Project no. 027087

TENCompetence

Building the European Network for Lifelong Competence Development

Project acronym: Integrated Project TENCompetence

Thematic Priority: 2.4.10

D2.1 Integrated Roadmap

Due date of deliverable: 01-06-2007
Actual submission date: 07-12-2007

Start date of project: 01-12-2005 Duration: 4 years

ALTRAN SDB

Version 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Building the European Network
For Lifelong Competence Development

TENCompetence IST-2005-027087

Building The European Network for Lifelong Competence Development

Project Deliverable Report

D2.1 – Integrated Roadmap

Work Package	WP2 – Requirements & Analysis of the Integrated System
Task	Task 1
Date of delivery	**Contractual**: 01-06-2007 **Actual**: 07-12-2007
Code name	D2.1 **Version**: 1.0 Draft ☐ Final ☒
Type of deliverable	Report
Security (distribution level)	Public
Contributors	Miguel Arjona Villanueva (Altran SDB), Antonio Sánchez (Altran SDB), Naiara Sacristán (Altran SDB), Christian Glahn (OUNL), Marcus Specht (OUNL), Steven Verjans (OUNL), Ruud Lemmers (LOGICACMG), Henk Sligte (UvA)
Authors (Partner)	Altran SDB (Miguel Arjona, Antonio Sánchez, Naiara Sacristán) OUNL (Christian Glahn, Marcus Specht, Steven Verjans)
Contact Person	Miguel Arjona
WP/Task responsible	Miguel Arjona
EC Project Officer	Mr. M. Májek
Abstract (for dissemination)	Deliverable 2.1 – *The Integrated Roadmap* – summarises the first 18 months of requirements gathering and analysis in the TENCompetence project. The document consists of a summary description and a number of annexes with detailed results. The methodology chosen by the project is the Unified Process, supplemented with scenario-based software development techniques. On the basis of initial scenario's and specific use cases, six high-level use cases were identified that summarise the future functionality of the TENCompetence integrated system. These high-level use cases build on the domain model that is also included. The four main components of the TENCompetence project, i.e. (a) the high-level use cases, (b) the domain model, (c) the project objectives and (d) the experimental setup of the pilots were then critically analysed in order to identify possible gaps between them. On the basis of this gap analysis, some recommendations were formulated for the next development cycles. On the basis of all the work in the four components and the gap analysis, detailed extended use cases with activity diagrams and a data model were developed and formulated, which again serve as the basis for the first version of the integrated system, the Personal Competence Manager. Finally, the document describes the future of the requirements process in the form of a research roadmap, and a detailed procedure for handling change requests to the integrated system.
Keywords List	Requirements, methodology, integration, use cases, domain model, data model, activity diagrams, research roadmap, change request.

TENCompetence Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 45 5762624 – Fax: +31 45 5762800

Table of Contents

Executive Summary

The focus of this deliverable is the definition of the enabling structure and systems for the first integrated pilots. It contains the following chapters:

1. The introduction describing the role of WP2 in TENCompetence
2. The **methodology** for collecting user requirements
3. A first main result of the project's first phase, the **Initial User Requirements,** formulated in the form of high-level use cases and a domain model, both of which have been derived directly from the TENCompetence project objectives
4. The second main result from this phase, the ongoing work (after the initial requirements formulation) is reported in the **Gap Analysis Report** and the section on **Elaborated Use Cases and data model.** This chapter specifies the current functionality of the integrated system that has been implemented in the first version of the Personal Competence Manager (PCM)
5. The sixth section of the report contains a **research roadmap** that forecasts the future functionality that will be progressively incorporated into the new versions of the integrated system and a time-plan for the delivery of these releases
6. This research roadmap is complemented with a **change request procedure** that details the mechanisms that will facilitate the communication between the different development and research groups and the management of change requests for the system functionality.

Details of chapters 2 to 5 of this report can be found in the following annexes:

- Annex A: Initial User Requirements (ID2.1)
- Annex B: Domain Model
- Annex C: Gap Analysis Report (ID2.4)
- Annex D: Elaborated Use Cases.

1 Introduction

1.1 WP2 within TENCompetence

The aim of the TENCompetence project is to develop an infrastructure for lifelong competence development. To this end, four work packages (the so-called 'Aspect work packages') within TENCompetence are devoted to the elaboration of one of the four core aspects of competence development:

1. Knowledge Resource Sharing & Management (WP5)
2. Learning Activities and Units of Learning (WP6)
3. Competence Development Programmes (WP7)
4. Networks for Lifelong Competence Development (WP8).

The core outcomes of these work packages are models, methods and tools.

Three other work packages are devoted to the integration of the outcomes of the four aspect work packages by developing and validating an integrated model and infrastructure for competence development; these are called 'integration work packages'. WP2 'Requirements & analysis of the integrated system' is one of these. The other two are WP3 on Technical design & Implementation of the Integrated System and WP4 on Validation of and Pilots with the Integrated System. Within WP2, the emphasis is on overall integration of the first two core workflows of the Unified Process: Requirements and Analysis.

Because of its overall integrative function, WP2 integrates as well as steers the activities of the Aspect work packages. Besides, WP2 affects the other two integration work packages. The ongoing work in WP4 (Pilots) will be coordinated with work in WP2, translating the plans and scenarios of the pilots into use cases and a conceptual framework, all in relation to the theories underpinning the project. The evaluation plan will also be translated into user needs and requirements for the system. Work will also be closely coordinated with WP3.

In the first 18 months the focus was on the definition of the enabling structure and systems for the first integrated pilots. Its objectives for the first 18 months of the project are:

1. Define the requirements of the Integrated System in terms of a Use Case Model.
2. Define the non-functional requirements for the Integrated System.
3. Analyse the requirements & select and use underlying theories, models & methods used in the project to model the conceptual structure of, and processes in the integrated system (in terms of a Domain Model).
4. Create an integrated roadmap for further development and refinement of the functional requirements and integrated pilots for the next 18-month periods of the project.

This report starts out from the seven objectives of the TENCompetence project and reflects the work performed in WP2 during the first 18 months of the TENCompetence Project. Next to that, the future short and medium term research and development plans are presented. The report consists of a concise summary of the results and a number of annexes, which present the detailed outcomes of this phase of the project for WP2.

As a reminder: the **seven objectives** of the TENCompetence project are to research and develop:

1. New, promising, innovative pedagogical approaches for lifelong competence development, supported by the TENCompetence infrastructure.
2. Tools to support individuals, groups and organisations in Europe to find the best solution for their formal or informal learning problem.
3. Policies and software agents that support the pro-active sharing of knowledge and learning resources.
4. Models and software tools to assess the competences of individuals, groups and organisations in an exchangeable way.
5. Software for the effective and efficient support of users who create, store, use and exchange knowledge resources, learning activities, units of learning, competence development programmes and networks for lifelong competence development.
6. Software solutions to establish a decentralized, self-organized and empowered management model when using the TENCompetence infrastructure.
7. Integration of isolated tools that are available in the field.

2 Methodology for Collecting User Requirements

In relation with the functionality presented in this report, TENCompetence chose the **Unified Process** (an open version of RUP –Rational Unified Process-) as the main development framework as it conjugates a robust methodology with the rapid development of prototypes that are refined in several cycles allowing for an increasing understanding of the problem through successive refinements.

Inside this methodology, collecting user requirements constitute the starting point of the conception phase. This task, usually complex, is more in ambitious research projects as TENCompetence in which the quick development of prototypes is required to test and validate the concepts that are being investigated and where the potential users have not a clear idea about the implications that such as investigations could have in their organisations. For this reason, we decided to use **scenario-based software development techniques**, method not common in UP but profusely used in other methodologies such as **GDD** (Goal Directed Design) or **USBD** (Unified-Scenarios Based Design).

According to UP, Use Cases come from customers and users, and subsequently require prototyping, iteration (of code), and repeated customer/user involvement to validate, reconcile, and consolidate the Use Cases and the software defined by the Use Case. This process is predicated on the assumption that it is impossible to make sense of requirements until you have written some code and put it in front of users and customers, because there is really no anticipating or accounting for what people will actually want. Alternatively, **it is held that anticipating and serving human needs can be achieved in advance, using scenarios**. By determining motivations (rather than simply tasks), we can anticipate and proactively serve needs and requirements that our users and customers can not yet identify. This stems the flow of requirements delivered after design is completed.

The TENCompetence user requirements' definition process started with the collection of scenarios and specific use-cases conceived by the project research groups and potential users. In these scenarios they expressed their vision of the future use of tools that are to be developed. After a first analysis, the scenarios were grouped in six use-cases that expose the high level functionality that TENCompetence's integrated system should offer to its potential users.

These High Level Use Cases were completed with others coming from the "Aspect Work Packages" (WP8: Learning Networks, WP7: Competence Development Programs, WP6: Unit of Learning & Learning Activities and WP5: Knowledge Resources). Additional use cases were created to cover the required functionality of the pilots (WP4) in which our research will be tested and validated (see Annex A: Initial User Requirement Report for a detailed description of High Level and Aspects Use Cases).

