, usability and cultural requirements are of only secondary relevance for the assessment of requirements for a software platform, but are nevertheless important for the assessment of qualities and aspects of the user interfaces to be developed. The current set of user requirements can be found in Chapter 4 of this deliverable and thus has become accessible for all users and also traceable for evaluation of design solutions.

4.5 Requirement description (Volere Schema)

The workflow to ensure that all necessary details and procedures in the Volere schema are adhered to is rather complex, and it was decided to support this process with a tool for all partners within the project.

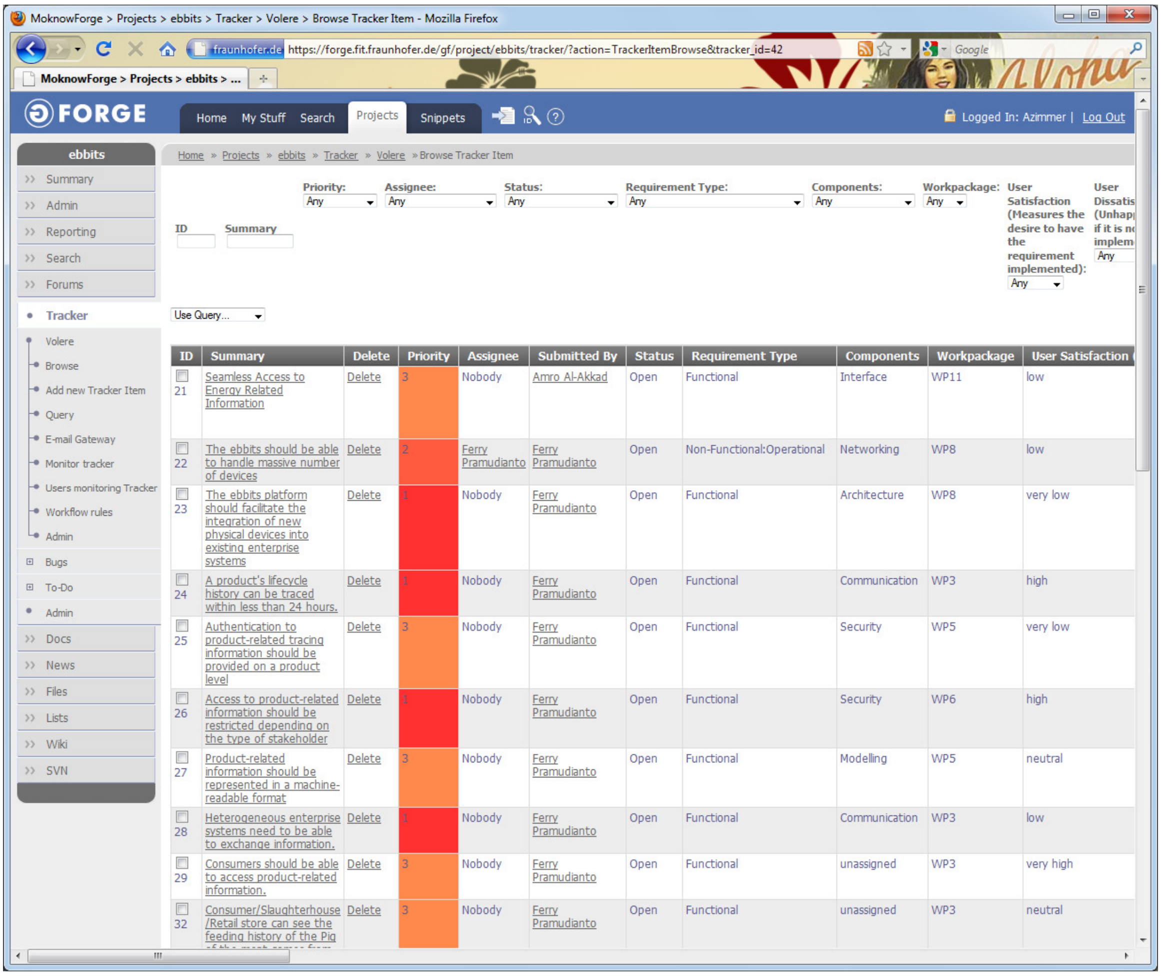

Figure 1: Screenshot of GForge with a list of requirements

We decided to use GForge, which is a web based bug tracker that allows implementing and tracking the workflow of the Volere schema. Figure 1 shows a screenshot of GForge with a list of open requirements. The requirements were all entered into GForge, which ensured that always two persons controlled the quality of the requirements. One person entered the requirement into GForge, whereas another person controlled and passed the requirement through the quality gateway. You can see the roles of "Submitter" and "Assignee" in the screenshot in Figure 2. The most important fields grouping the requirements according to the Volere schema are described in the following:

