



Project no. FP6-028038

## PALETTE

Pedagogically sustained Adaptive LEarning Through the exploitation of Tacit and Explicit knowledge

Instrument: Integrated Project

Thematic Priority: Technology-enhanced learning

D.IMP.03 – Revised specifications of services and guidelines for services orchestration

Due date of deliverable: 31 May 2007  
Actual submission date: 15 July 2007

Start date of project: 1 February 2006

Duration: 36 months

Organization name of lead contractor for this deliverable: UNIFR

<b>Project co-funded by the European Commission within the Sixth Framework Programme</b>		
<b>Dissemination Level</b>		
<b>R</b>	<b>PUBLIC</b>	<b>PU</b>

Keyword List: integration of services, interaction of services, participatory design methodology, categorization of CoPs needs, functional specifications  
Responsible Partner: Bernadette Charlier, UNIFR

<b>MODIFICATION CONTROL</b>			
Version	Date	Status	Modifications made by
1	03 April 2007	Draft	Nathalie Deschryver
2	16 April 2007	Draft	Nathalie Deschryver, Géraldine Vidou and Dorel Gorga
3	18 April 2007	Draft	Nathalie Deschryver
4	9 May 2007	Draft	Yannick Naudet
5	25 May 2007	Draft	Nathalie Deschryver, Aida Boukkotaya, Dorel Gorga, Amaury Daele
6	29 May 2007	Draft	Nathalie Deschryver, Bernadette Charlier
7	30 May 2007	Draft	Nathalie Deschryver, Aida Boukottaya, Dorel Gorga
8	31 May 2007	Draft	Aida Boukottaya
9	6 June 2007	Draft	Sandy El Helou, Liliane Esnault, Frédéric Vermeulin, Romain Zeiliger
10	9 June 2007	Draft	Liliane Esnault, Romain Zeiliger
11	13 June 2007	Draft	Nathalie Deschryver, Bernadette Charlier, Romain Zeiliger
12	18 June 2007	Draft	Amaury Daele, Nathalie Deschryver
13	20 June 2007	Draft	Bernadette Charlier, Aida Boukottaya, Sandy El Helou, Yannick Naudet, Nathalie Deschryver
14	22 June 2007	Draft	Nathalie Deschryver, Aida Boukottaya Amaury Daele
15	4 July 2007	Version for evaluation	Nathalie Deschryver, Etienne Vandeput, Dorel Gorga, Bernadette Charlier, France Henri, Amaury Daele, Aida Boukottaya, Sandy El Helou, Géraldine Vidou
16	10 July 2007	Final	Nathalie Deschryver, Bernadette Charlier, Vincent Quint, Amaury Daele

#### **Deliverable manager**

- Nathalie Deschryver (UNIFR)

#### **List of Contributors**

- Aida Boukottaya (UNIFR), Bernadette Charlier (UNIFR), Amaury Daele (UNIFR), Nathalie Deschryver (UNIFR), Sandy El Helou (EPFL), Liliane Esnault (EM-Lyon), Dorel Gorga (UNIFR), France Henri (TELUQ), Yannick Naudet (CRP-HT), Etienne Vandeput (ULG), Frédéric Vermeulin (Gate-CNRS), Géraldine Vidou (CRP-HT), Romain Zeiliger (Gate-CNRS)

#### **List of Evaluators**

- Bernadette Charlier (UNIFR), Vincent Quint (INRIA)

#### **Summary**

The purpose of this deliverable is to set the context for the provision of the revised functional specifications of PALETTE services and the guidelines for the orchestration of services. In practical terms, PALETTE services' functional specifications are produced through an incremental and participative process that: specifies and categorizes CoPs' needs (this includes the analysis of CoPs practices, resources and environments), and refines PALETTE services functional specifications in order to identify specific functions that meet CoPs needs. The categorization of CoPs needs, as well as the functional specification of services, are aimed at offering a common understanding between PALETTE developers and CoPs' mediators and members. This enables the efficient identification of possible interactions between PALETTE services' categories that will be used as the basis to enhance the development guidelines produced in D.IMP.01 and will be related to interoperability issues (this enhancement will be introduced in D.IMP.04), as well as the suggestion of future strategies for the elaboration of services uses.

## Table of content

<b>1</b>	<b>– Introduction</b> .....	<b>4</b>
<b>2</b>	<b>– Foreword: Main Concepts</b> .....	<b>4</b>
<b>3</b>	<b>– Methodology for Producing Functional Specifications and Scenarios</b> .....	<b>7</b>
3.1	Working in Teams: Use-cases and Scenarios .....	7
3.2	Beyond Use-cases, Scenarios and Current Functional Specifications: .....	9
<b>4</b>	<b>– Categorization of CoPs’ Needs</b> .....	<b>11</b>
<b>5</b>	<b>– Interaction between PALETTE Services</b> .....	<b>15</b>
5.1	Addressing CoPs Needs through Services Interaction.....	15
5.2	Classification of Services Interaction .....	16
5.3	Challenges with Information Exchange.....	16
5.4	Challenges with Services Integration.....	17
5.5	Future considerations .....	17
<b>6</b>	<b>– Conditions for Acceptance of the Services by Users: Addressing Acceptability Issues in PALETTE</b> .....	<b>18</b>
<b>7</b>	<b>– Conclusion</b> .....	<b>21</b>
<b>8</b>	<b>– References</b> .....	<b>21</b>
<b>9</b>	<b>– Appendices</b> .....	<b>22</b>
9.1	Appendix 1 – Template for the functional specifications .....	22
9.2	Appendix 2 – Illustration of the functional specifications of a service: Amaya .....	23
9.3	Appendix 3 – Template for the scenario .....	27
9.4	Appendix 4 – Questions related to integration issues .....	29
9.4.1	Discussion for ePrep.....	29
9.4.2	Discussion for @pretic .....	30
9.4.3	Discussion for Form@HETICE.....	31
9.4.4	Discussion for Did@ctic.....	32
9.4.5	Discussion for Learn-Nett.....	33

## 1 – Introduction

The purpose of this deliverable is set the context for the provision of the revised functional specifications of PALETTE services and the guidelines for the orchestration of services. As described in the DoW, it concerns the task 4 of the WP5. *“Based on developments made on WP2, WP3 and WP4, as well as on the pedagogical scenarios established in WP1, this task will contribute to refining the specifications of services and scenarios of use that will serve as a base for experimentation.”*

In practical terms, PALETTE services functional specifications are produced through an incremental and participative process that:

- specifies and categorizes CoPs’ needs (this includes the analysis of CoPs practices, resources and environments)
- refines PALETTE services functional specifications in order to identify specific functions that meet CoPs needs.

The categorization of CoPs’ needs, as well as functional specification of services, is aimed at offering a common understanding between PALETTE developers and CoPs’ mediators and members. This enables the efficient:

- Identification of possible interactions between PALETTE services’ categories that will be used as the basis to enhance the development guidelines produced in D.IMP.01 and related to interoperability issues. This enhancement will be introduced in D.IMP.04;
- Suggestion of future strategies for the elaboration of services uses.

As stated earlier, the purpose of this deliverable is to first describe the adopted methodology for producing categorizations of CoPs’ needs and their relationship with the functional specification of services. The outcomes of this report consist in a categorization of CoPs’ needs and an analysis of their related PALETTE services.

## 2 – Foreword: Main Concepts

In order to offer a good understanding of the deliverable, it’s important to define some used concepts:

- **Service:** From a general point of view, a service is a single or a set of software components or facilities or activities that a third-party offers (temporarily) to whoever is interested, to answer a specific need. A service is bound to some constraints due to the context in which it is executed, comprising the targeted client, the kind of service, environmental constraints, usability, specific requirements, etc.
- **PALETTE services:** In the context of PALETTE, targeted users are CoP’s members, and a service must answer to one or multiple CoP’s needs. We consider two types of PALETTE services, technological and learning ones, both for supporting the development and learning of CoP members. PALETTE Technological services are organised in three categories: information, knowledge management and collaboration services. Such services can be operationalised by one or a composition of functions offered by IT Tools. Learning services are aimed at supporting CoPs in defining their development needs, supporting their members’ learning, and at assisting them in the choice, appropriation and adaptation of technological services. Learning services as they are conceived now, are operationalised by methodological instruments and various strategies. Technological components might be added to increase their accessibility and usability.
- **Function:** a function offered by a service is basically an action that the end-user can perform. It answers a specific need and satisfies the constraints linked to the related service. Functions