However, this heterogeneous set of scenarios and use cases, collected from diverse sources, presented several incoherencies and gaps when considering them as a whole. Moreover, other considerations had to be taken in account. First, TENCompetence is conceived over a theory, result of previous studies, which is expressed through a **Domain Model** [Annex B, Koper, R., 2006]. This model constitutes the project background; therefore it should be the reference framework for future developments. Second, the consortium has imposed a concrete set of **research objectives** [DOW v3 28-9-05] that should guide the developments too. And third, the research outcomes should be validated, in successive experimentation phases, in real environments (**pilots**) that, of course, have their own needs.

In summary, four factors had to be taken into account after the first requirements gathering phase:

1. the main system functionality expressed in the **six high level use cases** and in the **aspect work packages uses cases**,
2. the underlying theory expressed in the **domain model,**
3. our **research objectives,**
4. our **experimentation needs**.

The next diagram illustrates the main relationships between these four factors:

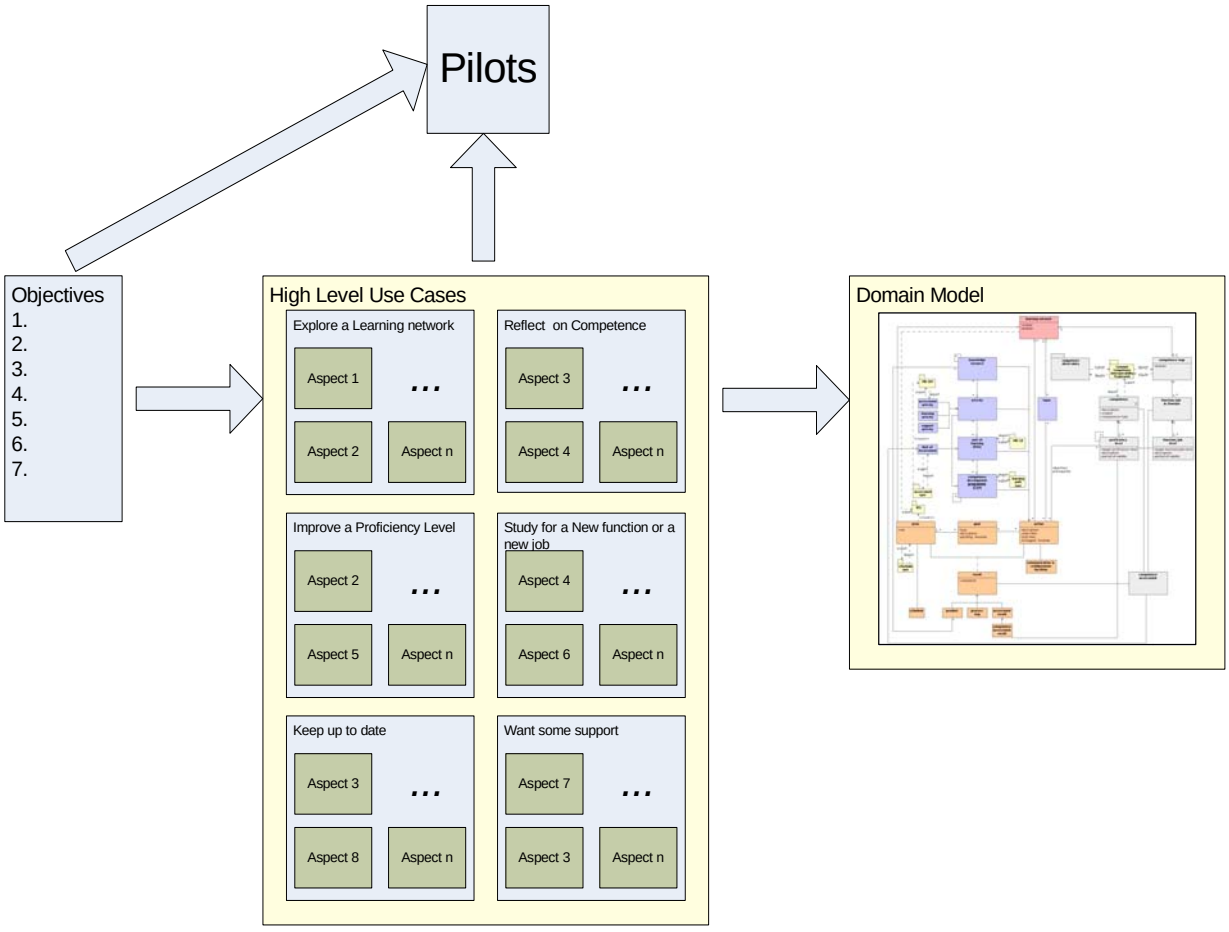


Figure 1: Relations between four important components of the requirements gathering work

From the cross-analysis of the complex scenario in figure 1 the need arose for **analysing the gap** between (a) current specifications, (b) the base theory, (c) our research objectives and (d) the experimentation environments. This Gap Analysis is included as annex C to this document and was intended to get answers to questions such as:

- Do we reach our research objectives if we develop the system that has arrived from the analysis of the scenarios proposed by the different research groups and potential users? (see Section 4 in Annex C: Gap Analysis). If we are not able to reach them, how do we have to extend this vision?
- Does our theoretical model fulfil our research objectives? (section 5 in Annex C: Gap Analysis). In case it is not possible, how can we extend this model?
- Do the theoretical concepts support our vision of the project expressed in the six high level use cases? (section 6 in Annex C: Gap Analysis). In case we get a negative answer, do we have to limit our vision to ensure the viability or do we have to extend our theoretical base to support this new vision?
- Are the proposed experiments adequate to validate our researches? In the same way, is the current system functionality enough to complete the needs of the experimentation environments? (section 7 in Annex C: Gap Analysis). If they are not sufficient, how do we have to modify these experimentation scenarios to complete our needs?

As a result, several recommendations were proposed to be taken in account in the following development cycles (DIP-2/3). These recommendations (1) have affected the next step of requirements elaboration (the **Extended Use Case** model presented in this report), (2) have advised us the definition of a new task in WP2 dedicated to the research in new pedagogical models (3) have provoked the modification of our Domain Model including new concepts present in the scenarios, (4) have changed Aspect Work Packages' research roadmaps and (5) have make clear the need of conceiving new experimentation environments that permit the validation of our outcomes.

The last step in such as methodology is presented in the **Annex D** and consist in the elaboration of extended use cases based on the previous work and represented as activity diagrams and data models. This functionality has to be implemented in the first version of the Personal Competence Manager (PCM).

In the following paragraphs, specific outcomes of the methodology will be presented in more detail.

3 Main outcomes of initial user requirements process

3.1 High Level Use Cases

Below you can find a brief description of the six high level use-cases. The models that describe them can be found in **Annex A-Initial User Requirements**.

1. **Improve Proficiency Level:** the learner's motivation in this use case is to improve his/her proficiency level for a specific competence.
2. **Keep up-to-date:** Knowledge, skills, etc. change over time and professionals must update their competences to maintain their proficiency level. The mission of this use case is not getting a better job or acquiring a higher proficiency level, it's preventing the learner's proficiency level from going down.
3. **Reflect on Competences:** from the current Learner's proficiency levels, the system will show him/her all new functions/jobs that match or are compatible with his ePortfolio. Based on this information, the learner will decide how to proceed.
4. **Study for a New Function or Job:** The motivation of the learners in this use case is the wish or the need for the development of competences and skills for a new job or a new position. The goal of this use case is to find or create appropriate competence development programmes (CDPs) for the development of competences and skills that are necessary to master a new function or a new job.
5. **Want some Support:** The motivation of the learners in this use case is the need for support for an action. The objective of this use case is to provide support and to rate the received support.
6. **Exploring a Learning Network**: the learner's motivation is to explore a Learning Network looking for topics, actions, issues, etc., that suits his/her aspirations and/or wants. It may be considerer as a prerequisite to other use-cases.

The functionality described in these high level use cases was supplemented by the functionality which have been described in the aspect work packages: Learning Networks, Competence Development Programs, Unit of Learning & Learning Activities and Knowledge Resources. UML Diagrams providing details of each component can be found in **Annex A** too.

3.2 Domain Model

The first release of the Domain Model is described in the **Annex B**. It is expressed as an UML class diagram and a vocabulary defining each concept (class) in the model. The Domain Model serves several functions in the project:

a. to define the scope of the project, including the scope for the use cases,
b. to define the vocabulary used,
c. to define the relationship between the concepts used,
d. to define the overall conceptual architecture,
e. to provide a technological theory for the project that must be tested in the pilots,
f. to provide a starting point for the design of other models, like the data model and services,
g. to provide the minimal functional components that must be present in the TENCompetence infrastructure.