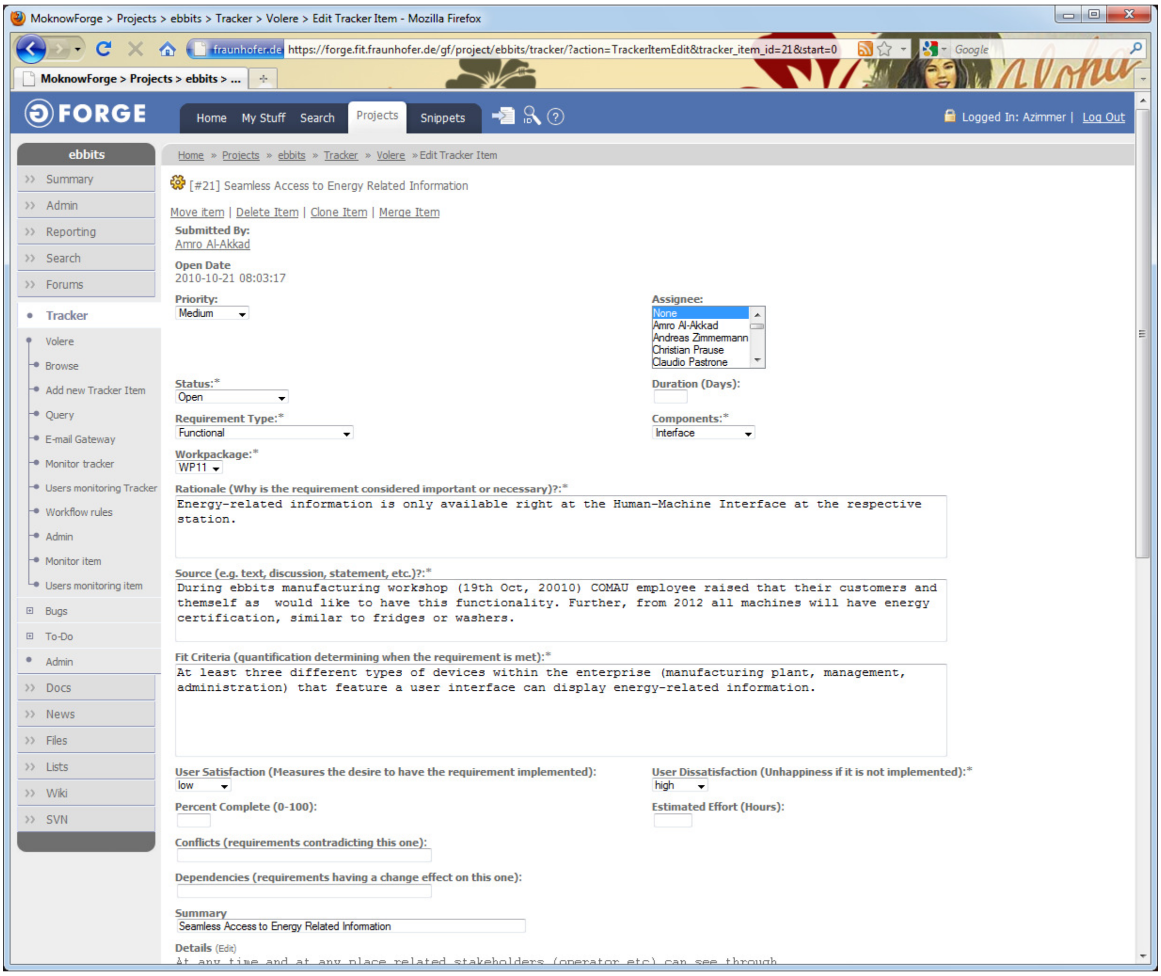

Figure 2: Screenshot of GForge with a requirement in the edit mode

In an internal workshop at Fraunhofer FIT it was decided to structure the process of requirement gathering and evolving several high level aspects, such as assigning components to a requirement:

- *Architecture*
- *Communication*
- *Configurability*
- *Context*
- *Device Discovery*
- *Device*
- *IDE*
- *Interfaces*
- *Modelling*
- *Networking*
- *Security*
- *SDK*
- *Service Discovery*

The requirements gathered affect one or more of those aspects and functions of the system, and furthermore they were collected within different work packages. Therefore it was decided to maintain this information in order to structure the requirements. Each requirement has a reference to the work package it relates to and to the component of the future ebbits platform or the respective quality or attribute of a component that it describes. It is obvious that this helps structuring the requirements. The screenshot aboveshows the respective input fields in GForge.

The requirements were prioritized. Priority is a very important field that defines the relevance of this requirement in relation to the other requirements. It allows classification of the specified requirement in five categories: "Blocker", "Critical", "Major", "Minor", and "Trivial". The rating was carefully assigned and was the last step of the requirement specification before it passes the quality check. The *priority* of a requirement is based on several important aspects included in the Volere schema:

- *The source defining if this requirement was raised by primary or secondary stakeholders [the latter less important], or by discussion within the consortium, by vision and technical scenarios or the DOW.*
- *The assessment of customer satisfaction and dissatisfaction if this requirement is achieved respectively missed.*
- *The estimation if the requirement is within the scope of the project.*
- *The component that the requirement is associated to.*

The *summary* of a requirement contains a one-sentence description of the requirement. The description is the intent of the requirement and should be clear and brief.

The *rationale* of a requirement expresses the reason behind the requirement's existence. The rationale provides the reason why the requirement is important and the contribution it makes to the product's purpose. The rationale contributes to the understanding of the requirement.

The *Fit Criteria* is the most important field. Fit criteria are the quantified goals that the solution (i.e. the realization of the requirement) has to meet. This field describes how to determine if the requirement is met. It should be written in a precise quantified manner. The fit criterion sets the standards to which the developer constructs the product.

In order to express *dependencies* and *conflicts* among requirements, GForge allows the definition of links between two requirements.

4.6 The Requirements Workflow

One part of the requirements process is the quality control of all requirements. People who perform the quality control must have the possibility to decide, which requirements will become part of the specification and which must be revised. In the latter case, feedback with the submitter of that requirement is necessary. The quality control in GForge is realised by processing requirements along the steps of a workflow. Each requirement has a status that changes depending on the current workflow step. Figure 3 displays the possible status and accordant transitions between them.

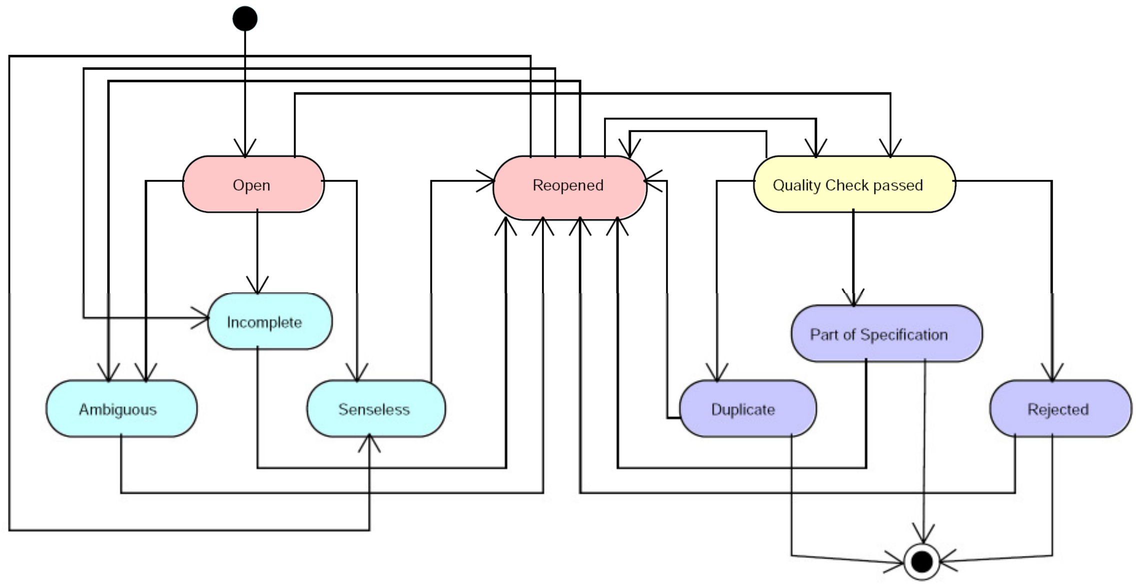

Figure 3: Structure of the Requirements Workflow

When a requirement is entered it is assigned the status 'open'. If it is complete and unambiguous, it passes the quality check. In this case it is important, that not only the text fields are filled in sensibly, but also appropriate values are chosen from the drop-down lists. The priority must be selected to make it possible to rank requirements among each other.

A requirement can fail to pass the quality gateway for three reasons:

1. A requirement can be incomplete. Some fields may have meaningless entries like '?'
2. A requirement can be ambiguous; certain terms are not clearly specified
3. A requirement is completely senseless. This can happen for example when a requirement is entered into the system for testing.

If a requirement fails the quality check, it gets one of the statuses 'Requirement is incomplete', 'Requirement is ambiguous' or 'Requirement does not make sense'. Once the requirement is updated properly, its status is changed to 'reopened'. This status equals exactly the initial status 'open' and the quality check process restarts. The status 'reopened' is used to indicate that a requirement went through the quality control at least once. This helps to detect requirements that are yet untouched.

Eventually, all requirements will pass the quality gateway. That means that they are complete and all fields are filled in sensibly. A requirement that passed the quality gateway cannot be edited any more. The last step is to decide whether a requirement becomes part of the specification, or whether it is not considered any longer. This can happen for two reasons:

1. A requirement can be a duplicate of another requirement
2. A requirement can be rejected e.g. because it is out of the project's scope.