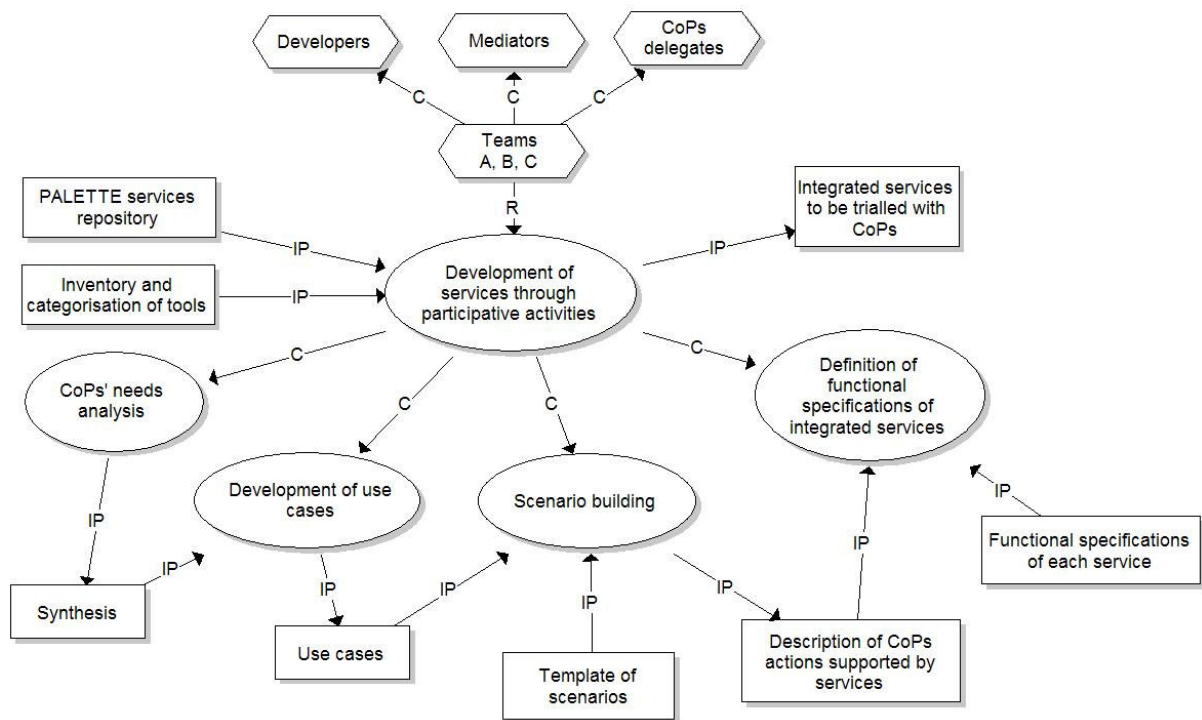
materialize services as they are actually the executive entity providing the service. Functions are not necessarily atomic and can be composed of other sub-functions.

- **Functional specification of a service:** the functional specification gives an overview of a given PALETTE service and the motivation behind its development. It includes : 1) the functions offered by the services from a user's perspective; 2) the characteristics related to the end-user (e.g., expected expertise with software and application domain); 3) functional design considerations that detail the attributes affecting the service functional design (examples: prerequisites for the correct working of the service (e.g., needed operating environments, resource requirements in terms of hardware, other software or equipment, installation, security).
- **Use case:** A use case in the context of PALETTE is made up of a set of possible sequences of interactions between PALETTE services and CoP members in a particular environment (CoP environment) and related to a particular goal (CoP practice). The use case should contain all the steps the user goes through in order to achieve a given goal using a combination of offered services.
- **Scenario:** a scenario in the PALETTE project has the following characteristics: “The contents of the scenarios are descriptions of the activities of the CoPs (collaboration, information use, production of documents, knowledge management...) and their uses of tools within a specific context (history, actors, roles...). The purposes of the scenarios are to encounter the developers' information needs and to present structured information to the CoPs about their functioning and their activities. The lifecycle of the scenarios depends on the different negotiation steps within the participatory activities involving the developers as well as the CoPs. The scenarios will act all along the project as "boundary objects" useful for the negotiation of the scenarios themselves, the experimentation modalities with the CoPs and the development of the CoPs' activities.” (D.PAR.02, p.39-40) In PALETTE we distinguish specific scenario (correspondent and answering the specific needs of a CoP) and generic one (answering similar needs of various CoPs, for instance to manage information)
- **Integration of services:** it's important to distinguish conceptual integration and technical integration. The first one answers the question: How to conceptually integrate PALETTE services together in CoPs practices and environments. This is expressed in CoPs scenarios (from a user perspective). Technical integration consists in the integration of software components that materialise PALETTE services and deals with technical issues such as, for example, the language used in the service description or standards for service interaction.
- **Mediator:** a mediator is a member of PALETTE in charge of the negotiation and the working with one or several CoPs on expected tasks of PALETTE. He develops a privileged relationship with the CoP(S) on one side and with some of the PALETTE services' developers on the other. Some mediators are members of "their" Cops, some not, but all of them have an in depth knowledge about the CoP, its specificities, its activities, the knowledge that may be developed and the processes of interaction and of work in the CoP. He is not a simple "transmitter between the CoP and the project. He is an involved actor, who translates and inscribes the CoP's interests in order to better align them with the interests of the project and of the tools/services. He also has a sufficient knowledge about PALETTE tools (in their current state and in their potentialities), in order to identify possible uses for her CoP, or, at least, to identify the relevant interlocutors if the Cop wants or needs to deepen its knowledge about the possible use of a tool or a set of tool to support some of its activities. He is able to tie the relationships between the relevant persons, activities, functions, etc. He is a "boundary actor" (Esnault, Zeiliger and Vermeulin, 2006) who plays an active part in building and validating boundary objects such as the scenarios, the validating process for the CoP scenarios, the functional specifications of the PALETTE services, and the specification of the necessary interactions between services.

- **Delegate/representative of the CoP:** member of the CoP, possibly member of PALETTE, and privileged interlocutor of the PALETTE partners (mediators, developers, researchers) who gives an account of PALETTE work to the CoP and participates in the design of PALETTE services and scenarios.
- **Developer:** member of PALETTE, in charge of the development of the integrated technological and learning services of PALETTE. He participates in the writing of the scenarios with the mediators and the CoPs' members and is particularly in charge of the writing of functional specifications of the services. He trains as the need arise CoPs' members for the use of the PALETTE services and can advice them about this use according to their needs.

### 3 – Methodology for Producing Functional Specifications and Scenarios

This section describes the adopted methodology in order to produce the functional specifications of PALETTE services as well as CoPs’ scenarios. It essentially outlines the different steps and their related outputs. Functional specifications and scenarios are produced through an incremental and participative design process. Mediators and delegates of the CoPs work together to produce scenarios and collaborate with developers to prepare functional specifications of services related to these scenarios. Figure 1 illustrates the main stages of this work. The stages are based on the synthesis of CoPs needs done in the context of WP1 through several interviews (D.PAR.01). The interviews as well as a synthesis are available on the BSCW as internal reports (<https://bscw.ercim.org/bscw/bscw.cgi/214337>, restricted access for Project members).



Caption:

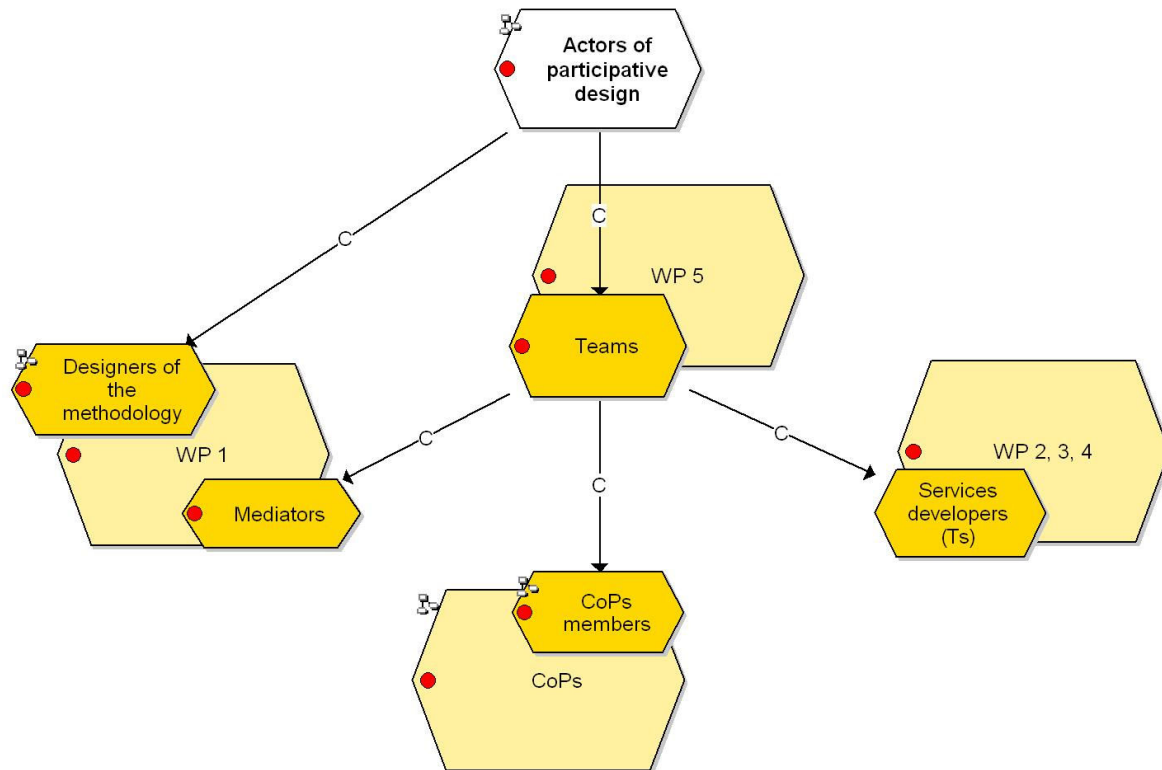
<p>“C” means “is Composed of”                  “IP” means “input product”                  “R” means “regulate”</p>	<p>⬡ = Actor    ○ = Process, action                  □ = Concept, object, product</p>
---	---