The TENCompetence System that we are developing must meet at least 7 core functional requirements according to the project plan:

1. Support for new, promising, innovative pedagogical and organisational approaches for lifelong competence development that use the possibilities of new technologies available. This includes an integration of formal and informal learning.
2. Help learners to get an overview of all the possible formal and informal knowledge resources, units of learning, programmes and learning networks that are available, and to identify the most appropriate for their needs and background.
3. Stimulate the pro-active sharing of knowledge resources.
4. Provide support for competence assessment, including the assessment of the competences of applicants, employees and learners who have studied and worked in a variety of formal and informal settings.
5. Provide effective and efficient support for users during the performance of the various tasks in various roles (learner, teacher, assessor, etc.).
6. Provides support for decentralized, self-organized and empowered management.
7. Integrates four different types of models and tools used for competence development, i.e. tools and models for:

a. knowledge sharing & management,
b. the creation & use of learning activities and units of learning,
c. creation & use of formal and informal competence development programmes for lifelong learning and
d. creation and use of learning networks & learning communities for lifelong learning.

The model is a further elaboration of the initial domain model that was specified in the project plan. It includes more details than the original version and it concentrates on the aspects of the system that will be represented in the TENCompetence System to fulfill the 7 requirements.

All primary use cases are connected to the 'goal' class in the Domain Model. In the current description a lifelong learner can use the system to get support for the attainment of the following goals:

1. I want to keep up to date within my existing function or job
2. I want to study for a new function or job or improve my current job level
3. I want to reflect on my current competences to look which functions and jobs are within my reach or to help me define new learning goals
4. I want to improve my proficiency level of a specific competence
5. Want some support on a non-trivial learning problem
6. Want to explore the possibilities in a new field (learning network) to help define new learning goals.

The model was drawn with the UML tool 'MagicDraw' version 11.0. The documentation in Annex B is a copy of the documentation in the MagicDraw document.

The current version of the Domain model is depicted in figure 2 below. The most recent version of the Domain Model can be found at http://hdl.handle.net/1820/649

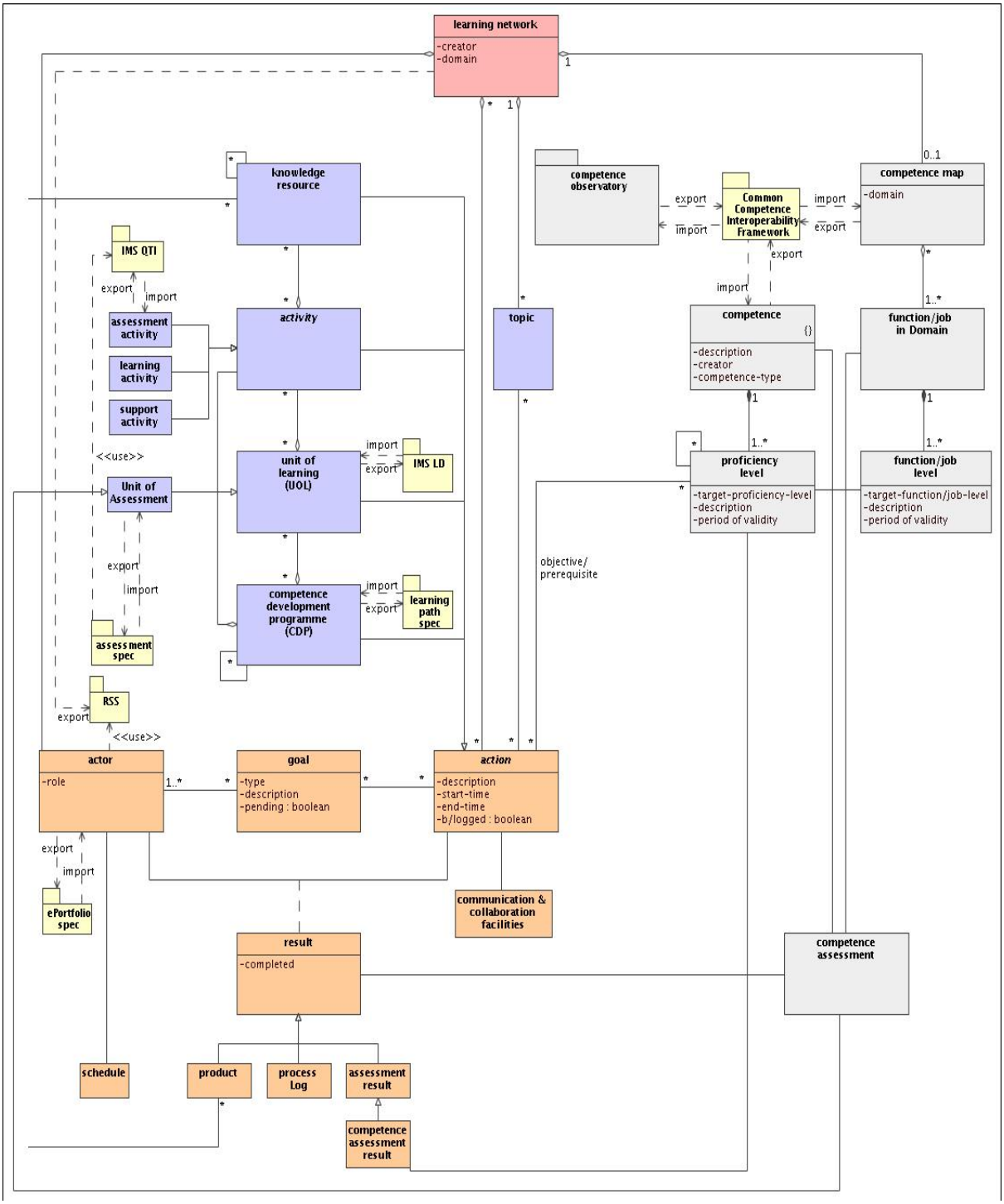


Figure 2: Domain model for the TENCompetence project.

3.3 Gap analysis: process and conclusions

As argued above, at a certain stage in the project, TENCompetence had produced a heterogeneous set of scenario's and use cases which – when taken together as a whole – presented several inconsistencies and gaps. Summing up, after the initial phase of requirements collection four factors had to be taken into account (cf. Figure 1 in the methodology chapter above):

1. the main system functionality as expressed in the **six high level use cases** and in the **aspect work packages uses cases**,
2. the underlying theory expressed in the **domain model,**
3. our **research objectives**,
4. our **experimentation** needs, as detailed in the pilot plans in the project proposal.

Four main analyses were performed in which the gaps between the following entities were investigated:

- between current project vision (six high level use cases) and (seven) project objectives
- between theoretical model (domain model) and project objectives
- between theoretical model (domain model) and current project vision (high level use cases)
- between project objectives and the experimental pilots' validation potential.

The main conclusions from the gap analysis were the following:

- The underlying theory – as schematised in the domain model – provides a solid joint reference and vocabulary for the research and development work in TENCompetence. However, consistency in the use of terms and descriptions needs to be guarded and the efforts on a common glossary need to be enhanced. It was also found that the relations between classes and data are not well specified. For this purpose, the elaborated use cases and data models were added in the following phase (cf. below).
- The processes of creating and managing the resources were lacking from the domain model and high-level use cases. These processes too were given additional attention in the subsequent phase.
- In the original model and use cases, informal learning activities and resources were not given enough explicit attention. Implicitly, the importance of informal learning was acknowledged, but that did not reflect in the original models. The subsequent work takes this remark into account.
- Similar recommendations were made regarding specific aspects of the models, such as dependencies, workflow, version management, organisational services, etc.

Those additions to the initial blueprint of the system and the research focus were dealt with (and will receive continuous attention) in subsequent descriptions of project work (DIP 2/3).

4 Current results: elaborated use cases and data model

Taking the recommendations from the gap analysis into account, the first version of the TENCompetence Personal Competence Manager (PCM) was developed, which implements the use cases and data model that are described in this chapter. The use cases in this chapter are elaborations of the initial requirements and the high level use cases in previous chapters. This chapter also contains a summary of the data model that is the current reflection of the relation between the Domain Model and the first version of the TENCompetence Personal Competence Manager (PCM).

4.1 Elaborated use cases

The high level use cases described above have sub-ordinate tasks assigned. In order to map these high level use cases to the technical constraints for a personal competence management, a set of nine use cases have been identified. These use cases refer to technical aspects that are shared across the different high level use cases. These use-cases are shown in figure 2. The following sections describe these use cases in greater detail.