If its status is either 'Rejected', 'Part of specification' or 'Duplicate', a requirement is said to be resolved.

5. Requirements for the ebbits Platform

This section contains the condensed list of functional and non-functional user requirements, extracted from the original list of user statements based on the initial vision scenarios for the farm and manufacturing domain. The aim of this approach is to provide a simple and structured representation of requirements, to be used as a reference for the development of the first pilot applications. The list of requirements will be updated during the project lifetime, as soon as the need for new features is identified. We will apply various methods to improve our understanding of the user needs and to improve the user-perceived qualities of the prototypes. In particular, we will review the user requirements during the evaluation of the first application prototypes in order to get the second, improved set of user requirements (see for example Schmidt-Belz et al., 1999).

Each requirement listed in the following tables obtains a unique ID to refer to. The description of a requirement is a synthetic but clear description of the requirement. The rationale gives a reason why this requirement is relevant for the system and thus has been included into the table. The column source gives an indication of where to find the requirement, i.e. scenario, interview or user workshop. According to the Volere scheme the requirements are divided into non-functional and functional requirements.

5.1 Functional Requirements

User statements that explicitly refer to the functionality of the future ebbits platform are called functional user requirements.

List of functional requirements

Architecture:

- The ebbits platform should facilitate the integration of new physical devices into existing enterprise systems
- Flexible Integration of HW/SW components
- Aggregating collected sensor data at a central point
- Aggregate data from various data bases and sources

Communication:

- A product's lifecycle history can be traced within less than 24 hours.
- Heterogeneous enterprise systems need to be able to exchange information.
- Controlling of machines/stations in manufacturing plant remotely
- Interoperability needs to be created between various subsystems in the manufacturing area
- reduce paper based communication
- back tracing production problem from complaints
- producers can push notification of recalled products to costumers
- automatic analysis of cross enterprises product life cycle data
- notification through out the chain
- Transfering IDs cross enterprises
- ebbits should bridge communication between different applications in farms
- Information needs to be described in a standardised way
- The ebbits platform should amplify branding for enterprises.
- Alarms are send when specific situations occur

Context

- Stakeholders should be able to access product-related information instantly.
- Retrieve manufacturing data history of any relevant event during production
- integration of mobile sensing devices on running infrastructure manufacturing plant without interrupting running processes
- Access to energy-related information from production machines needs to be provided.
- Slaughter house needs to know how many pigs they will get from farmers
- products rating by experts
- Loggiing of Quality related informaation of each Manufacturing Part
- supplier can predict when to make the next delivery to a consumer
- Predict if his suppliers will not be able to fulfill their demands (quantity)
- system should aware of building condition in each farm
- feeding systems should aware of the animals weight
- system should provide location tracking of the stocks/livestocks
- System should show Energy Cost for different granularity of production processes
- Adjust production processes according to energy price policies
- filter/fusion information for each operational process
- early maintenance notification when needed
- automatic calibration
- support system for comparing different energy consumption among plants and corresponding processes
- Summary of energy related information at operational level for supporting management level optimizing energy use
- recognition of energy wasting behaviors
- Items need to be traced within an enterprise
- Items need to be traced between enterprise
- Associate meta-information to items
- Retrieve the behavior on an individual animal level
- Synchronisation of Acquired Data is necessary
- The system should be self-ware

Device

- Devices save historical information in farms
- Each sow carry an unique ID
- Diagnostic component to detect and correct mal functions
- 3 Measurement Points for every station in body welding
- automatic start up synchronization among machines
- Device and service exception handling
- Report errors in devices

Interfaces

- Seamless Access to Energy Related Information
- Seamless data collection
- officials have a back door access to highly important information
- ebbits should support legacy network interfaces
- Interfacing with Programmable Logic Controller of production robots
- Item identification system should provide open interfaces to other systems
- Different views on the device ontology
- Support for interfacing with external workflow systems
- Different Views on the Data is necessary
- Integration of legacy systems into ebbits platform

Modelling

- Product-related information should be represented in a machine-readable format
- semantic relationships between data
- farmers need to save and able to reflect breedings history information
- correlate founded problem with production batches
- Definition of smallest unit can be traced or uniquely identified
- Batches need to be identified on a farm level
- A standard way of numbering batches needs to be provided
- Support fuzzy or probability concepts for reasoning
- Handling of different device versions in device ontology
- Download and harmonisation of third party device ontologies
- waste of energy act definitions
- Common structure of information is needed
- The system should allow the correlation of information emerging from several sources
- Store meta-information with package labels

Networking

- Reliable wireless solution for new sensors
- bring data from fieldbus network to ethernet network

Security

- Authentication to product-related tracing information should be provided on a product level
- Protection to sensitive information
- Protection of System Integrity
- Access-control of data sets

5.2 Non-functional Requirements

Non-functional requirements address the quality of the future system and are classified by various criteria according the Volere schema (Usability, performance, operational requirements, maintainability, etc.).

List of non-functional requirements

Operational

- Hazardous Environmental Monitoring of Manufacturing Plant
- The ebbits should be able to handle massive number of devices
- Affordable tagging/tracking system for pigs
- Resilience and adaptable to environment condition changes
- The network infrastructure needs to have self-configuration capabilities
- New products should be networked with mainstream enterprise systems easily and cost-efficiently.
- Improve air compression energy usage
- location tracking should be implemented as independent app
- reliability of the system should be more than 30 years
- hardware components are able to handle harsh condition
- central point to start the whole plant
- reduce water consumption in PWT
- Support profiling device performance
- Mobile management of farms
- Applied stand-alone devices should have a long battery life span
- Electronic exchange of reports between enterprises

Performance

- Higher bandwidth and range of wireless connection
- Systems built using HYDRA should be scalable in terms of devices communicating

Usability

- Configurable and easy to install middleware
- End-users need to be able to management their distributed data
- Filtering to Obtain relevant Information
- Display plant activities in real-time

Maintainability

- Comply with industrial standards
- Life-cycle of a robot and its components is traceable
- Scalable solution (scale up and scale down)

Security

- Meta-information associated with entities needs to be trustworthy

6. Conclusion

The ebbits project has implemented a human-centred development process according to ISO 9241-210. The specific challenges in developing a highly innovative technology raised the need to choose appropriate methods and implement the process in an adequate way.

Using scenarios is a powerful approach to keeping a human-centred focus throughout the application specification process, and to allow early user involvement. In iterative steps, the technical scenarios have been created, and will be validated and used to collect user feedback. The user workshops conducted were well structured and focused on context of use, benefits of support by the future ebbits platform, and user requirements.

The user requirements are based on empirical data, i.e. on real users' comments on the vision depicted by the ebbits project. The discussions within the user workshops were qualitative, explorative and meant to explore the context of use, as well as the users' values and concerns. In some cases, several users shared a concern, which can be taken as a confirmation of the issue. On the other hand, if only one user mentions an issue, this should not be dismissed merely because it was just a small number of users.

A major challenge was to aggregate the original user statements to a traceable set of more prescriptive user requirements. The Volere template proved to be useful for this step, since the results need to be documented in a way that can be communicated efficiently to the developers of the ebbits project. The condensation was done mainly by abstraction, i.e. by eliminating redundancy in clusters of equivalent statements and phrasing the essential meaning in one statement. Assigning a rationale and an appropriate classification was done in several group discussions within the Fraunhofer FIT project team.