Figure 1 – Steps for the development of the services

#### 3.1 Working in Teams: Use-cases and Scenarios

In order to work on the CoPs synthesis to produce functional specifications and scenarios, three teams (A,B,C) are formed on the basis of the first analysis of CoPs’ needs and services that could fulfil these needs. The aim of these teams is to facilitate the flow of crucial information, the coordination of activities, and exchanges between the Research and Development Work Packages (1,2,3,4), as schematized in Figure 2. The teams play a major role in the PD methodology because they gather the various categories of actors and coordinate actions of the PD. More specifically the teams have two major focuses:

- To support the interoperability between PALETTE services: in each team two services-developing partners (for instance WP3 and WP4) at least approach the CoPs together.
- The creation of specific and more generic scenarios: each team addresses two CoPs at least and negotiates two specific scenarios. With the integration of both, a more generic scenario can also be created.



Caption:

“C” means “is Composed of”	= Actor
----------------------------	---------

**Figure 2 – Excerpt from the MOT diagram of the PALETTE R&D methodology**

As schematized in Figure 2, the teams work in the context of the objectives and guidelines of the WP5 (task 4) and are composed of the actors from the WP1 (mediators), the WP2,3,4 (Services developers) and the CoPs (CoPs’ members, CoPs’ delegates).

Table 1 presents these teams in which the CoPs are involved in order to meet their needs by interacting with their mediator and some of the PALETTE technological services developers. Currently WP2, 3 and 4 partners proposed a set of tools (example: Amaya, CoPe\_it!, etc.). Based on the analysis carried out in the present deliverable as well as the model of services description (task 1, WP5), tool developers will specify a set of functions as more specific end-users PALETTE services (to be defined in D.IMP.04).



**Table 1- Synthetic description of the teams**

	<b>Team A</b>	<b>Team B</b>	<b>Team C</b>
<b>CoPs involved</b>	Badge UX11 ePrep Doctoral Program Lancaster	Form@Hetice @PRETIC	Did@ctic Learn-Nett Adira Aradel
<b>Technological Services (tools)</b>	<i>Information services</i> (DocReuse, LimSee3, Amaya)  <i>KM services</i> (Corese, Meat, SeWeSe, SweetWiki)  <i>Collaboration services</i> (e-Logbook)	<i>Information services</i> (LimSee3, Amaya)  <i>KM services</i> (Generis, Corese, Meat, SeWeSe, SweetWiki, BayFac)	<i>Information services</i> (DocReuse)  <i>KM services</i> (LinkWidget, SweetWiki, ECCO)  <i>Collaboration services</i> (e-Logbook, CoPe_it!)

Let us present now the work undertaken in these teams.

1. As schematized in Figure 1, the first task of the teams was the writing of the **use-cases**. Taking into account the CoPs needs (synthesis), the developers wrote use cases and negotiated them with the mediators. A set of use-cases describing the first version of services functional specifications as well as examples of their use within CoPs environments were produced and available as internal reports (<https://bscw.ercim.org/bscw/bscw.cgi/183034> , <https://bscw.ercim.org/bscw/bscw.cgi/173188> , <https://bscw.ercim.org/bscw/bscw.cgi/194052>, restricted access for Project members). Use-cases present also a first attempt to study the possible interactions of PALETTE services.
2. Based on the produced use-cases, first versions of **scenarios** and **functional specifications of services** were produced (examples of functional specifications are available in Appendix 9.2 and on <https://bscw.ercim.org/bscw/bscw.cgi/239915> , restricted access for Project members). A template was provided for each of these documents (see these templates in Appendices 9.1 and 9.3). The contents of these documents are described in the foreword. The functional specifications of services were refined by the developers. Scenarios describe CoPs needs as well as the necessary steps to use PALETTE services by CoPs members. The scenarios were analysed but not presented here. They will be presented in another report (D.PAR.03) with their validation process (examples of scenarios could be found at <https://bscw.ercim.org/bscw/bscw.cgi/236536> , <https://bscw.ercim.org/bscw/bscw.cgi/236541>, <https://bscw.ercim.org/bscw/bscw.cgi/238154>, restricted access for Project members).

### 3.2 Beyond Use-cases, Scenarios and Current Functional Specifications:

The documents mentioned in the previous section (section 3.1) present two main characteristics:

- They are CoP specific: each scenario is dedicated to one specific CoP; however some similarities between scenarios can be observed (e.g. the use of the same services in the same manner and under the same conditions, the same need expressed differently, etc).
- Scenarios describe the use of PALETTE services within CoPs; however we notice that some points regarding the integration of the services in CoPs environments are not addressed. During virtual and face-to-face meetings developers, as well as mediators, have the same questions about integration issues (e.g. which repository CoPs will use?)

What to do with existing resources? , How to switch from existing tools from Palette services?, etc). Moreover, other questions are asked regarding the interaction of services (When do the services interact? How do they interact?) (see in Appendix 9.4).

To go beyond the current documents produced, we try in this deliverable to:

- i. Categorise CoPs needs: this categorization offers a global view of CoPs needs that permits to generalise the use of PALETTE services in order to produce generic scenarios. Moreover, the effort of categorization permits to describe needs that were provided by the synthesis written by WP1 members but not all taken into account in the scenarios.
- ii. Refine the categorization of PALETTE services: through the analysis of the produced scenarios, we also noticed that the specification of PALETTE services could be refined in order to answer efficiently identified categories of CoPs needs. This enables the developers to better identify the offered services (e.g. a specific function of a tool is offered as an end-user service).
- iii. Study the different types of PALETTE services interactions (at conceptual level) which will be a helpful input to enhance the technical guidelines regarding the integration of services (Task1, WP5).
- iv. Suggest future strategies for the development of PALETTE services.

In order to achieve these goals, we proceed in several steps:

- We first design a common template for CoPs. The template summarises the needs of the CoP as well as the PALETTE technological services (expressed in terms of functions) that could meet this need. For each need, the template describes existing resources, environment and practice. Each adopted service is described in terms of changes in existing resources, environment and practice. The latter templates have been designed collaboratively by developers and mediators during face-to-face meetings. Moreover, interactions between services are also presented as well as questions related to the use of services (individually or with other services). Examples of templates are presented in the Appendix 9.4.
- The produced templates are analysed and generalised to refine the categorization of PALETTE services, categorization of PALETTE services interactions and future strategies to develop PALETTE services with CoPs.

In the remaining of this report, the different obtained results are detailed.

## 4 – Categorization of CoPs’ Needs

One of the objectives of PALETTE is to develop “configuration of services” (technological and learning ones) which meet the developmental and learning needs of CoPs. This means that it is important to firstly represent patterns of needs that orient configuration of PALETTE services. In order to find these needs patterns we developed a categorization of the PALETTE CoPs’ needs (to see in use cases and scenarios). In order to produce this categorization we used two models. First, we used the model of professional development within a community of practice, developed by Daele (2006) (see Figure 3). This model constitutes one of the “conceptual” bases of the PALETTE project (see in the DoW). Second, we used the model of CoPs’ actions proposed by Künzel, Charlier & Daele (2007), a model which is anchored in observations of several CoPs in different domains of activity.