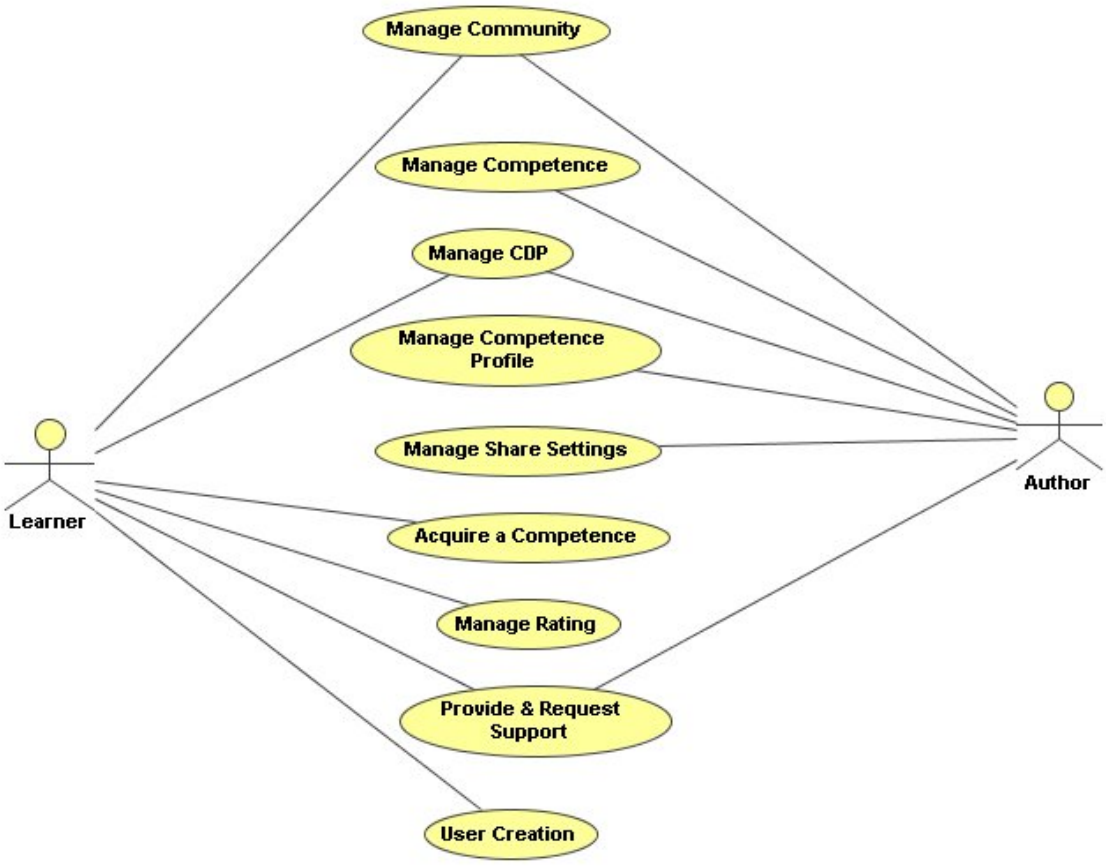


Figure 3: Elaborated use cases

Annex D provide a more detailed perspective for each of the nine elaborated use cases. Each use case description has the following parts.

1. The general objective for the use case from the perspective of the main user
2. A UML diagram of the use case of sub-ordinate use cases
3. All actors that are directly involved in the use case
4. For each sub use case a UML activity diagram
5. For each sub use case a brief narrative description of the activity flow.

4.2 Data Model

The data model provides a different view on a software system's organisation. While activity diagrams show processes, workflows, and interactions between the different components of the system, the data model shows the relation between the data objects that are created, processed, and managed by the system. In that sense, the data model provides a view on the organisation and structure of a software system, whereas the activity diagrams show the dynamics within it.

The following sections show the different levels of abstractions of TENCompetence's PCM. First, the *domain model* – elaborated above – describes the relation of components and concepts. This model provides the conceptual structure of the PCM. Second, the *database model* describes the internal organisation of data in terms of a database. The database model shows dependencies and relations between data objects and allows estimating potential aggregations and usages of the data to the level of atomic data fields. Third, the *data model* shows the relations between the different data objects provided by the TENCompetence services. The data model links the domain model with the database model.

4.2.1 Database model

A database model is a theory or specification describing how a database is structured and used. The fundamental assumption of the relational model is that all data is represented as mathematical *n*-ary relations, an *n*-ary relation being a subset of the Cartesian product of *n* domains.

The relational model of data permits the database designer to create a consistent, logical representation of information. Consistency is achieved by including declared ***constraints*** in the database design, which is usually referred to as the logical schema. A table is an accepted visual representation of a relation.

The schema below represents the database model of the first release of the TENCompetence Personal Competence Manager (PCM). The different tables represent the objects that can be found in the PCM and the relations between them. In the future releases of the PCM, this database model will change including new tables that will add functionalities to the system.