Additionally the vision and technical scenarios themselves are one major way of communicating the context of use. Technical scenarios will become part of the specification of possible application of the ebbits platform. The impact of a set of validated technical scenarios on all project members is high. Also, scenarios are flexible and provide further inspiration, since it is vital to create visions and novel solutions.

The next major step within this human-centred development process is the determination of the impact of the requirements on each work package. In order to achieve this, the requirements need to be further condensed and the level of quality needs to be improved. The total number of requirements will be reduced to a set of requirements within the scope of the ebbits project. Not all the requirements from the initial set will be addressed and implemented. The determined impact of the selected requirements on the work packages will initiate the creation of a software architecture specification draft and the technical work in the work packages.

The first iteration of the development process is finished by testing and evaluating the prototypes. The test environment will be created according to the technical scenarios that built the foundation for the first iteration. The evaluation of the prototype yields further requirements. Besides new aspects, these requirements can be considered a refinement of the initial set of requirements presented in this document. From our experience in similar projects, we expect that the refined set of user requirements will be more complete, more concrete, and more reliable.

7. References

Carroll, J. M. (2000): Making Use: scenario-based design of human-computer interactions. MIT Press.

Clark, Peter G. Lobsitz, Richard M. & Shields, James D.(1989): Documenting the evolution of an information system. IEEE Software, pp. 1819-1826.

Dzida, W. (1999): Developing Scenario-based Requirements and Testing them for Minimum Quality. HCI, Erlbaum.

Dzida, W.; Freitag, R. (1998): Making Use of Scenarios for Validating Analysis and Design. IEEE Transactions on Software Engineering 24(12), pp. 1182-1196.

Emery, F.E.; Trist, E.L. (1960): Socio-Technical Systems. In: Management Sciences, Models and Techniques, Vol. 2, London.

Hickey, A.M.; Dean, D. L.; Nunamaker, J. F. (1999): Establishing a Foundation for Collaborative Scenario Elicitation. The DATA BASE for Advances in Information Systems. 30, 3-4, pp. 92-110.

Hickey, A.M.; Dean, D. L. (1998): Prototyping for Requirements Elicitation and Validation: A Participative Prototype Evaluation Methodology. Proceedings of the 1998 Americas Conference on Information Systems, Baltimore MD, pp. 798-800.

ISO (2010) 9241-210:2010 Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems

Robertson, S.; Robertson, J.R. (1999): Mastering the requirement process. Addison Wesley, London, ACM Press Books.

Schmidt-Belz, B.; D. Fleischhauer (1999): Scenario-based System Validation by Users. Human-Computer Interaction - INTERACT '99, Edinburgh, Swindon: British Computer Society.

Sutcliffe, A. (2003): Scenario-based Requirements Engineering. 11th IEEE International Requirements Engineering Conference (RE '03), Monterey Bay, California, US.

Weidenhaupt, K.; Pohl, K. (1998): Scenario usage in System Development: a Report on Current Practice. IEEE Software 15(2), pp. 34-45.

List of Figures

8. Appendix A - Complete Requirements

8.1 Functional Requirements

No	Summary	Rationale	Source	Fit Criteria	Priority	Requirement Type	Components	User Satisfaction	User Dissatisfaction
1	Seamless Access to Energy Related Information	Energy-related information is only available right at the Human-Machine Interface at the respective station.	During COMAU workshop (19th Oct, 20010)	At least three different types of devices within the enterprise (manufacturing plant, management, administration) that feature a user interface can display energy-related information.	3	Functional	Interface	low	high
3	The ebbits platform should facilitate the integration of new physical devices into existing enterprise systems	Enterprises that already have a running ebbits system may need to add new devices.	TNM scenario workshop, issued by Jesper	ebbits provides a plug 'n' play framework for the integration of new devices into existing running systems.	1	Functional	Architecture	high	very high
4	A product's lifecycle history can be traced within less than 24 hours.	EU Regulation in the future will obligate enterprises to be able to produce a requested information within 24 hours.	TNM scenario workshop in Copenhagen	A product lifecycle cross Enterprises can be traced within less than 24 hours.	1	Functional	Communication	high	high

5	Authentication to product-related tracing information should be provided on a product level	Enterprises want to share valuable product-related information, if there is a business value for them, and they want to be able to grant access to this information on a product level.	TNM scenario workshop in Copenhagen, issued by Jesper	Only authorized stakeholders can see the product's information.	3	Functional	Security	very low	very high
6	Product-related information should be represented in a machine-readable format	Automatic processing requires that machines can understand and process information	TNM scenario workshop in Copenhagen	Machines can process information of a product automatically.	3	Functional	Modelling	neutral	very high
7	Heterogeneous enterprise systems need to be able to exchange information.	Enterprises use various information systems that need to exchange information. Information needs to be propagated throughout the chain.	TNM scenario workshop in Copenhagen.	At least three different enterprise systems can exchange information.	1	Functional	Communication	low	very high
8	Stakeholders should be able to access product-related information instantly.	Farms want to able to sell their "brand" to the costumer, which creates a good business case to motivate farmers to participate in sharing information. A certificate of origin would be good for the customers. There exists also a safety issue: when the end-user is allergic to something, we can provide information about the contents of the beef.	TNM scenario workshop in Copenhagen.	For each meat product, stakeholders can access the name of the producing farm, the millage, Feed stuff, medicine and the CO2 foodprint. stakeholders e.g.: meat quality control of slaughterhouse and retail store, end consumers	2	Functional	Context	very high	low
9	Each sow carry an unique ID	if sows are let loose, farmers need to identify sows uniquely because each sow produces piglets with different quality and productivity.	Thomas N.M. during traceability workshop.	Farmers can identified sows uniquely if they are on the loose.	3	Functional	Devices	low	high

11	Controlling of machines/stations in manufacturing plant remotely	To optimize production process.	COMAU workshop in Turino (19th Oct, 20010) employee (Fulvio) raised this issue.	Relevant stations that operate automatically can be started/stopped via remote calls.	3	Functional	Communication	low	high
13	Flexible Integration of HW/SW components	The process to set-up a plant and to fine-tune each machine costs at least a day. In this sense, the integration of new HW/SW must be very easy to not disturb the overall process.	COMAU workshop in Turino (19th Oct, 20010) (Fulvio) that they need to have this functionality for deployment.	In plug'n'play manner new HW/SW components can be easily integrated into an existing system without loosing much time.	2	Functional	Architecture	low	high
14	Retrieve manufacturing data history of any relevant event during production	If production defects are recognized, it is helpful to look at the production process history in order to find out what caused the defects.	COMAU workshop in Turino (19th Oct, 20010) employee raised this issue.	Any manufacturing relevant (pressure, energy consumption, temperature, humidity, time etc) data is retrievable.	1	Functional	Context	very low	very high
16	integration of mobile sensing devices on running infrastructure manufacturing plant without interrupting running processes	In brown field it is too risky or too expensive to stop production in order to install missing sensors (e.g. smart meter).	COMAU workshop in Turino (19th Oct, 20010) employee raised this issue.	It is possible to enhance a station/machine/robot with mobile sensing devices to gather data.	2	Functional	Context	very high	very low