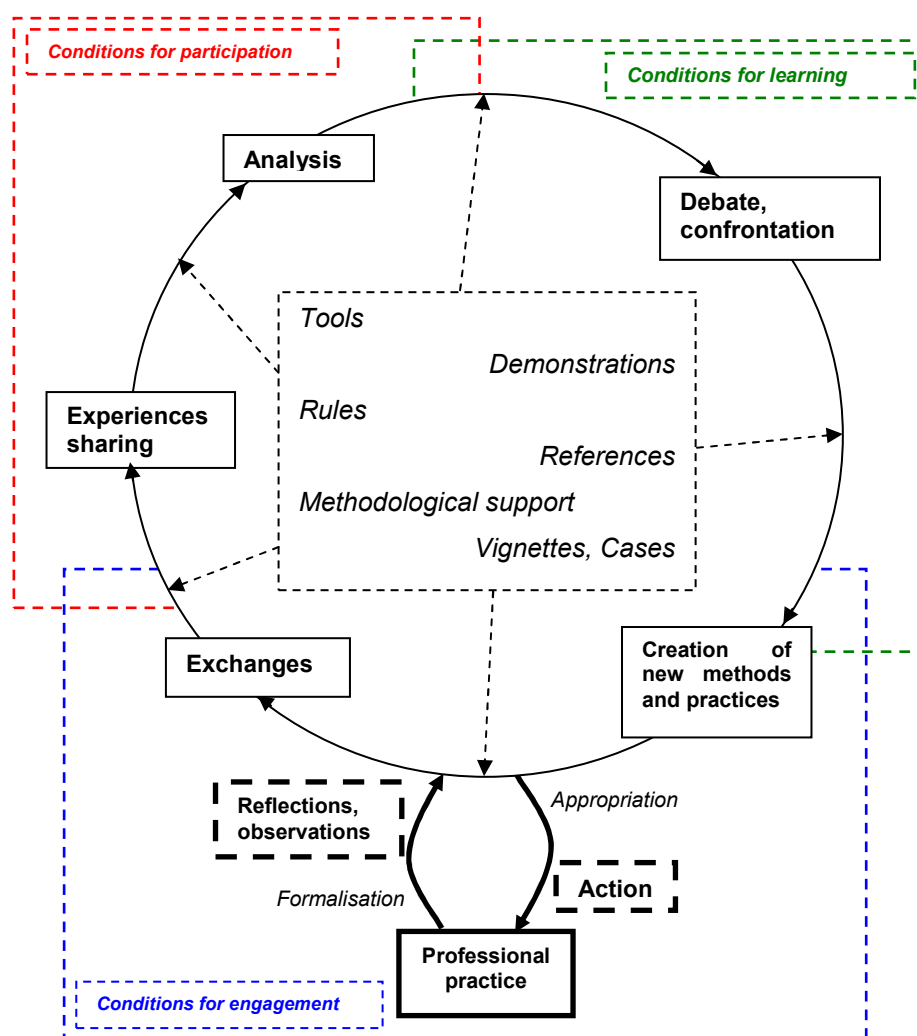


Figure 3 - Model of professional development within a community of practice (Daele, 2006)

Daele’s model (Figure 3) represents the different processes involved in the larger process of professional development. All of these processes occur following a number of conditions linked with: the engagement, the participation (in the various modes of social interactions) and the learning (supported notably by the formalization of the exchanges) in the CoP.

The first two conditions “engagement and participation” appeared in the PALETTE CoPs needs.

As illustrated in Table 2, the need for supporting participation is largely expressed and covers some modes of participation defined in the Daele's model (exchanges, debate and confrontation). We mean by 'participation' the extent to which members are involved in the activities of the CoP more or less actively and the extent to which they interact with other members of the CoP. Consequently to support participation means to support social interactions. De Montmollin (1977) defines 'social interaction' as the effects resulting from the presence, the words and action of a person on the responses of another to his environment. So for supporting social interaction we consider as important to support verbal exchanges (from single exchanges to richer one like debate-confrontation) as the awareness of the presence of the participants. Some other conditions could be associated with the participation like: to give social and technical training for members, to enable them to truly participate.

**Table 2 – Categories of needs of the CoPs**

<b>Categories of needs</b>	<b>PALETTE CoPs Needs</b>
<p>1. To support participation :</p> <p>To support social interactions : verbal interactions (exchanges, experiences sharing, analysis, debate, confrontation, creation of new methods and practices) and presence</p>	<p><b>Doctoral Program Lancaster</b></p> <ul style="list-style-type: none"> <li>- to tell students submit their work and to let tutors annotate them on-line</li> <li>- for discussion : easier to use and understand than LUVLE (institutional platform)</li> <li>- to engage members critically</li> </ul> <p><b>@pretic</b></p> <ul style="list-style-type: none"> <li>- to share information</li> <li>- to improve cooperation</li> </ul> <p><b>Learn-Nett</b></p> <ul style="list-style-type: none"> <li>- Encourage the tutors to share about practice</li> <li>- task sharing, analyzing the project, assessing the project, managing different opinions at a distance, support argumentation</li> </ul> <p><b>Form@Hetice</b></p> <ul style="list-style-type: none"> <li>- To support argued discussion and decision making</li> </ul> <p><b>Didactic</b></p> <ul style="list-style-type: none"> <li>- to support exchanges in discussion groups (f/f meetings)</li> <li>- to support the communication within the communities of practice during distance work periods</li> </ul> <p><b>Adira</b></p> <ul style="list-style-type: none"> <li>- to create documents through debates in f/f and at a distance</li> </ul>
<p>2. To constitute common resources:</p> <p>To formalize tacit knowledge, to archive common resources and to make them retrievable and reusable</p>	<p><b>Doctoral Program Lancaster</b></p> <ul style="list-style-type: none"> <li>- for copyright clearance of articles available in LUVLE</li> <li>- to make documents available in LUVLE easier to access</li> <li>- to archive and make available anonymized students' corrected assignments</li> <li>- to provide online presentation that can be annotated and updated</li> </ul> <p><b>Learn-Nett</b></p> <ul style="list-style-type: none"> <li>- To reuse students' research papers and other documents for the design of tutors' tools and for the work of the coordination team.</li> </ul> <p><b>@pretic</b></p> <ul style="list-style-type: none"> <li>- To structure shared information and resources</li> <li>- to retrieve archive content</li> <li>- to produce web, text and presentation documents easily and collaboratively</li> <li>- To make members aware of the benefit of accessible and retrievable knowledge</li> </ul> <p><b>Form@Hetice</b></p> <ul style="list-style-type: none"> <li>- capitalization of shared documents for reusing (categorization)</li> </ul> <p><b>Didactic</b></p> <ul style="list-style-type: none"> <li>- To capitalize discussions and documents shared during f/f meetings about teaching practices</li> <li>- To reuse illustrations of teaching practices</li> </ul> <p><b>ePrep</b></p> <ul style="list-style-type: none"> <li>- to create pedagogical resources</li> </ul>
<p>3. To support commitment:</p> <p>To develop the membership, to help members to clarify their own project and see how it can interact with the project of the CoP, etc.</p>	<p><b>Lean-Nett</b></p> <ul style="list-style-type: none"> <li>- Develop resources to better welcome new partners (the charter)</li> </ul> <p><b>ePrep</b></p> <ul style="list-style-type: none"> <li>to welcome new members</li> </ul>
<p>4. To support realization of the activities:</p> <p>To support organization, follow-up and management (the work of the coordinator(s), animator(s) or moderator(s)) To have a common environment for all the activities of the CoP</p>	<p><b>Learn-Nett</b></p> <ul style="list-style-type: none"> <li>- to propose a way for the coordination team to have a "context aware view" about what happens in collaborative groups in terms of activities of the actors and use of documents</li> <li>- to decide for a new workspace for all the activities</li> <li>- A tool that integrates forum and email messages for tutors.</li> </ul> <p><b>ePrep</b></p> <ul style="list-style-type: none"> <li>- to have a workspace for all the activities of the CoP</li> </ul>

Concerning the dimension of “engagement” in Daele’s model, we prefer the term “commitment”, which we define in terms of members clarifying their own project, seeing how it can interact with the project of the CoP, being actively involved in the activities of the CoP, and being personally committed to development of the membership. This is clearly linked with the welcoming of new members. It supposes some other conditions like the definition of the project of the community (and the regulation of it), the formalization of the project and the history of the CoP, and having a common knowledge of the participants (their competencies, interests, activities in the CoP, etc.). The need to support commitment is not often expressed by PALETTE CoPs (see need 3 in the table) but constitutes a potential need, which could be important. This potential need is observed in the practices of mature CoPs and could be interesting for new ones.