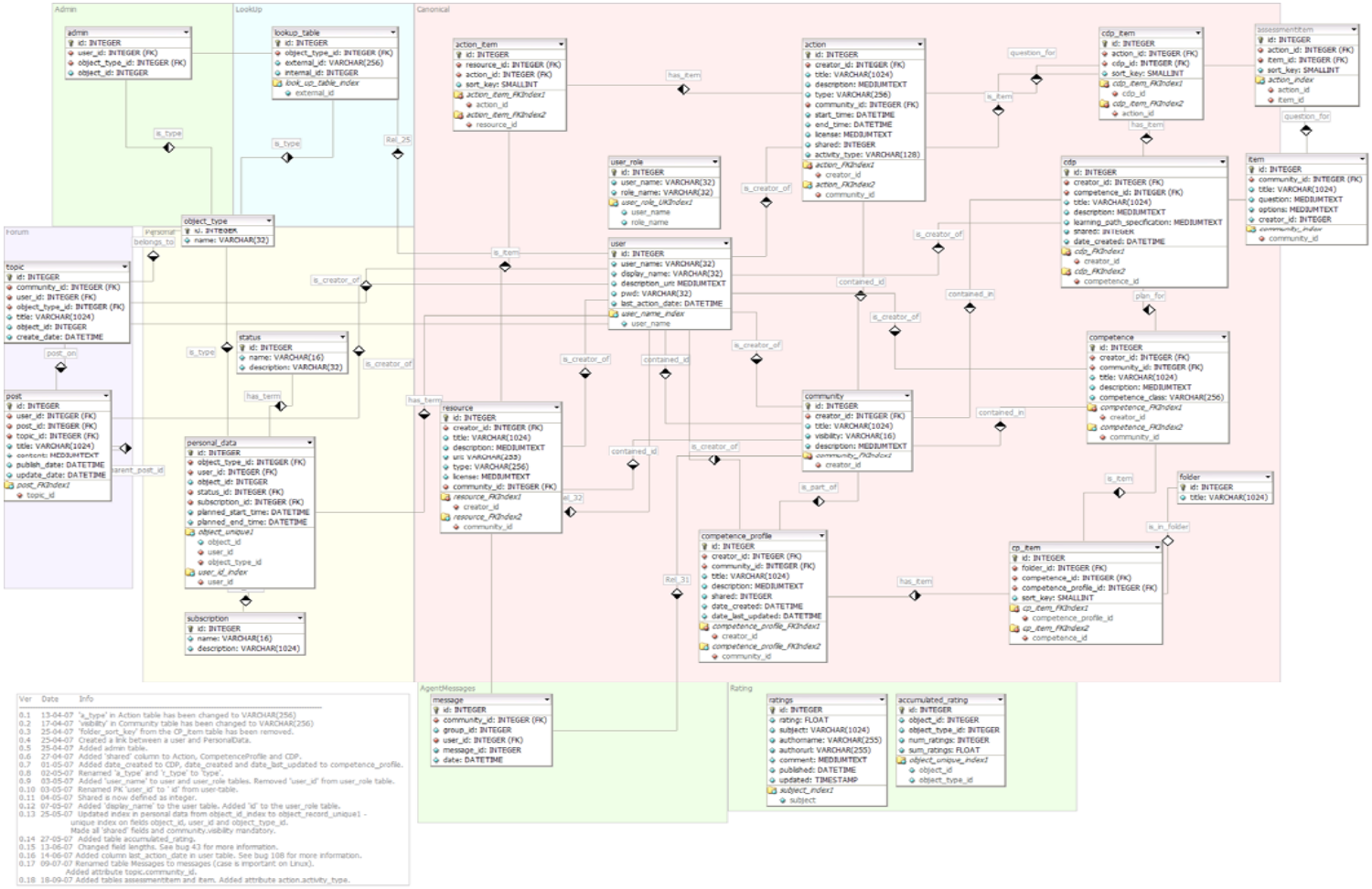


Figure 4: Database Model

4.2.2 Data Model

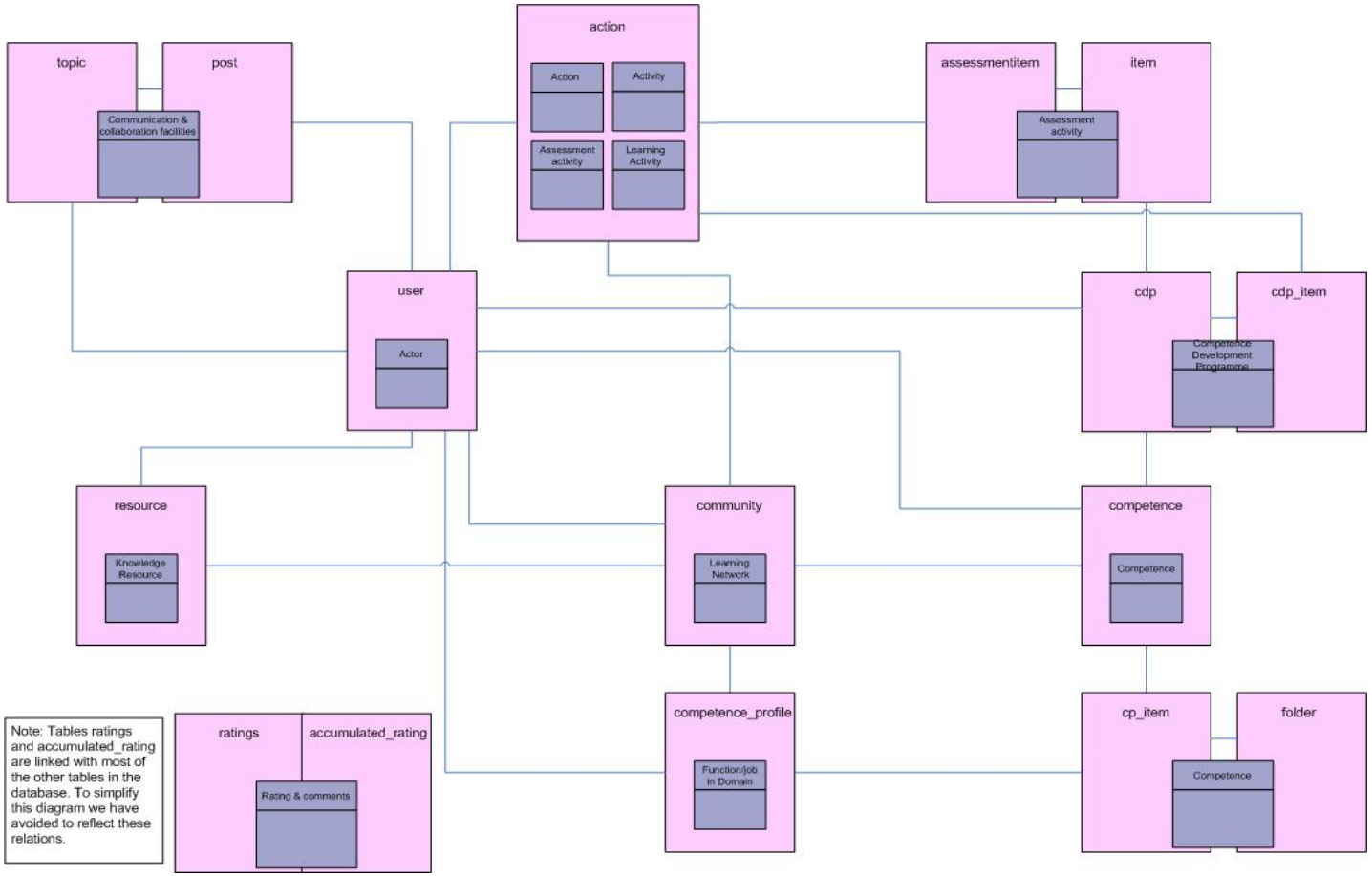


Figure 5: Data Model

The diagram above is meant to be a link between the domain model and the database model, a diagram that can be used for a better understanding of the current implementation of the TENCompetence Personal Competence Manager (PCM) in relation with the last version of the domain model.

This diagram, called Data Model, shows in a very visual and graphical way, the classes and entities of the domain model that have already been implemented in the current version of the PCM, and that have their correspondence in the database model of the PCM. Across this diagram, the correspondences between the classes of the domain model and the tables of the database model can be seen.

The system of TENCompetence, the PCM, will produce some releases before its last version, in the first one of all of them, the current one, only some of the functionalities described in the objectives of the project and some parts of the high level use cases have been implemented, for this reason it hasn't been necessary to implement all the classes of the domain model in the database model of the PCM.

In *'Figure 6: Classes of the Domain Model'* there is a list of the classes that appear in the Domain Model but don't have correspondence in the Database Model, and therefore they won't have their correspondence in the PCM.

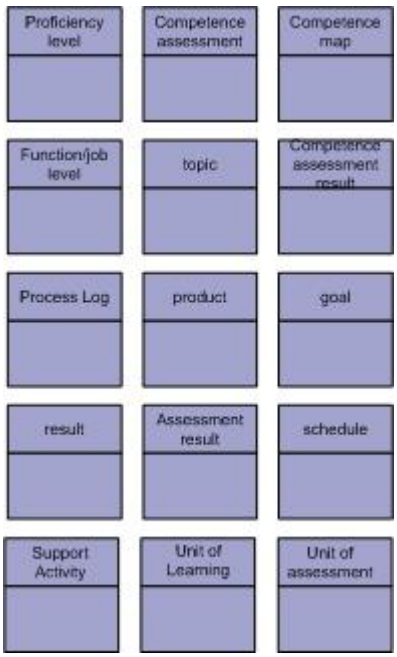


Figure 6: Classes of the Domain Model

On the same way that some classes of the domain model are not reflected in the database model, some of the tables of the database model do not have correspondence in the domain model. In order to develop a database model complete and consistent, some administrative or auxiliary tables have been used, in 'figure 7: Tables of the Database Model', are described the tables just mentioned.

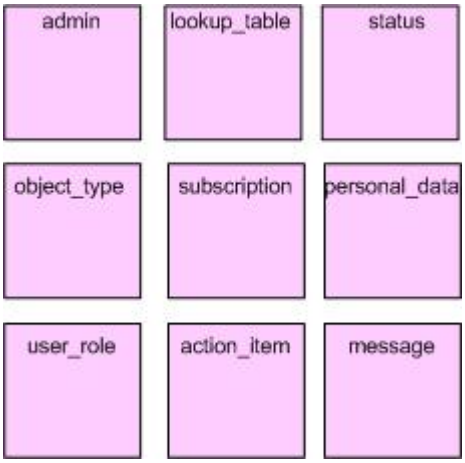


Figure 7: Tables of the Database Model

This work was implemented in the first version of the Personal Competence Manager (PCM), as it was published on the project website (http://www.tencompetence.org/).

5 Future work: Research Roadmap

After having described the requirements work up to this point in the project, this chapter contains a research roadmap that forecasts the future functionality that will be progressively incorporated into the new versions of the integrated system and a time-plan for the delivery of these releases.

5.1 Introduction

The functionally detailed in the previous chapters is being developed through the different pre-releases of the PCM (0.1 to 0.9) and will be consolidated in its first Public Release (PCM 1.0). Such as functionality constitutes the starting point of the research roadmap of TENCompetence for the following development cycles.

To create this roadmap, the development plans of each Aspects Work packages (WP5: Knowledge Management, WP6: Learning Units and Path, WP7: Competence Development Programs and WP8: Learning Networks) have been analysed together with the new functionality, from their research work, they are planed to incorporate in each of the anticipated PCM releases.

Regarding, the anticipated PCM releases, we have decided to use the following numbering convention:

- Pre-releases (x.9 versions)
- Public releases (x.0 version)
- Patch release (x.y version) Patches are only supplied if necessary.

Regarding public releases, following are the main planed ones as well as an estimated schedule for their delivery:

- **Version 1.0: 01.12.2007 (Month 24)**
- **Version 2.0: 04.02.2008 (Month 27)**
- **Version 3.0: 05.05.2008 (Month 30)**
- **Version 4.0: 06.10.2008 (Month 35)**
- **Version 5.0: 02.02.2009 (Month 39)**
- **Version 6.0: 04.05.2009 (Month 42)**
- **Version 7.0: 05.10.2009 (Month 47)**

Following section provide details of the functionality that will be incorporated by each aspect work package to the different Public Releases. These individual plans are consolidated in an integrated one in next section that is completed with a last one including a procedure for proposing, approving and implementing change requests to PCM. It includes the communication mechanisms between aspects work packages and WP2 (in charge of the functional specification of the requests) and WP3 (in charge of the implementation of such as requests in PCM).

5.2 Release Development Roadmap

Following table offers details about the main contribution of each aspect work package to the planed PCM releases:

WP5- KNOWLEDGE MANAGEMENT	
Version 1.0: 1.12.2007	**PCM-base version**
Version 2.0: 04.02.2008	• Initial integration of the services related to the 'discovery' of connections among Knowledge Resources and between KRs and users. • Improved indexing, metadata handling and GUI design.
Version 3.0: 05.05.2008	• Full integration of the services related to the 'discovery' of connections among KRs and between KRs and users.
Version 4.0: 06.10.2008	• Refinement of integration of the services related to the 'discovery' of connections among KRs and between KRs and users after the KRSM 2.0 validation. • Possible integration of new use cases, if any.
Version 5.0: 02.02.2009	• Initial integration of the collaboration cost reduction-related services. • Final integration of the services related to the 'discovery' of connections among KRs and between KRs and users.
Version 6.0: 04.05.2009	• Full integration of the collaboration cost reduction-related services. • Improved GUI design.
Version 7.0: 05.10.2009	• Refinement integration of the collaboration cost reduction-related services after the KRSM 3.0 validation. • Possible integration of new use cases, if any. • Final integration of the collaboration cost reduction-related services • Any other refinement.
WP6- LEARNING UNITS	
Version 1.0: 1.12.2007	• **PCM-base version**
Version 2.0: 04.02.2008	• An initial version of the "widget server" running, providing more flexible runtime services for Learning Design. • A first version of the IMS LD Authoring tool. This will be a working application, but may not yet cover the whole of the specification.
Version 3.0: 05.05.2008	• Tested and validated release of the Widget Server, with a selection of services available • Authoring tool o Tested and validated release of the IMS LD Authoring Tool o QTI authoring tool integrated with the IMS LD Authoring Tool. Detailed plans for the development of this tool have yet to be made. o High level pre-authoring layer leveraging Eclipse cheat sheets and wizards. • Integration with LearnExact.
Version 4.0: 06.10.2008	• New release of the Widget server, with a more complete set of services • Provisioning from PCM to CopperCore. (to be confirmed) • Second release of the LD Authoring Tool with revised interface and more pre-authoring support.
Version 5.0: 02.02.2009	• One possibility is to develop a Widget based IMS LD player, on top of CopperCore. • Needs further discussion.
Version 6.0: 04.05.2009	• Needs further discussion.
Version 7.0: 05.10.2009	• Needs further discussion.