17	semantic relationships between data	Currently, any data is stored in a simple database. Hence, data is availabe, but cannot be interrelated intelligently.	COMAU workshop in Turino (19th Oct, 20010) employee raised this issue.	Data can be queried and inferred in order.	1	Functional	Modelling	very low	very high
18	Aggregating collected sensor data at a central point	The aggregation of collected data is important for analyzing the data.	TNM said that they currently can obtain different sensor data, though the aggregation is missing.	A framework is provided that aggregates collected sensor data at a central point of an application.	1	Functional	Architecture	very high	low
19	Farmers are able to retrieve optimized models from research	Farmers are willing to share data if they could get something in return such as models to optimize feeding process.	TNM workshop	Farmers can get optimized models electronically.	4	Functional	unassigned	high	low
20	System can feed the farms data to research	Most of the farming models are developed by research organizations, universities etc.	TNM workshop (Thomas)	Researchers are able to get their hands on life data on farms.	4	Functional	unassigned	high	low
23	Access to energy-related information from production machines needs to be provided.	Energy-related information is measured by some of the operational machines (e.g. in the production plant), but it is not distributed into a network.	COMAU scenario workshop (10/19/2010).	If any machine provides access to energy-related information, ebbits distributes this information to all interested parties.	2	Functional	Context	neutral	very high

26	Interoperability needs to be created between various subsystems in the manufacturing area	The subsystems in manufactoring environments are currently not interconnected and not able to exchange information.	TNM scenario workshop in Copenhagen.	Three independent subsystems from the manufacturing area can exchange information (show a use case)	2	Functional	Communication	neutral	high
28	Slaughter house needs to know how many pigs they will get from farmers	slaughter house needs to know the flow of the pigs coming to them so that they can plan and balance supply from farmers and demands from retail store.	TNM Workshop Copenhagen	slaughter house can forecast how many pigs are going to be supplied by farmers.	3	Functional	Context	high	high
29	reduce paper based communication	paper based communication between enterprises takes time and efforts for inputing data to the system	TNM Workshop Copenhagen	50% Reduction of current paper based communication.	1	Functional	Communication	neutral	very high
30	farmers need to save and able to reflect breedings history information	breeders combinations produce different quality of piglets	TNM Workshop Copenhagen	farmers can at least trace breeders combinations that produce unhealthy piglets.	2	Functional	Modelling	very high	high
31	products rating by experts	customer satisfaction is an important factor in business area.	TNM Workshop Copenhagen	experts are able to give rating to products	1	Functional	Context	neutral	very high
32	back tracing production problem from complaints	the source of the problem during production need to be localized and used for repairing recalled products.	TNM Workshop in Copenhagen	50% of time reduction to localize the source of the problem. (no determinant time is possible because it depends on how complex the production processes are)	1	Functional	Communication	high	high

35	Seamless data collection	Data collection is the required input for simple and complex analysis in both manufacturing and traceability scenario.	COMAU workshop in Turino (19th Oct, 20010) employee raised this issue.	Both mobile or static sensors are affixed to any medium (animal, robot etc) in order to sense the environment.	1	Functional	Interface	very low	very high
36	Diagnostic component to detect and correct mal functions	If a male function has slipped in the plant it should be corrected asap. In fact, if possible any fault behavior should be prevented at all.	COMAU workshop in Turino (19th Oct, 20010) employee raised this issue.	Male functions or strange behaviour of machinery are recognized early enough.	3	Functional	Devices	high	neutral
37	Loggiing of Quality related informaation of each Manufacturing Part	Quality is very important inside an assembly line as it is the essential parameter used for force tests or lack tests.Furthermore, if failures are detected lately when a car is already in the market, but shows some lack, the production history can be traced to find the devil in the detail.	COMAU workshop in Turino (19th Oct, 20010) employee (Fulvio) raised this issue.	Quality related information is logged inside a proper carrier medium.	1	Functional	Context	low	high
38	producers can push notification of recalled products to costumers	producers want to avoid getting sued because the weren't fast enough notifying consumers about recalled products. the common methods is through TV, Radio, Website, for cars can be through phones	TNM Workshop in Copenhagen, Comau Workshop in Turino	customers who bought the products are notified within 24 hours since products being recalled	2	Functional	Communication	high	high
39	correlate founded problem with production batches	when the source of problem have been isolated, producers must know which products/batches are affected.	TNM Workshop	production batches affected by problems can be identified.	2	Functional	Modelling	neutral	very high

40	automatic analysis of cross enterprises product life cycle data	searching production problem from end costumer complaints need to track back data from several enterprises and logistic.	TNM Workshop Copenhagen	analyzing data cross enterprises can be done online and automatically.	3	Functional	Communication	high	high
41	notification through out the chain	reduce time for recalling a product from end consumers.	TNM Workshop Copenhagen	at least 2 levels of the link in the chain can be notified automatically. e.g.: a farmer found a mutating pig problem, it can notify slaughterhouse and sausage factories who bought meat from the slaughterhouse automatically.	2	Functional	Communication	neutral	very high
42	supplier can predict when to make the next delivery to a consumer	suppliers can make an early offer when the inventory of their costumers almost depletes	TNM Workshop Copenhagen.	supplier can make an estimation when the inventory of their costumer are almost empty.	2	Functional	Context	neutral	high
43	Predict if his suppliers will not be able to fulfill their demands (quantity)	consumers are able to find other suppliers to fulfill their demands	TNM Workshop Copenhagen	consumers are able to make estimation if the a supplier will not be able to supply his demands.	3	Functional	Context	high	very high
44	Definition of smallest unit can be traced or uniquely identified	small parts of products / cheap parts of products cannot be identified anymore because of physical limitation of tags. / price of tags is too expensive to tag cheap parts.	TNM Workshop in Copenhagen	clear definition in both domains of what can be tagged.	2	Functional	Modelling	neutral	very high
45	officials have a back door access to highly important information	officials want to avoid enterprises commit information / documents forgery	TNM Workshop in copenhagen	offcials have an access to certain information	4	Functional	Interface	neutral	low

46	Transfering IDs cross enterprises	when IDs have no standard (such as EAN), IDs are generated from each ERP. thus each enterprise has their own IDs.	TNM Workshop in Copenhagen	IDs from cross enterprises can be linked automatically.	1	Functional	Communication	low	very high
47	system should aware of building condition in each farm	each building in farms have different characteristic that influences climate, insulation, etc	TNM Workshop in copenhagen	system can adjust itself according to building condition.	3	Functional	Context	very high	high
48	feeding systems should aware of the animals weight	farmers must keep poultry products on certain weights	TNM Workshop in Copenhagen	feeding system can control the portion of feed based on the weight	3	Functional	Context	high	high
49	ebbits should support legacy network interfaces	many legacy systems still use old network interfaces	TNM Workshop in Copenhagen	at least 3 types of common old interfaces of each domain (manufacturing and farm) are supported	1	Functional	Interface	very high	very high
50	system should provide location tracking of the stocks/livestocks	users sometimes lost track where the goods /animals are.	TNM Workshop Copenhagen, Comau Workshop Turino	users can identified where the stocks / livestocks are	1	Functional	Context	very high	very high
52	ebbits should bridge communication between different applications in farms	application between vendors are not able to communicate to each other, while data sometimes needs to be shared among applications.	TNM Workshop Copenhagen	at least applications from 3 different vendors are able to exchange information.	3	Functional	Communication	high	high
53	System should show Energy Cost for different granularity of production processes	energy cost at different levels is needed to do benchmarking of operational processes.	TNM Workshop in Copenhagen, Comau Workshop in Turino	each automated process, machine is able to show energy cost	1	Functional	Context	very high	very high