When we try to match Daele’s model with CoPs needs, it’s difficult to situate the dimension of exploitation of “produced resources” which is most expressed by CoPs. In the Daele’s model (see Figure 3) resources appear in the centre like used and exchanges objects (tools, rules, etc.). The formalization of the resources produced during the various forms of exchanges is contained in the conditions of learning (see above). So in our categorization of the needs we decided to dedicate one category for the constitution of the “common resources” (formalization, retrieval and reusing). These “common resources” belong to the CoP and can be appropriated by each member to support their own development. They represent the “wealth” of the CoP including its “memory”. It is similar to Wenger’s concept of “shared repertoire” (1998).

Finally, a fourth need appears within CoPs: the need to support the realization of the activities. It means the support of the organization, follow-up and management of activities (the work of the coordinator(s), animator(s) or moderator(s)). This need is common across all CoPs. It could concern each of the previous needs. We include here the need expressed by some CoPs to have a common environment for all the activities of the CoP.

We also make the connection between the CoPs’ categories of needs developed in the Table 2 and the model of CoPs’ actions proposed by Künzel, Charlier & Daele (2007). This allows us to explicitly relate CoPs’ specific identified needs with possible actions. This can orient the elaboration of actions expressed in the CoP scenarios. This model is anchored in observations of several CoPs in different domains of activity. It depicts five groups of actions in order to highlight questions relating to the CoPs’ development: towards which organization or project does a CoP intent to develop? How to support it throughout its development? The five types of actions are:

- Projects: they are actions oriented towards a specific and well-identified goal, possibly organised in the long term and requiring a high degree of coordination between the members.
- Social: they are actions oriented to promote community cohesion; they can be spontaneous or consist in specific and organized events.
- Sharing actions: they are short term actions, integrated in the day-to-day life of a CoP; they are not necessarily coordinated at a high level and can be more spontaneous than projects.
- Management: they are actions oriented towards the organization and the facilitation of the CoP as a whole such as distribution of roles, meetings organization, management of the work process, etc.
- Metacognition: these actions are related on CoP’s self understanding and self-direction; they can be spontaneous or coordinated; their purpose is to get feedback about the functioning of the CoP and to develop new actions taking into account the feedback.

So we make the connection between the CoPs’ categories of needs and their actions in order to illustrate what kind of actions are privileged by CoPs in order to fulfil their needs. This can orient the actions outlined in the CoP scenarios.

For example, CoPs organise social actions to support commitment, sharing actions (FAQ) to elicit the constitution of common resources, management actions to support the efficient realisation of CoPs activities and projects to constitute common resources.

In the Table 3 the categories of needs are matched with the categories of services that could be offered in PALETTE.

**Table 3 – Categories of needs and adapted services**

Categories of needs	Categories of technological services	Illustrations of learning services
1. To support participation	Collaboration and awareness <sup>1</sup> services	Pedagogical scenarios for technical and social training
2. To constitute common resources	KM and information services : Production, Restructuring, Metadata, Retrieval, Reusing, Awareness	Strategies to capitalize, retrieve and reuse information
3. To support commitment	Collaboration and awareness services	Methodological tool to support the definition and the regulation of CoP activities, illustrations of CoPs practices to define themselves, their identity
4. To support realization of the activities	Collaboration and awareness services	Methodological tool to support the choice of a technological environment and its adaptation to PALETTE services

Each category of technological service represented different services which can interact to meet CoPs needs. We attempt in the section 5 to analyse these interactions between services.

---

<sup>1</sup> In the context of the collaborative work at distance and the use of a virtual environment shared by a group **awareness** indicates the perception which each people possesses of the presence, the localization, the identity, the availability of another people, at a moment, during the connection. It's also the perception of what was realized between two successive connections, in the history of the activity of the group. So awareness tools are tools which support this awareness.

## 5 – Interaction between PALETTE Services

As stated in Appendix 3, the interaction between services is a commonly expressed need among PALETTE CoPs. Thus, the feasibility, requirements and implementation of the interaction of services need to be examined. This section starts by describing the expressed CoPs' needs requiring services interaction. Afterwards, the different types of services interaction are classified and then the problems related to each required interaction type are tackled.

### 5.1 Addressing CoPs Needs through Services Interaction

In order to fully answer the needs of CoPs, PALETTE services need to be able to communicate with each other or to be integrated. As a matter of fact, CoPs have explicitly expressed specific needs requiring interactions between PALETTE Services (see CoPs templates in the Appendix 3). In particular, problems linked to resources storage seem recurrent: for example, some CoPs would like to store their documents in one or more repositories and be able to access them transparently from one particular service; or to annotate documents stored e.g. in e-Logbook environment, using other services offered by e.g. Amaya or SweetWiki. Another example would consist of calling CoPe\_It! functions from e-Logbook in order to sustain argumentation for a community of practice using e-Logbook environments, at a time where the latter does not offer this feature. Alternatively, CoPe\_It! users could benefit from the e-Logbook context-Aware View, a rendering service not supported by CoPe\_It!, but however important in collaborative environments supporting mediation and argumentation, because it provides seamless embedded awareness information crucial for decision-making.

Moreover, more examples can be found in Table 4 which provides a summary of the expressed CoPs needs and specifies for each case, which technical services should interact to satisfy those needs.

**Table 4 – Examples of PALETTE services interaction**

Category of needs (see table 2)	CoPs	Technological services	Examples of Interactions of services
1. To support participation	Learn-Nett Adira	CoPe_it! services e-Logbook services	CoPe_it! services should call e-Logbook services.
2. To constitute common resources	Learn-Nett Did@ctic ePrep Adira	Document Production: services offered by Amaya, LimSee3, Sweetwiki tools (ePrep, Did@ctic, LN)  Restructuring service (to produce structured documents from existing ones) (Did@ctic)  Metadata production: e-Logbook and, Sweetwiki tagging services. Amaya, Linkwidget, Generis, BayFac annotation services (ePrep, Did@ctic, Adira, LN)  Information Retrieval: Generis, Corese (Linkwidget and Sweetwiki), e-Logbook search engines. (ePrep, Did@ctic, Adira, LN)  Reusing of structured documents: DocReuse matching service. (Did@ctic)  Awareness : e-Logbook services	Documents produced by Amaya should be consumed by DocReuse and restructuring services.  Documents tagged within Sweetwiki could be retrieved using Corese

		(ePrep, LN, Adira)	
3. To support commitment	Adira (LT) LN (LT) ePrep	e-Logbook	e-Logbook services could interact with external services (e.g., calendar)
4. To support realization of the activities (common environment)	ePrep Learn-Nett Adira	e-Logbook	e-Logbook services could interact with external services (e.g., calendar)

## 5.2 Classification of Services Interaction

From a general point of view, there are different levels of interaction between services:

1. Information Exchange: transmission of data and metadata between two or more services;
2. Integration: direct call to a service function from another service;
3. Composition: strict composition refers to a service, which is built from a composition of other services' functions.

Allowing information exchange between services requires the adoption of a common protocol, as well as a common understanding of the exchanged messages. This is also the case with integration, which requires the calling service to know how to actually call the function it wants to use, being able to input information to the function in the good form; and being able to retrieve and understand returned information so as process and integrate it. Composition requires all that is needed for information exchange and integration, plus orchestration; service orchestration dealing with the composition of services or their functions and the management of the information flow between these services.

The examples listed in the previous section show that until now the CoPs needs require services integration (call of a service's function from another service) and information exchange (data and meta-data access and sharing among services). To start with, we will tackle the issues related to the exchange of information between services supposed to satisfy the expressed CoPs needs to access data and metadata stored in different environments by interfacing one particular environment or some kind of cross service. Then, we will address the questions related to the need for integration.

## 5.3 Challenges with Information Exchange

PALETTE services are implemented based on different partners' tools. They have their own data and meta-data storages, and specific ways to handle these data and meta-data, using their own vocabulary and data structures. From a pure service composition point of view, we are in a context where independent peers need to exchange and share information, which is technically related to service choreography (Peltz, 2003).

The way data and meta-data will be shared and accessed by services is an important issue that raises several questions. In particular, is a common data repository needed? Do metadata and data need to be replicated in the different storage environments a CoP uses? Should all data and metadata be stored on the web to improve accessibility and sharing? For example, when a user wishes to tag a document initially stored in e-Logbook, using Amaya, where should the tags or other metadata be stored?. Moreover, if a community member using a particular service, wishes to see the tags and other metadata associated with a document should the annotations done through this particular service only be shown, or should the service automatically send a request to other tools in a transparent way, asking them for metadata related to the document in question? Or should this be done only based on a user's explicit request, to go and search in other tools? The same applies when a user wishes to retrieve documents stored in different places based on specific selection criteria. Should he/she use some kind of cross tool responsible for bringing together documents stored in different locations and relate metadata and awareness information gathered from different places, or should every service be



capable of directly addressing other services for such purposes? Last but not least, right and access management issues should also be addressed, as exchange information stored in a specific environment is usually governed by right management policies, which are application or service-dependent. To solve this issue, common policies concerning the access to data and metadata need to be defined.