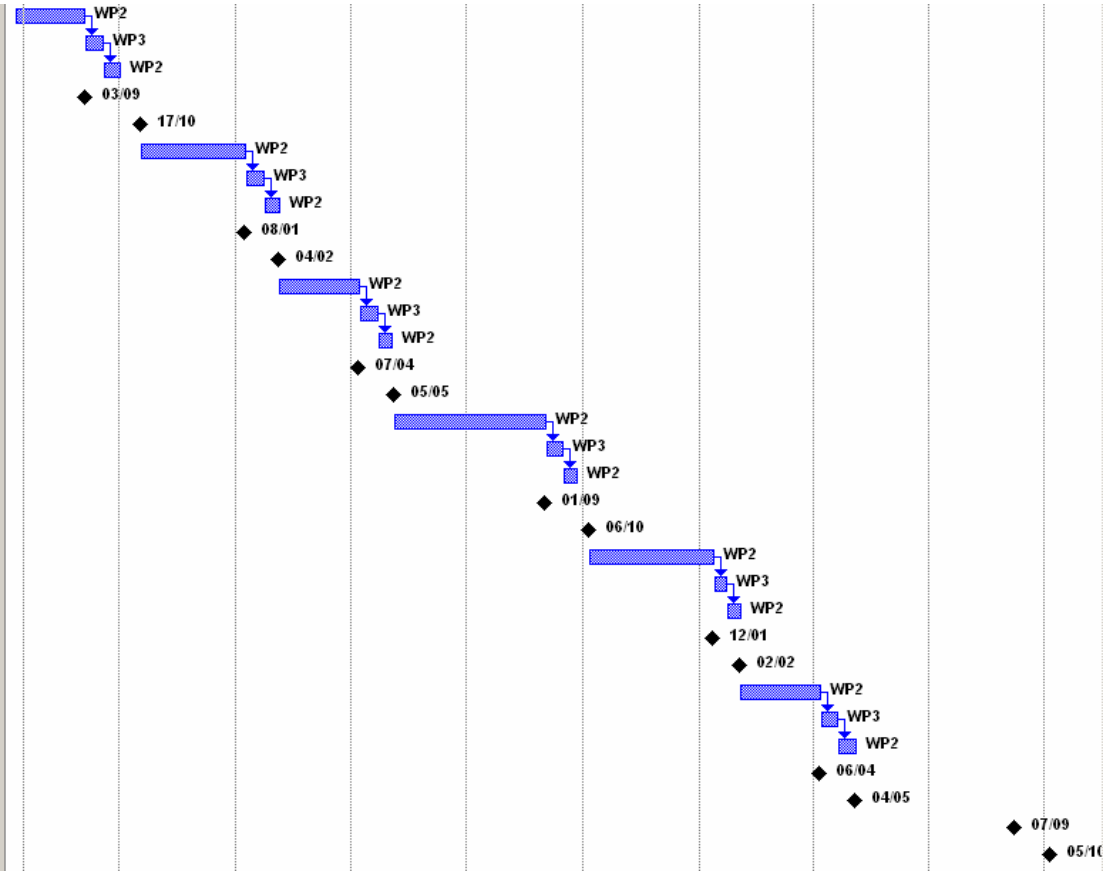
WP7- COMPETENCE DEVELOPMENT PROGRAMS	
Version 1.0: 1.12.2007	• **PCM-base version**
Version 2.0: 04.02.2008	• First iteration for the Personal Development development Programs.
Version 3.0: 05.05.2008	• First iteration for the Personal Development development Programs tuned and evaluated.
Version 4.0: 06.10.2008	• Second iteration: positioning and navigation services; integration of Skylining (preference-based navigation support). • Second iteration: graphical support for creating, editing and using curriculum development programs. • Most likely implementation of the Learning Path Description; data structure to be used for the tools described above.
Version 5.0: 02.02.2009	• To be determinated
Version 6.0: 04.05.2009	• To be determinated
Version 7.0: 05.10.2009	• To be determinated.
WP8- LEARNING NETWORKS	
Version 1.0: 1.12.2007	• **PCM-base version**
Version 2.0: 04.02.2008	• First iteration of the Overview tool. • Network management tool (it will allow ad hoc transient communities to be created for a second learner support purpose). • Static network policies. • Extensive access to log data and portfolio (read and write) on top of the functionalities listed above, such as: o Content competence. o Tutor competence. o Eligibility. o Availability. o Be able to determine status of resources. o Email. o Wiki. • Question form.
Version 3.0: 05.05.2008	• First iteration of the Overview tool tuned and evaluated: including a selection of visualization, agent and games, validated network management tool (with ad hoc transient communities). The precise selection will depend of the server-data available
Version 4.0: 06.10.2008	• Second iteration of the Overview tool :More dynamic application of network management policies. • Additional games, agents and visualizations.
Version 5.0: 02.02.2009	• A tested and validated released. • Add additional games, agents and visualizations.
Version 6.0: 04.05.2009	• Add additional games, agents and visualizations to be decided
Version 7.0: 05.10.2009	• Add additional games, agents and visualizations to be decided

	WP5	WP6	WP7	WP8
V1	• **PCM-base version**	• **PCM-base version**	• **PCM-base version**	• **PCM-base version**
V2.0	• Initial integration of the services related to the 'discovery' of connections among Knowledge Resources and between KRs and users. • Improved indexing, metadata handling and GUI design.	• An initial version of the "widget server" running, providing more flexible runtime services for Learning Design. • A first version of the IMS LD Authoring tool. This will be a working application, but may not yet cover the whole of the specification.	• First iteration for the Personal Development development Programs.	• First iteration of the Overview tool. • Network management tool (it will allow ad hoc transient communities to be created for a second learner support purpose). • Static network policies. • Extensive access to log data and portfolio (read and write) on top of the functionalities listed above, such as: o Content competence. o Tutor competence. o Eligibility. o Availability. o Be able to determine status of resources. o Email. o Wiki. • Question form.
V3	• Full integration of the services related to the 'discovery' of connections among KRs and between KRs and users.	• Tested and validated release of the Widget Server, with a selection of services available • Authoring tool o Tested and validated release of the IMS LD Authoring Tool o QTI authoring tool integrated with the IMS LD Authoring Tool. Detailed plans for the development of this tool have yet to be made. o High level pre-authoring layer leveraging Eclipse cheat sheets and wizards. • Integration with LearnExact.	• First iteration for the Personal Development development Programs tuned and evaluated.	• First iteration of the Overview tool tuned and evaluated: including a selection of visualization, agent and games, validated network management tool (with ad hoc transient communities). The precise selection will depend of the server-data available

V4	• Refinement of integration of the services related to the 'discovery' of connections among KRs and between KRs and users after the KRSM 2.0 validation. • Possible integration of new use cases, if any.	• New release of the Widget server, with a more complete set of services • Provisioning from PCM to CopperCore. (to be confirmed) • Second release of the LD Authoring Tool with revised interface and more pre-authoring support.	• Second iteration: positioning and navigation services; integration of Skylining (preference-based navigation support). • Second iteration: graphical support for creating, editing and using curriculum development programs. • Most likely implementation of the Learning Path Description; data structure to be used for the tools described above.	• Second iteration of the Overview tool: More dynamic application of network management policies. • Additional games, agents and visualizations.
V5	• Initial integration of the collaboration cost reduction-related services. • Final integration of the services related to the 'discovery' of connections among KRs and between KRs and users.	• One possibility is to develop a Widget based IMS LD player, on top of CopperCore. • Needs further discussion.	• To be determinated	• A tested and validated released. • Add additional games, agents and visualizations.
V6	• Full integration of the collaboration cost reduction-related services. • Improved GUI design.	• Needs further discussion.	• To be determinated	• Add additional games, agents and visualizations to be decided
V7	• Refinement integration of the collaboration cost reduction-related services after the KRSM 3.0 validation. • Possible integration of new use cases, if any. • Final integration of the collaboration cost reduction-related services • Any other refinement.	• Needs further discussion.	• To be determinated	• Add additional games, agents and visualizations to be decided