54	Protection to sensitive information	some sensitive information endanger company existence.	TNM Workshop, COMAU Workshop	system provides security in each level to access information??	1	Functional	Security	very high	very high
55	Adjust production processes according to energy price policies	reduce production cost by taking into account energy price policy from energy provider.	TNM Workshop in Copenhagen, Comau Workshop in Turino	at least production speed and start/stop production can be adjusted according to the price of energy.	2	Functional	Context	high	high
56	Interfacing with Programmable Logic Controller of production robots	production automation is controlled through PLC	Comao Workshop Turino	software and hardware interfaces to PLC is defined	3	Functional	Interface	very high	very high
57	3 Measurement Points for every station in body welding	energy cost that can be calculated includes: lightings, processes, energy for welding	Comau Workshop in Turino	3 measurement points are covered	3	Functional	Devices	very high	very high
58	Reliable wireless solution for new sensors	cable costs are high and due to harsh condition, cable might break	Comau Workshop in Turino	sensors are using wireless connection.	3	Functional	Networking	high	high
63	filter/fusion information for each operational process	each process needs different resolution of information	Comau Workshop in Turino	processes only get information needed	3	Functional	Context	high	high
64	early maintenance notification when needed	early maintenance prevent permanent damage to the robots, ensure the reliability of robots	COMAU Workshop Turino	users/technicians are notified if robots need maintenance	4	Functional	Context	high	high
65	bring data from fieldbus network to ethernet network	analytics is done by ERP program on a computer that work on TCP/IP.	Comau Workshop Turino	analytics software can analyse data from manufacturing robots	3	Functional	Networking	high	high

66	automatic calibration	calibration is still done manually it is error prone, and takes time.	Comao Workshop in Turino	75% of existing manual calibration is done automatically.	1	Functional	Context	high	very high
67	automatic start up synchronization among machines	starting up machines in a plant is complicated such as the order of machines, min temperature etc.	Comao Workshop in Turino	a plant can be "re-started" automatically in less than an hour.	2	Functional	Devices	very high	very high
69	energy benchmarking of different granularity such as machines, processes, plants	management would like to know how effective the energy is used in different operational levels.	Comau Workshop	management can do benchmarking in different operational level	1	Functional	Configurability	very high	very high
70	support system for comparing different energy consumption among plants and corresponding processes	management would like to learn from other plants if they use energy more efficiently.	Comau Workshop in Turino	management can compare energy profile of plants.	2	Functional	Context	very high	high
71	Summary of energy related information at operational level for supporting management level optimizing energy use	operational management needs a summary of energy related information that help them making decision to optimize the energy usage.	comau workshop in turino	management can access operational information.	3	Functional	Context	high	very high
72	recognition of energy wasting behaviors	help decision makers to optimize energy usage	Comau Workshop	decision makers are alerted when energy wasting takes place	4	Functional	Context	neutral	neutral

73	Items need to be traced within an enterprise	Goods and items need to be traced within one farm or enterprise.	TNM user workshop in Copenhagen	Any item that has an ebbits identifier can be reliably located within a determined area.	2	Functional	Context	high	high
74	Items need to be traced between enterprise	Goods, parts and items need to be traced when they leave one enterprise and appear in another.	TNM user workshop in Copenhagen	Any item with an ebbits identifier can be recognized when it leaves and enters an enterprise.	2	Functional	Context	high	high
75	Information needs to be described in a standardised way	Enterprises working in the same sector adapt different ways to describe the input, the production processes, and the output; thus it will not be possible to communicate information either to providers or to consumers.	TNM user workshop in Copenhagen	A standard for exchanging information between system components is provided.	3	Functional	Communication	high	neutral
76	The ebbits platform should amplify branding for enterprises.	Support for branding in the ebbits platform will help to overcome several challenges and needs to be analyzed in details in the ebbits business models. Thus, using and being part of the ebbits platform should be a quality attribute. This is a demand from consumer in future.	TNM user workshop in Copenhagen.	90% of such businesses exploiting the ebbits platform perceive this as a quality attribute.	3	Functional	Communication	high	very low
77	Associate meta-information to items	In parallel to the actual lifecycle (grow up of the animal, feeding, butchering, transportation, selling, consuming) there exists additional information such as the amount of food, medication an animal has had, the energy for the production and transportation, that needs to be acquired and associated with the (bits and pieces of) animal.	TNM user workshop in Copenhagen.	Any item with an ebbits identifier can be associated with a set of meta-information.	3	Functional	Context	high	low

78	Batches need to be identified on a farm level	The average feed production batch size is 20 tons, however farmers want their own specific production which is way smaller. One unique identification of the batch to the farm is necessary.	TNM user workshop in Copenhagen	Every batch has one unique identifier.	2	Functional	Modelling	high	neutral
79	A standard way of numbering batches needs to be provided	No world standard way of numbering the batches exists. The numbering depends on the local ERP system: Currently, one solution is to create a huge database to link the numbers together	TNM user workshop in Copenhagen	A standard way of numbering/identifying batches exists.	3	Functional	Modelling	very high	low
80	Energy consumption should be optimized automatically	Farmers want to optimize the energy consumption in their production and they adapt this manually, and they would like to have this done automatically (e.g. putting some production steps towards a cheaper time of the day).	TNM user workshop in Copenhagen	Rules can be defined to automatically optimize the energy Consumption	3	Functional	Configurability	very high	high
81	Item identification system should provide open interfaces to other systems	Identification of pigs is done with RFID tags at their ears and with antennas in corridors that recognize pigs passing by. The system should not be connected to a specific system, but rather provide open interfaces that can be exploited by any system.	TNM user workshop in Copenhagen	Any system can easily access the item identification system.	3	Functional	Interface	neutral	very high
82	Support fuzzy or probability concepts for reasoning	there is no reasoning algorithm that is able to solve any kind of cases	Hydra open requirements	Fuzzy concepts should be supported through e.g. probabilistic models.	4	Functional	Modelling	neutral	neutral
83	Device and service exception handling	The development and run-time environment should support exception handling constructs that the developer can employ to manage service and device availability and malfunctioning, isolated from the main application logic.	Hydra open requirements	Exception handling constructs that the developer can use to specify exception responses with a success rate of 9/10.	3	Functional	Devices	high	neutral