Some elements of solutions can be given, but agreeing on the best policies will require further and deeper investigations. Concerning data storage and access to data (most often multimedia or textual documents), a possible policy would be to consider that such data is stored in only one place, and any service that needs it works by reference (e.g. for adding metadata), or on a local copy of the data, that is serialized back to the original storage when the service has ended its processing. This however requires handling access rights on data, so as to avoid concurrent modifications; this can possibly be done by a specific orchestration service. Concerning metadata, annotations inside a document are de facto available to any service that can read them. Then, any service that does not annotate inside a document can work by reference, but references between a document and its public metadata needs to be kept somewhere in e.g. a specific awareness technical service, so that any service can have access to any metadata made public by other services.

From these reflections, it appears that as soon as a CoP will use multiple services, additional orchestration and awareness technical services will be needed, whether working in a centralized way or in a peer-to-peer architecture.

The second identified issue concerns the needs in term of interactions between services and the complexity of these interactions.

#### **5.4 Challenges with Services Integration**

Looking at the examples with CoPe\_it! and e-Logbook, we can derive the following generic scenario: *“A CoP making intensive use of a given service, ‘A’, offering multiple functions, wants to benefit from additional functions offered by other services and use them in A”*.

The integrations considered so far address interoperability very locally, as problems of integration have been often discussed between two tools offered as PALETTE services. A first step to reach a better interoperability level might be to agree on a standard for calling services’ functions and a common syntax to specify input and output information. However, even if this standardizes the external access to services’ functions, the logic necessary to allow actual integration remains on the calling service’s side. In particular, the integration at the user interface level will require specific coding, and semantic alignments between the terms and data structure used by both parties. Reaching a high interoperability level between PALETTE services, to avoid as much as possible specific coding could possibly be solved by securing interoperability at the semantic level. In other words, an adequate solution might be to define a common meta-model or ontology defining the concepts used by all the services, create mappings between this model and the vocabulary used each service, and modifying each service so that it can handle this model. The benefits would be that any service would be able to use any function of any other service, without having to know this latter service and the specific vocabulary it uses. Nevertheless, at this stage, no complete feasibility study of such an approach has yet been undertaken.

#### **5.5 Future considerations**

The challenges mentioned above and the issues raised in sections 5.2, 5.3 and 5.4 will pave the way for a deeper examination of the problems related to services interaction, in order to find solutions, which on one hand, are feasible for PALETTE services in terms of implementation and on the other hand, are able to satisfy the previously stated CoPs needs. More results will be described in D.IMP.04.

## 6 – Conditions for Acceptance of the Services by Users: Addressing Acceptability Issues in PALETTE

‘Acceptability’ is a complex concept and includes several aspects. Nielsen’s schema (1993) (Figure 4) offers a way of understanding it. Namely, acceptability refers to at least two other concepts like utility and usability. These three concepts can be illustrated by some investigation of actions in the PALETTE project. “Acceptance” is linked to a good acceptability level of the system.

Globally, user acceptance of information technology remains an important concern in our information age: it is well known that users have neglected – or even rejected – a lot of software tools that the designers have been providing with the “*necessary utility for someone with their (own) physical and skill capabilities*” (Dillon, 2001). As argued by Norman designers and users do not construct the same conceptual model of the tools (Norman, 2002). Users themselves construct different models depending of their experience. It is even controversial whether designers can fully understand and elicit user’s needs as far as “*revealing such needs is contingent on a theory’s capability to model users contexts at design time and thereby predict possible outcomes of the design at use time*” (De Paula, 2004). Critics of the technological determinism position have brought the “social construction of technology” (SCOT) perspective: SCOT holds that “those who seek to understand the reasons for acceptance or rejection of a technology should look to the social world; they argue that the ways in which a technology is used cannot be understood without understanding how that technology is embedded in its social context” (Wikipedia). This is a critical issue in PALETTE because this project aims at designing software tools that are destined to be embedded in a social context that has strong social dimensions: the CoP. PALETTE tools will be embedded in a very demanding social context. This is the reason why the participatory design approach has been chosen as a core dimension of the project. The participatory approach addresses the problem of technology acceptance through a process of negotiation of tool **usefulness** (PALETTE D.IMP.01) which involves many actors (users, designers – and others...) and unfolds from the very early stage of the design process. It is expected to improve the acceptance of the designed services. Other inter-related dimensions described in the Nielsen’s schema determine technology acceptance. They need to be defined here as clearly as possible in the context of the project

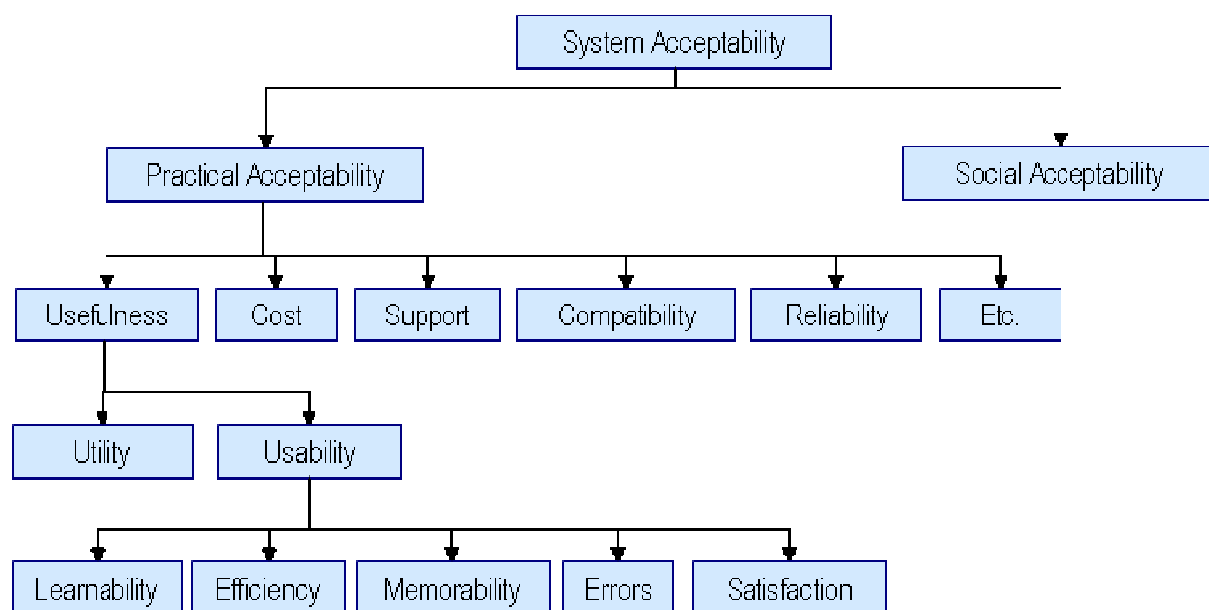


Figure 4 – Nielsen’s schema (1993)

### **Addressing Practical and Social Acceptability**

Participatory design is an overall approach that aims at designing acceptability of PALETTE technologies by the CoPs. Participation and negotiation are the key components of this process. However an efficient approach relies also on *pragmatic* strategies that support participation and negotiation in practice. One of the most important strategies is centred on elaborating scenarios of use. The “scenario” writing process is a democratic activity which i) provides opportunities for a large participation ii) focuses negotiation on concrete propositions iii) leads to functional specifications. The strategy of incremental design is another good example: as mentioned in 3.1 the setting-up of three sub-teams (A, B, C) who – each one - tackle two CoPs and two scenarios at least has brought substantial results. So, participative design looks like a chance for a good social acceptability but also contributes to increase utility. Even usability may be examined through validation of interface prototypes.

### **Addressing Usability**

The way participatory design is currently implemented in PALETTE is not sufficient. Usability cannot be reached only through the efforts of CoPs mediators. The problem is that usability is strongly related to user interface, and the user interface of the PALETTE services has not received enough consideration. Some developer’s proposals for the services user-interface are known, but its usability has not been tested yet in a real use context. Hopefully, this task is now planned (IP2). Referring to the Nielsen’s schema (1993), compatibility as a part of practical acceptability can be largely included in the usability analysis. Compatibility refers, on the one hand, to user’s and task characteristics and, on the other hand, to the organization of the dialog between the user and the service. It also concerns the degree of similarity to other well-known virtual environments. That is why compatibility has been, and will stay, an important concern in the first feedback sent to the developers. In addition, , and in the same way, general guidelines have already been proposed. They aim at standardizing work environments of the services in order to help users. A digest of a first version of these guidelines is available at the address: [https://bscw.ercim.org/bscw/bscw.cgi/d297096-5/\\*/\\*/\\*/\\*/\\*/\\*/Guidelines.htm](https://bscw.ercim.org/bscw/bscw.cgi/d297096-5/*/*/*/*/*/*/Guidelines.htm) (restricted access for Project members). They can be a base for a discussion between developers helping them to specify the interface of their tools.