5.3 Detail Planning

Requirement Collection 1.1	mar 10/07/07	lun 03/09/07	40 días	WP2
Work load estimation	mar 04/09/07	lun 17/09/07	10 días	WP3
Update Roadmap	mar 18/09/07	lun 01/10/07	10 días	WP2
Release 0.9	lun 03/09/07	lun 03/09/07	0 días	WP3
Release 1.0	mié 17/10/07	mié 17/10/07	0 días	WP3
Requirement Collection 2.0	mié 17/10/07	mar 08/01/08	60 días	WP2
Work load estimation	mié 09/01/08	mar 22/01/08	10 días	WP3
Update Roadmap	mié 23/01/08	lun 04/02/08	9 días	WP2
Release 1.0.9	mar 08/01/08	mar 08/01/08	0 días	WP3
Release 1.1	lun 04/02/08	lun 04/02/08	0 días	WP3
Requirement Collection 3.0	lun 04/02/08	lun 07/04/08	46 días	WP2
Work load estimation	mar 08/04/08	lun 21/04/08	10 días	WP3
Update Roadmap	mar 22/04/08	vie 02/05/08	9 días	WP2
Release 1.9	lun 07/04/08	lun 07/04/08	0 días	WP3
Release 2.0	lun 05/05/08	lun 05/05/08	0 días	WP3
Requirement Collection 4.0	lun 05/05/08	lun 01/09/08	86 días	WP2
Work load estimation	mar 02/09/08	lun 15/09/08	10 días	WP3
Update Roadmap	mar 16/09/08	vie 26/09/08	9 días	WP2
Release 2.9	lun 01/09/08	lun 01/09/08	0 días	WP3
Release 3.0	lun 06/10/08	lun 06/10/08	0 días	WP3
Requirement Collection 5.0	lun 06/10/08	lun 12/01/09	71 días	WP2
Work load estimation	mar 13/01/09	jue 22/01/09	8 días	WP3
Update Roadmap	vie 23/01/09	lun 02/02/09	7 días	WP2
Release 3.9	lun 12/01/09	lun 12/01/09	0 días	WP3
Release 4.0	lun 02/02/09	lun 02/02/09	0 días	WP3
Requirement Collection 6.0	lun 02/02/09	lun 06/04/09	46 días	WP2
Work load estimation	mar 07/04/09	lun 20/04/09	10 días	WP3
Update Roadmap	mar 21/04/09	lun 04/05/09	10 días	WP2
Release 4.9	lun 06/04/09	lun 06/04/09	0 días	WP3
Release 5.0	lun 04/05/09	lun 04/05/09	0 días	WP3
Release 5.9	lun 07/09/09	lun 07/09/09	0 días	WP3
Release 6.0	lun 05/10/09	lun 05/10/09	0 días	WP3



6 Change Request Procedure

This final chapter details the mechanisms that will facilitate the communication between the different development and research groups and the management of change requests for the system functionality.

6.1 Introduction

A change request in TENCompetence can be understood as a request for incorporating new functionalities to PCM or a request to modify or to improve existent ones. Considering this definition we can consider following categories:

- Feature requests
- Improvements of existing features (while the function remains)
- Integration requests (of other work-packages).

This definition explicitly excludes request for solving Bugs that have to be directly managed by the development team of WP3.

Next paragraphs describe the process to manage such as a kind of change request including the roles of the different TENCompetence's actors, the communication mechanism and the workflow.

6.2 Actors

WP2	Tasks: • Collect, analyse and prioritize change requests assigning each one to corresponding integrated system release • Translate change requests into functional design • Send change requests to WP3 for their implementation and check that new functionalities have been included in the planned release • Maintain release roadmap map and functional design as well as the high-level data model. Outputs: • Functional design for each change request • Updated Research Roadmap
WP3	Tasks: • Estimate workload for each requests • Implement and test the system • Maintain the physical data model • Handle system bugs. Outputs • Estimated workload for each change request • Software and system releases Special Situations: • WP3 will raise new change request when they detect that reported bugs are not really bugs. WP2 will handle these change requests using the normal procedure we describe below.
Other WPS	Tasks: • Provide change requests • Clarify the functional design of the change requests when it is required from WP2 • Verify and Validate the implementation of their requests
Vision Group	Tasks: • Extend existing high level use cases • Set development objectives for the future releases • Provide new high-level use cases • Maintain the domain model.

6.3 Roadmap Elaboration Procedure

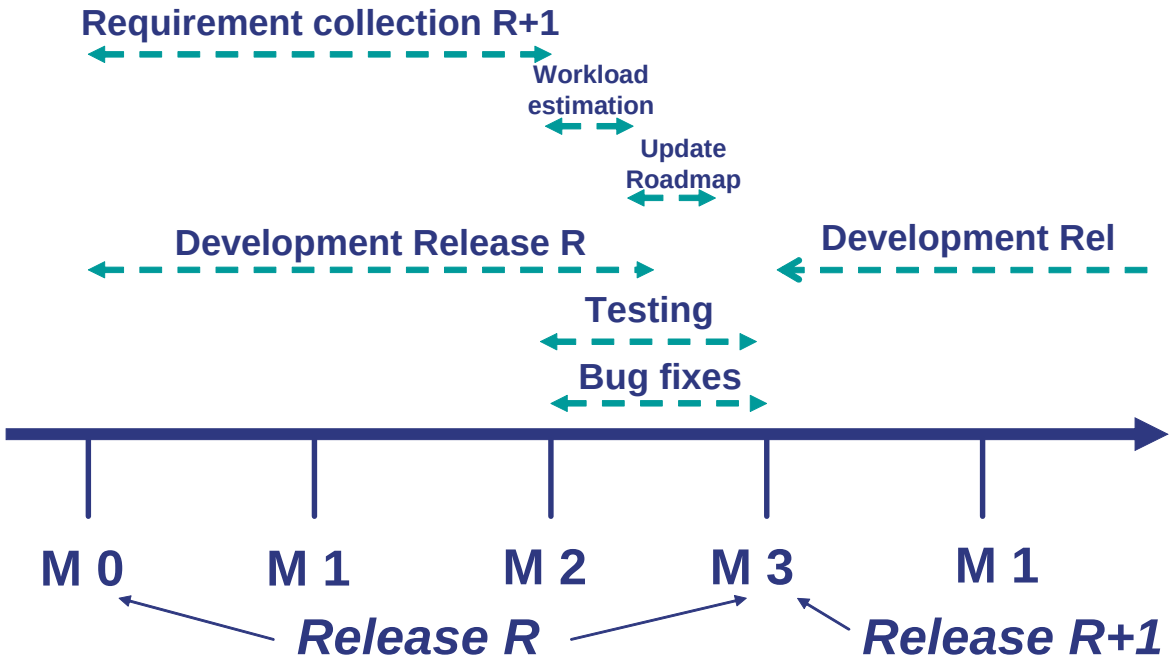
The main steps to elaborate the roadmap are:

- WP2 collects change requests as we define above.
- WP2 assigns priority of change request based in:
  - High level use-cases.
  - Project objectives.
    (In some cases, priority will be decided by Vision Group)
- WP2 translates change request into functional design (extend ID2.5)
- WP3 estimates workload for change request
- WP2 updates the release roadmap.

According this flow, we have defined the next tasks to elaborate and update the roadmap.

Task	Description	Resp	Subtasks	Input	Output
Requirement Collection	In this task, WP2 collects all the change requests (New feature, bugs, improvements and WP request for integration), it analyze and prioritize to enable the estimation of the workload needed.	WP2	• Inform other WPs. • Collecting requirements and requests from other WPs. • Send a reminder to WPs. • Analyzing according High Level Use Cases. • Setting priorities to the Change requests. • Elaborate Use Cases and Activity diagrams. • Send Change Requirements Proposal to WP3.	Change Requests from WPs. (from WP4-8)	*Change* Requirements *Proposal.* (to WP3)
Workload estimation	In this task, WP3 makes an estimation of the workload needed to perform the change request described in the *Change Requirements Proposal*.	WP3	• Estimate workload for each change request contained in Change Requirements Proposal. • Send workload estimation document to WP2.	*Change Requirements Proposal*	*Workload estimation* (to WP2)
Update Roadmap	In this task, WP2 select the change requests to be included in the next release taking in account the *workload estimation* from WP3 and the priorities.	WP2	• Select the high priority change request according the available resources. • Update the roadmap document. • Send release requirements document to WP3.	• *Change* Requirements *Proposal.* • Workload *estimation*	*Release Requirements.* (to WP3)

6.4 Inter-Release planning



Requirement Collection for Release R+1 is made in the first two month of the Release R development. The last month is used to estimate the workload and update the roadmap.

6.4.1 Handling of Change Requests Procedure

All the changes need in the specifications and requirements will be handled using the next procedure.