84	Different views on the device ontology	It should be possible to present a developer user with different perspectives on the device ontology, depending on that users functional needs (e.g., a services perspective, device category perspective. etc.)	Hydra open requirements	At least two different views are available in the ontology browser.	3	Functional	Interface	high	neutral
85	Ability to self-adaptation	A knowledge model enables the middleware to contain a representation of itself and manipulate its state during its execution. This feature should serve as the basis for self-adaptation of the middleware (e.g. reconfiguration of resource usage, triggering the component-based services).	Hydra open requirements	Middleware is able to adapt its configuratiton in 60% of identified cases requiring reconfiguration.	2	Functional	Configurability	neutral	low
86	Protection of System Integrity	In order to prevent an inexperienced user to cause malfunctions by changing system configurations, the middleware should monitor, analyse and, if necessary, prevent or give notifications about faulty changes.	Hydra open requirements	Middleware provides mechanisms to monitor system integrity and to react in the case of failures.	2	Functional	Security	high	high
87	Handling of different device versions in device ontology	The device ontology should be able to handle different versions of a device.	Hydra open requirements	The device ontology can maintain at minimum 2 versions of any single device	3	Functional	Modelling	neutral	neutral
90	Support runtime reconfiguration	To supporting monitoring leading to adaptation, the architecture should be dynamic in the sense that components/services should be connectable at runtime.	Hydra open requirements	Services and devices can be connected during runtime.	3	Functional	Configurability	high	high
92	Report errors in devices	Devices should be able to report errors.	Hydra open requirements	Devices provide services for reporting errors.	2	Functional	Devices	high	neutral

93	Download and harmonisation of third party device ontologies	Device ontological models describing devices, which will be provided by manufacturers or third parties, should be automatically downloaded (updated) and harmonised to ensure the same ontological view. Formal definition of ontologies should be realised using the world wide accepted formats, recommended by W3C, such as RDF, OWL, OWL-S.	Hydra open requirements	Ontologies from different manufacturers can be used if they are in RDF, OWL or OWL-S.	1	Functional	Modelling	very high	very high
97	Support for interfacing with external workflow systems	Applications must include workflow management possibilities	Hydra open requirements	Supports at least one workflow system, for instance OpenWorkFlow.	2	Functional	Interface	high	high
98	waste of energy act definitions	some users are waisting energy without realizing/being conscious that there are better alternatives.	Comau Workshop	energy wasting behaviors are modeled	3	Functional	Modelling	very high	very high
100	Retrieve the behavior on an individual animal level	Monitoring the drinking behavior allows to recognize diseases 20 hours before. However, today it is not able to retrieve the drinking behavior on an individual pig level, rather in a group. But with the emerging RFID identification it would be possible	TNM user workshop in Copenhagen	The drinking behavior can be retrieved on a individual pig level	3	Functional	Context	very high	low
102	Common structure of information is needed	Relevant data to describe specific situations of a pig needs to be determined and put into a common structure such as an ontology, a common definition of data.	TNM user workshop in Copenhagen	Common data structure to describe situations of arbitrary entities is available	3	Functional	Modelling	high	high

103	The system should allow the correlation of information emerging from several sources	In order to easily analyse information, the system should allow for the correlation of information from different sources on a farm or enterprise	TNM user workshop in Copenhagen	Acquired information is timestamped.	3	Functional	Modelling	high	low
104	Store meta-information with package labels	Temperature, location, humidity needs to be acquired and assigned with the packages. Dimension and weight is also required for the package, box, palette, etc.	TNM user workshop in Copenhagen	A minimal set of 5 attribute-value pairs can be associated with any package/label.	3	Functional	Modelling	high	neutral
105	Aggregate data from various data bases and sources	Information will be stored in several places, but needs to be combined in some place and assigned to the actual product or entity.	TNM user workshop in Copenhagen	A data aggregation component is available.	3	Functional	Architecture	neutral	high
106	Synchronisation of Acquired Data is necessary	Data synchronization might be necessary, because data will be acquired automatically, manually, semi-manually with different timestamps.	TNM user workshop in Copenhagen	A data synchonization component performes a timestamp-based synchronisation of a data set.	3	Functional	Context	high	neutral
108	Different Views on the Data is necessary	We need services that provide different views on the data cloud by combining data from different sources.	TNM user workshop in Copenhagen	Data can be filtered and sorted based on an arbitrary set of parameters.	3	Functional	Interface	very high	very low
109	Alarms are send when specific situations occur	Issue alarms when an animal behaves differently. However, context-aware behavior is wished for experts NOT for the farmers.	TNM user workshop in Copenhagen	Simple rules can be defined that trigger alarms.	4	Functional	Communication	high	very low
111	The system should be self-ware	User statement: "Let me make my system self-aware"	TNM user workshop in Copenhagen	User gets informed of any event she is interested in.	3	Functional	Context	high	neutral

113	Access-control of data sets	Access to data needs to be controlled, because some authorities require having access to this data, other stakeholders might have restricted access, other information could be made publicly available.	TNM user workshop in Copenhagen	Access rights can be defined for several stakeholder roles.	1	Functional	Security	high	very high
114	The traceability chain should be computed on demand	The traceability chain should be computed on demand and not stored and recalled, due to storage. Thus, the combination of data is the challenge and a handover of identities might be required. A combination of searches should be realized by semantics.	TNM user workshop in Copenhagen	The traceability chain is computed and not stored.	3	Functional	Communication	neutral	high
117	Integration of legacy systems into ebbits platform	ebbits platform is deployed for solving interoperability of the existing systems (software and devices)	Michael Jacobson, Requirement Quality Check	3 different existing systems used in each domain is supported e.g.: Farm : Climate, Farm Management, Feeding Slaughterhouse : ERP Body welding : ERP + Robot Management	1	Functional	Interface	very high	very high
118	save historical information in farms	feeding history	Michael Jacobson	historical information is saved	3	Functional	Context	very high	high

8.2 Non Functional Requirements

No	Summary	Rationale	Source	Fit Criteria	Priority	Requirement Type	Components	User Satisfaction	User Dissatisfaction
2	The ebbits should be able to handle massive number of devices	The future use cases of eBBits need to handle massive number of devices and applications within and cross enterprises, i.e. ci. 300-1000 in a manufacturing plant and 500 in a farm.	TNM scenario workshop issued by Jesper.	eBBits is able to handle 500 devices simultaneously.	2	Non-Functional :Operational	Networking	low	high
10	Environmental Monitoring of Manufacturing Plant	Currently the environment of a plant provide is not monitored properly. However, this is quite important to guarantee the safety of an operator.	COMAU workshop in Turino (19th Oct, 20010) employee (Roberto) raised this issue.	The safety of the operator is improved by 20% on the basis of enviromental input information.	1	Non-Functional: Maintanability	Context	low	very high
12	Higher bandwidth and range of wireless connection	Currently, BT is used for wireles communicaction. It supports ranges <50m. Especially, the bandwith is too low: max 24 MBit/s.	During ebbits manufacturing workshop (19th Oct, 20010) a COMAU employee of the Body Welding & Assembly unit raised this issue.	Communication range and in particular bandwith are considerably higher.	3	Non-Functional: Performance	Networking	low	very high