### **Addressing Utility**

The issue of utility in PALETTE is addressed through Participatory design methodology leading namely to the writing of the scenarios for each CoP. The actual functional specifications are first results. Grounded of the analysis of CoPs needs and scenarios they identified specific services interaction as well as technological and users’ issues related with this services interaction. Thus the integrated services offered to each CoP will respect the utility criteria. Moreover interface analysis may provide information about utility. We plan to work on it in IP2.

### **Addressing Adaptability**

We may envisage two aspects in the adaptability of PALETTE Services: 1) the ability of users to customize or configure, to a given extent, some aspects of the service (user's customisation and configuration adaptability); 2) the ability of the different services to be adapted to different technical contexts with a minimum of development due to the respect of standards and architecture requirements (architectural adaptability).

The issue of adaptability has received recent attention in so far that the categorization of CoPs actions proposed by Kunzel, Charlier & Daele (2007) may point to an ecological approach to user-interface design focusing on cognitive affordances. This approach – which has been applied in complex domains like process-control or medicine - would potentially reduce the mental workload of users in so far it succeeds in making the CoPs’ working environment perceptually evident. The proposed model with five types of actions is a first step towards this.

Adaptability can also be considered through the interface analysis. It remains a delicate issue because an immoderate adaptability may badly influence compatibility.

## 7 – Conclusion

This deliverable paves the way for a next important phase in our participatory design of integrated technological and learning services to support CoPs learning and development. In conjunction with the D.PAR.03: “Description of six scenarios and of the results of six validated trials” to be produced at month 18 by WP1, we will be able to identify generic scenarios fulfilling similar needs of various CoPs with specific uses of integrated services and learning services. For example, the need to constitute common resources expressed by LEARN-NETT, ePrep and Adira will lead to a specific interaction between a document and metadata production services (Sweetwiki) and an information retrieval service (Corese). The technological challenges highlighted in this deliverable (sections 5.3 and 5.4) as well as the conditions for acceptance of the services by users (section 6) will be addressed by the teams reorganised according to this identification of generic scenarios. With this methodological approach we intend to find solutions which on one hand, are feasible for PALETTE services in terms of implementation and on the other hand are able to satisfy CoPs’ needs. More results will be described in D.IMP.04.

## 8 – References

- Daele A. (2006) A model for representing professional development through the participation in a virtual CoP: uses for developing enhanced services. Paper presented in *TEL-COPS 2006*. [https://bscw.ercim.org/bscw/bscw.cgi/d173003/tel-cops2006\\_amaurydaele\\_camera\\_ready.pdf](https://bscw.ercim.org/bscw/bscw.cgi/d173003/tel-cops2006_amaurydaele_camera_ready.pdf) (restricted access for Project members)
- De Montmollin, G. (1977). *L'influence sociale. Phénomènes, facteurs et théories*. Paris: Presses Universitaires de France.
- de Paula A., 2004, *The Construction of Usefulness: How users and Context Create Meaning with a Social Networking System*, PhD dissertation, University of Colorado.
- Dillon, A., (2001), *User Acceptance of Information Technology*. In W. Karwowsky (ed). *Encyclopedia of Human Factors and Ergonomics*, London, Taylor and Francis.
- Esnault, L., Zeiliger, R., Vermeulin, F. (2006) On the Use of Actor-Network Theory for developing Web Services Dedicated to communities of Practice, *1<sup>st</sup> International workshop on Building Technology Enhanced Learning Solutions for Communities of Practice* (in conjunction with 1<sup>st</sup> European Conference on Technology Enhanced Learning), Crete, Greece October 2, 2006
- Künzel M., Charlier B. & Daele A. (2007). Modelling activity and development of communities of practices. *Conférence EIAH 2007* (Environnements Informatiques pour l'Apprentissage Humain).
- Nielsen, J., 1993, *Usability Engineering*, Boston, MA, Academic Press.
- Norman, D., (2002), *The Design of Everyday Things*, MIT Press.
- Peltz, C., 2003, *Web Services Orchestration and Choreography*, IEEE Computer, Volume 36, Issue 10, Page(s):46 – 52.
- Wenger, E. (1998). *Communities of practice : learning, meaning and identity*. Cambridge University Press.

## 9 – Appendices

### 9.1 Appendix 1 – Template for the functional specifications

#### **Functional Specification of PALETTE services Template**

This document presents a template for Functional Specification of PALETTE services. The main purpose is to describe the offered services from a user's perspective (CoP member).

#### **1. Introduction**

##### **1.1 Purpose of the service**

This section provides a brief overview of a given PALETTE service and the motivation behind its development. It also describes functions related to the service and details how the service could work with other services (if necessary).

##### **1.2 Glossary and document conventions**

This section defines technical terms used in the document (only include those with which the reader may not be familiar).

#### **2. General description**

##### **2.1 Service' Functions**

Describe the general functions of the service.

##### **2.2 User Characteristics**

Describe the features of the user of the service (e.g., expected expertise with software and application domain).

##### **2.3 Example of use**

This section should describe an example of use of the service from the user's perspective in order to have a collective understanding of the main functions of the service.

##### **2.4 Functional design considerations**

Functional design considerations detail the attributes that affected the service's functional design.

Examples of attributes include:

- \* Assumptions that were made
- \* Prerequisites for the correct working of the service (e.g., needed operating environments...)
- \* Resource requirements in terms of hardware, other software or equipment
- \* Installation
- \* Security

#### **3. List of functions**

This section defines the complete list of functions offered by the service with their associated input/output arguments. This can be done as fully text section or using tables for each individual function. Each function' description includes:

- \* Purpose: the purpose of the function
- \* Input arguments: input format, who supplies the input
- \* Process: describes the main steps performed by the function
- \* Output arguments: desired output format, destination for the output
- \* Comments

## 9.2 Appendix 2 – Illustration of the functional specifications of a service: Amaya

### Functional specification for Amaya

Amaya is a web editor, i.e. a tool used to create and update documents directly on the web. Browsing features are seamlessly integrated with the editing and remote access features in a uniform environment. This follows the original vision of the web as a space for collaboration and not just a one-way publishing medium.

#### 1. Introduction

##### 1.1 Purpose of the service

The purpose of the service is to produce and publish web documents in an easy and possibly collaborative way.

Web documents are structured documents represented in HTML or better in its most recent version, XHTML. The presentation of web documents exploits the document structure, but it is clearly separated from it, thus allowing the aspect of a document to be changed according to the environment where it is used (screen size, disabilities of the user, etc.). Presentation is based on CSS, a simple mechanism for adding style (e.g. fonts, colors, spacing) to web documents.

Amaya includes the usual editing functionality of a word processor, but it also provides help to manage the document structure and its presentation. With Amaya, users can easily insert, delete, copy, paste, and transform the XHTML structure of edited documents. A document generated by Amaya is always correct (w.r.t. the XHTML language) and is then accessible with *any* web browser.

Amaya allows users to simultaneously browse and edit web documents containing not only text and images, but also mathematical expressions (coded in MathML) and animated graphics (coded in SVG). It provides a collaborative annotation mechanism that allows users to add information (comments, notes, remarks) about a document that cannot be edited.

Amaya can be used to create a broad variety of web documents such as institutional pages, technical reports, slides shows, curriculum vitae, address books, agendas, etc. All these kinds of documents have their specific conceptual components. For example, menus and lists of events are often used in institutional pages, chapters and sections in technical reports, slides in slides shows, etc. To address the specific requirements of each kind of document, Amaya uses templates. A template is a document skeleton that is used to automatically generate the most specific parts of each type of document. Users have then just to provide content and to create new pre-declared components derived from the template.

Finally, Amaya does more than web documents, even if it is its main point. It can be used as a word processor, as a slide show editor and its template engine allows it to create syntactically correct forms that you just have to fill. With no knowledge of XHTML (or MathML, SVG), one can produce simply potentially complex documents, virtually anything that can exist in the form of a classic paper form.

##### 1.2 Glossary and document conventions

**XHTML** - is the publishing language of the web. XHTML defines a set of structure elements such as divisions, headings, paragraphs, lists, list items, inline elements, tables, cells, etc.

**CSS** - Cascading Style Sheets is a simple mechanism for adding style (e.g. fonts, colors, spacing) to web documents.

**MathML** - is a language for describing mathematical expressions integrated in web documents.

**SVG** - is a language for describing two-dimensional, animated vector graphics integrated in web documents.

#### 2. General description

## 2.1 Service' Functions

Amaya can be considered as a word processor with some particular features. The most salient feature is its ability to produce documents that strictly conform to web standards and fully exploit the most advanced features of these standards. This allows documents to be used in many different environments, with a very wide variety of tools and devices.