The Change request can come from everyone. We distinguish three principal sources of change requests:

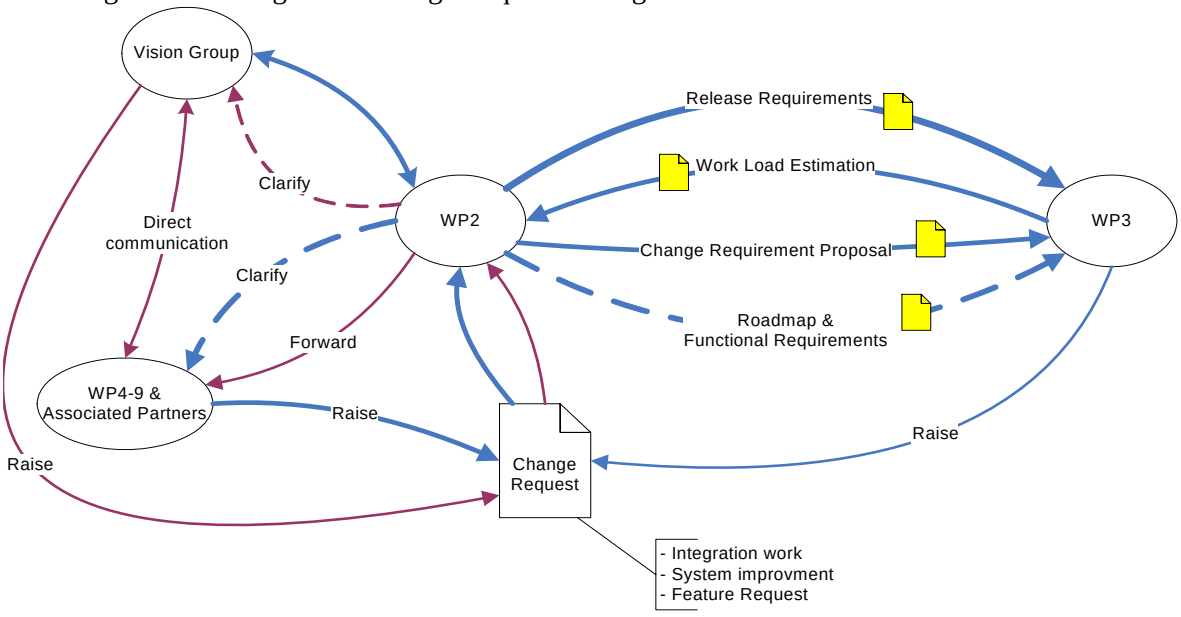
- WP3
- WP4-9 and Associated partners
- Vision Group.

In all the cases, the change request will be specified using a change request form that we will provide.

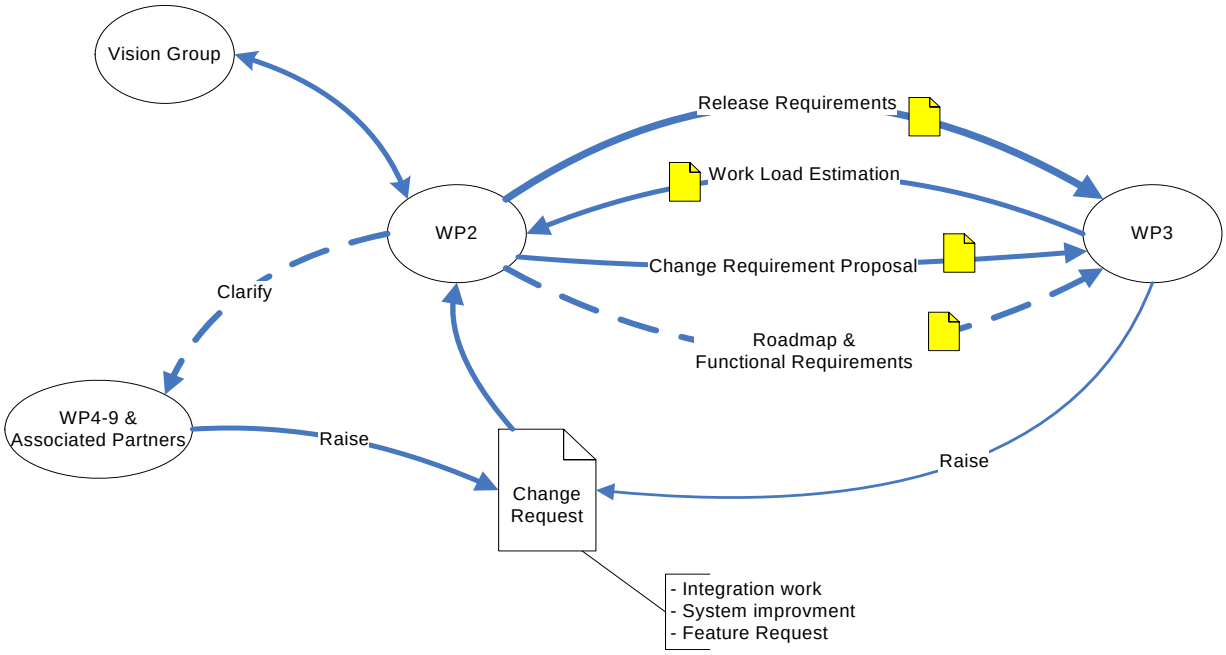
The general sequence of event is as follows:

- Someone raise the change request using the form and send it to WP2.
- WP2 evaluates and ask for clarifications if it needed.
- WP2 forwards to the appropriate WP the change request that doesn't belong to WP3.
- WP2 asks for Vision Group clarification if the change request is out of scope.
- WP2 elaborates a Change Requirement Proposal and sends it to WP3 for evaluation.
- WP3 elaborates a workload estimation for the implementation of the change request and sends it to WP2.
- WP2 evaluate the priority of the different change request.
- WP2 elaborates the release requirements using the workload estimations and the priorities and sends it to WP3.
- Finally WP2 will check if the request is implemented in the planned release.

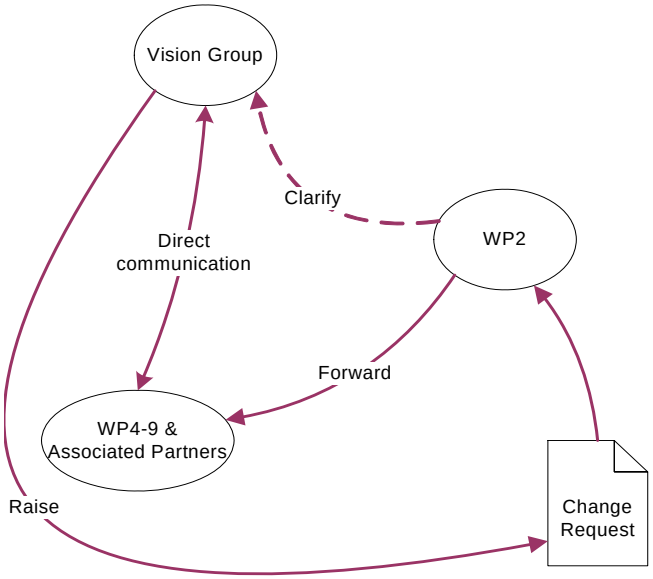
Next diagram shows a general Change Request Management Flow:



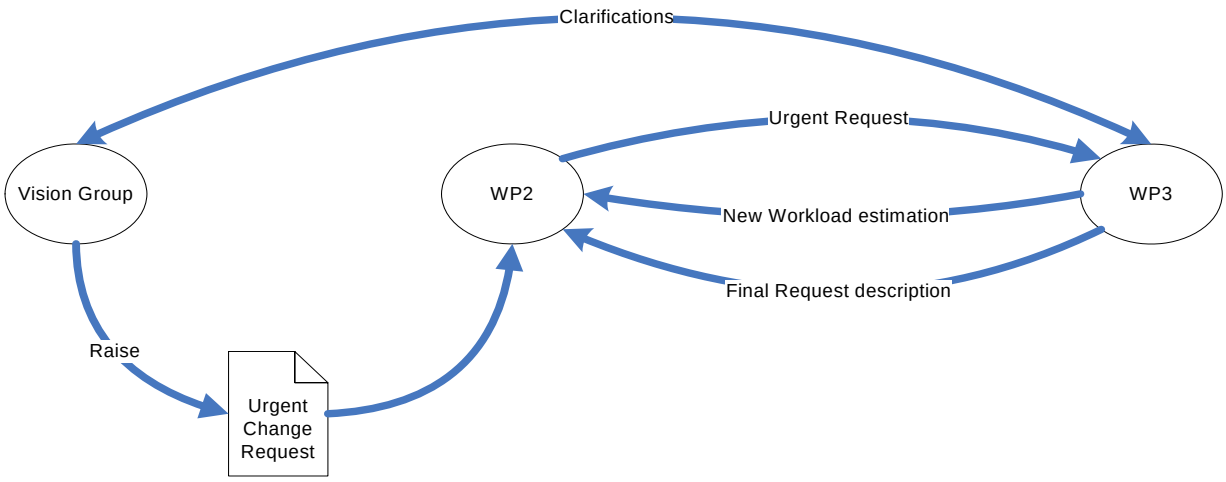
The usual case will be a change request related with WP3.



There is a special case when the Vision Group raises a change request that is not related with WP3. In this case, WP2 forward the request to the right WP after clarify the scope of the request. From this moment, WPx and the Vision Group must establish a direct communication to further clarify the request without WP2 intervention.



For special and urgent change request, we will follow the next procedure. We have to keep in mind that this urgent request will only come from Vision Group and they will not be allow from anyone else.



The main difference in this procedure is that clarification of the request will be done directly between Vision Group and WP3 to avoid delays in communication. After getting the complete picture of the change request, WP3 will send this information and the new workload estimations to WP2 to be able to update the release roadmap.

6.5 Detail Work Flow for Change Request Handling