15	Life-cycle of a robot and its components is traceable	At the moment the lifecylce of a robot's component is not predictable. However, being able to predict its life-cylce could support to avoid deviations during production.	COMAU workshop in Turino (19th Oct, 20010) employee raised this issue.	Based on analyzing data of real field tests the lifecylce can be predicted properly.	2	Non-Functional:Maintanability	Context	very high	neutral
21	Affordable tagging/tracking system for pigs	If the tagging price is to high, the farmers are reluctant to use this tags	Thomas (TNM workshop)	The price of a tag is less than 5% of the total profit a farmer can get from a pig.	2	Non-Functional: Operational	Devices	very high	low
22	Resilience and adaptable to environment condition changes	environmental changes such as lighting, temperature affect the results of manufacturing process. so far machines are tuned manually by technicians. adapting to environmental condition can lead to reducing energy consumption e.g.:reduce heater temperature when it's warm outside.	During ebbits manufacturing scenario workshop in Oct 2010 this is issue had been raised by a COMAU employee.	machines can adapt its parameters adapting to environmental changes.	2	Non-Functional: Operational	Configurability	low	high
24	Filtering to Obtain relevant Information	Too much information overwhelm farmers while making decisions.	TNM scenario workshop in Copenhagen	Farmers are able to view the relevant information out of the whole.	3	Non-Functional: Usability	Interface	low	high
25	The network infrastructure needs to have self-configuration capabilities	Due to the huge amount of heterogeneous devices that can be connected to one network, this network needs to support the deployment through some sort of self-configuration.	TNM scenario workshop in Copenhagen.	A network of 20 devices can be deployed within one hour.	3	Non-Functional: Operational	Configurability	high	neutral

27	New products should be networked with mainstream enterprise systems easily and cost-efficiently.	New products should be integrated into existing systems easily and cost-effectively, in order to support higher value-added, interoperable solutions.	TNM scenario workshop in Copenhagen.	A new product can be connected to an existing enterprise system within one day by one person.	3	Non-Functional: Operational	Configurability	high	very high
33	Improve air compression energy usage	Air compression is one main energy guzzler. Only 40% of air can be transferred effectively.	COMAU workshop in Turino (19th Oct, 20010) employee (Roberto) raised this issue.	More than 40% of air can be transferred effectively (@Roberto is this realistic?)	3	Non-Functional: Operational	Devices	very high	very low
34	Display plant activities in real-time	To observe the complexity of a production inside the plant.	COMAU workshop in Turino (19th Oct, 20010) employee raised this issue.	A user-friendly interface is provided to the relevant stakeholders to view activities inside the plant.	3	Non-Functional: Usability	Interface	low	high
51	location tracking should be implemented as independent app	decoupling from existing system	TNM Workshop Copenhagen	tracking system is implemented independently	2	Non-Functional: Operational	Configurability	very high	high
59	reliability of the system should be more than 30 years	a production plant of trucks can runs for 30 years	Comau Workshop in Turino	99% of the time system is able to run 30 years.	2	Non-Functional: Operational	Devices	very high	very high

60	hardware components are able to handle harsh condition	harsh condition damage electronic devices	Comau Workshop in Turino	devices fulfill manufacturing insulation standard for cables and sensors.	3	Non-Functional: Operational	Devices	high	high
61	Scalable solution (scale up and scale down)	adjustment to desired number of production, require to add or reduce machines	Comao Workshop in Turino	configuration of scaling up / down a plant can be achieved in max 8 hours.	3	Non-Functional:Maintanability	Architecture	very high	very high
62	central point to start the whole plant	machines have to be started in the right order.	Comau Workshop Turino	the whole machines in a plant can be started from a central point.	3	Non-Functional: Operational	Interface	very high	very high
68	reduce water consumption in PWT	water consumption for cooling and lubricating purposes in PWT is really high (300-500 lt./minute)	Comao Workshop Turino	10% of water consumption can be reduced	4	Non-Functional: Operational	Devices	high	high
88	Systems built using HYDRA should be scalable in terms of devices communicating	In large installations (e.g. in large factories) there will be many (embedded) devices in total. The middleware should support the development of such big systems.	Hydra open requirements	The middleware supports applications in which more than 500 devices exist.	3	Non-Functional: Performance	Configurability	neutral	neutral

91	Support profiling device performance	The middleware should contain services that allow monitoring on what devices are doing. This includes monitoring response time, device load (e.g., CPU), and message interchanges per second.	Hydra open requirements	Devices provide monitoring services.	3	Non-Functional: Operational	Devices	neutral	neutral
94	Comply with industrial standards	The middleware should embrace existing industrial device integration and communication standards, e.g. EIB/KNX	Hydra open requirements	Claimed support for any specific standard in ebbits can be verified using the conformance rules / procedures available from the issuing standards body.	2	Non-Functional:Legal	Networking	very high	high
95	Configurable and easy to install middleware	The middleware should be configurable and easy to install/deploy.	Hydra open requirements	The average installation time is less than 1 hour.	1	Non-Functional:Usability	Configurability	very high	very high
99	Mobile management of farms	Mobility and mobile devices are required to manage aspects of the farm locally and remotely if you are on holiday or in the cinema.	TNM user workshop in Copenhagen	Mobile access to important function of the management system is granted.	3	Non-Functional: Operational	Devices	very high	low
101	Applied stand-alone devices should have a long battery life span	Data analysis is not done on the chip, because this consumes energy and the battery life-span comes to 3 months which is far too low.	TNM user workshop in Copenhagen	Applied stand-alone devices have a minimal battery life span of one year.	3	Non-Functional: Operational	Devices	high	low

107	Meta-information associated with entities needs to be trustworthy	ebbits needs to guarantee that the information associated with a specific product is the right and correct one that has not been manipulated.	TNM user workshop in Copenhagen.	Acquired information cannot be manipulated by unauthorized people.	3	Non-Functional:Security	Security	high	very high
110	End-users need to be able to managment their distributed data	Farmers want to manage their distributed data, because today they have no full control of data.	TNM user workshop in Copenhagen	End-users can easily manage data from distributed sources.	3	Non-Functional:Usability	Interface	very high	neutral
112	Electronic exchange of reports between enterprises	Report on the slaughtered pig is sent back ON PAPER to the farmer. This should be done electronically, in order for the farmer to combine it with the feeding information, e.g. identification of a father that "produces" ill pigs.	TNM user workshop in Copenhagen	Information is automatically put into a report format and send to a recipient.	3	Non-Functional: Operational	Communication	low	very low

8.3 Constrains

No	Summary	Rationale	Source	Fit Criteria	Priority	Requirement Type	Components	User Satisfaction	User Dissatisfaction
89	Distributed Intelligence should not lead to resource-heavy systems	We have a need for "intelligence" (Semantics, reflection etc.). We have a need for supporting embedded systems. This should not conflict	Hydra open requirements	Minimum hardware requirements (which must be supported by all target hardware) are defined and all hardware that meets the specifications is guaranteed to work with hydra.	1	Constraint:Scope of the product	Architecture	neutral	neutral
96	Support of low-end devices	Middleware must support low-end devices like RFID tags. Therefore, it must be compatible with at least 32-bit devices with < 512 KB RAM/FLASH or less. For smaller devices, proxies can be used.	Hydra open requirements	Middleware is able to be installed and run on low-end 32-bit devices with 512 KB RAM/FLASH in 90% of all cases. Proxies can be created to support more limited devices in 40% of all cases.	2	Constraint:Scope of the product	Architecture	high	neutral
115	Consider ethical issues	It is an ethical issue if you track the driver driving a van or that a product is at the end-user's fridge	TNM user workshop in Copenhagen	-	3	Constraint:Ethical	unassigned	neutral	very high
116	Consider cost benefit	There is no NICE-To-HAVE in the agriculture domain, there always needs to be cost benefit. A slaughter pig brings 5-15 Euros.	TNM user workshop in Copenhagen	Cost benefit is considered.	3	Constraint:External factor	unassigned	neutral	very high