Amaya provides a template mechanism. A template is a document with predefined contents and holes where the user can insert information. Some parts of a template may look like a form while some others are much less constrained. A template defines the skeleton of a document of a certain type. It declares components that are specific to that document type. For example, a slide show template defines a slide component that generates a division with a class attribute "slide" which includes a heading followed by a list of items. A template also indicates where this kind of components must or may appear in the document, how many occurrences are mandatory or possible, etc. Each template comes with at least one style sheet that specifies its presentation.

Once an expert user has described a template and its presentation, an author can quickly and easily create a document instance according to this template. The document skeleton is automatically generated and the user has just to add information. In accordance with the template, components are proposed to the user for insertion at the right position in the document instance. With that approach, Amaya becomes a customized document editor for the user. It allows many different types of documents to be produced consistently and easily, provided a template is created for each type.

With the publishing feature of Amaya, an author really gets the feeling that s/he is editing the web. Working on remote documents is exactly the same as working on local files. A single click on the Save icon immediately updates the document on its server. Creating a new document on a remote server just requires the author to enter the address of its location. Obviously, to write a file remotely, the user must be allowed to do so, and an authentication process is involved the first time a file is saved to a remote server during a session. This allows several remote users to efficiently share documents.

Another interesting feature of Amaya is its ability to send the edited document by email, or to just to send a notification to another user when a document is published or updated on a web server.

## 2.2 User Characteristics

Document authors with some XHTML expertise may freely manipulate the XHTML structure of their documents. A very convenient feature of Amaya is that it uses the same markup whatever document you intent to write. XHTML's learning curve, with the help of Amaya, is very short.

An average user, with no knowledge of XHTML at all, can also quickly and easily generate a document instance belonging to a template (technical report, slides show, curriculum vitae, address book, agenda) provided by expert.

The creation of a template and its style sheets requires more expertise. The author must understand XHTML and CSS. He/she has also to learn about the [XTiger language](#), in which templates are expressed.

## 2.3 Example of use

To create a new document instance from a template: use entry **New>From\_template** in the File menu. It displays a dialogue box where you can select:

The directory where the template is located if it is a local file that is not registered in the list of templates

The template itself, either by selecting in the list of registered templates or by typing the URI or file name of a template that is not in the list

The URI or file name of the document instance you want to create

Where to open the new document (Replace current, In new tab, In new window)

The Title of the new document

When a document instance is displayed (the following image presents an instance of a slide show document), Amaya shows the various parts of the template under the form of colored frames that enclose XHTML elements.



A `use` element is shown as a dashed blue box: the character string it contains can be freely replaced by the author

A `bag` element is shown as a dashed green box: it allows several elements of different types to be entered

A `repeat` element is shown as a dashed purple box: it allows a given structure to be repeated several times

An `option` element is shown as a dashed yellow box: it allows a given structure to be created or removed

The `attribute` element is not displayed in the main view, as it only impacts the Attributes menu.

Editing is allowed only inside the repeat, option, use, and bag elements, i.e. within the colored dashed boxes. The rest of the document is the fixed part of the template and can not be modified, such as the four-square logos or the blue hairlines in the screenshot.

When additional occurrences of a repeated structure are allowed by the template, clicking one of the purple '+' icons creates a new occurrence at that location. A pop up menu lists all the elements that can be inserted at that position. A click in that pop-up menu, creates a new occurrence of the corresponding element. In some cases the template offers only one type of element. The pop-up menu then contains a single entry, to clearly state what will be created.

Most `use` elements (blue) allow the author just to enter free text or to replace/edit existing text. Other `use` elements offer a choice between different types of elements that can be inserted at that position. In that case, a triangle icon is displayed in the top left corner of the box. By clicking this icon, the author can select one of the allowed types through a pop-up menu. When the type is chosen, the corresponding structure is generated and its content can be freely edited.

In a `bag` box (green), the author can insert any number of elements of the types specified by the template.

Option boxes (yellow) display a 'tick' icon in the top left corner. Clicking this icon creates or removes the optional structure.

## 2.4 Functional design considerations

Amaya must be installed on the user's machine. As an open source project, this software may be downloaded by anyone. It is available for a large variety of platforms (Windows, Linux and MacOS X).

Amaya uses the http PUT method to publish documents on a web server. See [how to configure the APACHE server](#) to support the PUT method.

## 3. List of functions

Amaya offers too many different functions to be listed in detail here. Users can check the help provided with the tool for more information (also [available online](#)). This section only gives a short overview of the main categories of functions.

### Templates

In a template, the skeleton document contains some statements, expressed in the XTiger (eXtensible Templates for Interactive Guided Editing of Resources) language, that specify how this minimal document can evolve and grow, while keeping in line with the intended type of the final document. Documents produced from this skeleton following the XTiger statements are called *instances* of the template. Some parts of the template may be frozen, if they have to appear in document instances as they are. Some parts may be changed when producing an instance document, some others may be added either freely or under some constraints.

### Editing

An important issue with structured editing is the user interface. Manipulating a possibly complex structure is not often something that can be done in an intuitive way. In particular, following too closely the intrinsics of XML structures does not lead to a user friendly interface.

Therefore a specific user interface is provided, that allows users to interact with the editor according to their own representation of the document. While some generic commands allow authors to manipulate

all types of elements the same way, additional commands are dedicated to the most complex structures. For instance, specific commands are available for editing rows, cells and columns in tables, that are different from commands for editing lists and their items.

Structure transformations constitute a powerful way to manipulate structured documents, especially when users frequently change the structure of their documents. Structure transformations are involved in the editing commands, like splitting and merging elements, or transformation commands where the user chooses the new type of elements.

Hypertext links are important elements on the web. Amaya provides specific commands to create and manage these links, and it allows the user to work on several documents at a time to facilitate this management.

### **Views**

While it is important to provide the author with convenient commands for creating or modifying the document structure, this is not enough. The user should also be able to comprehend the existing structure and to check the changes s/he is making to it. For that purpose, Amaya provides the concept of views. The idea is that several graphical representations of the document, with different points of view, help the user to perceive the various aspects of the document structure. An author may then choose the view what s/he feels the most appropriate for performing each editing task.

Sometimes, it is convenient to go directly to the source code and change things by hand. WYSIWYG is incredibly useful, but some WYSIWYG tools can make one become 'claustrophobic'. In Amaya, if something does not display as one thinks it should, one just looks what went wrong in the source of the document.

### **Styling**

Similarly, style may be entered with Amaya either through a graphical interface or by typing the CSS syntax directly. The user can also see the list of all style sheets involved in the presentation of a document and s/he can freely open and edit any of them.

If many style sheets are attached to a document, it is difficult to understand why an element is displayed with that color or these margins, etc. To help the user, Amaya provides a **Show applied style** command that displays in an separate window the CSS style properties applied to the selected element. In a single click, the user can then find the CSS source code that generates each property, and then change the rule that assigned a value to a property.

### **Publishing**

Amaya can directly save a document whatever its location thanks to the connection to the web. From the user's viewpoint, there is no difference between local and remote documents: all documents are saved with the same command. All resources are then seen in a homogeneous space that can be accessed in read and write mode transparently.

Additional complexity comes from the fact that some web resources depend on other resources, and when moving one of these resources, some others may have to be moved accordingly. A usual case is a document that includes images, style sheets, and scripts. When such a document is copied or moved from one location (local or remote) to another location, associated resources are copied and dependencies are updated.

Another method to publish a document is to send it by email to one or several colleagues. Like with other publishing methods, associated resources are handled accordingly.

### 9.3 Appendix 3 – Template for the scenario

#### Scenario for *\*\*\*Name of the community\*\*\**

Document version:

Date:

Contributors:

#### 1. Document's contents and target audience

*Here is a brief description of the contents of the document as well as the target audience in the community (all the community? or only subgroup(s)? or only a coordinator?...).*

1.1 Contents

1.2 Audience

#### 2. Community's needs and scenario's purposes

*In this section, the validated needs that the scenario deals with are presented, with brief "vignettes" describing activities of the community in which the needs are particularly obvious. It is expected that the problems lived by the community be told and explained regarding its context. The objectives of the scenario regarding the identified needs are also presented. Note that this section fully complies with the "purposes" of a scenario as described in PALETTE D.PAR.02. (see, p. 40).*

##### 2.1 Group of needs

##### 2.2 Purposes of the scenario

*Other groups of needs and scenario's purposes can be described. In the example above, one PALETTE tool is proposed to deal with the chosen needs. But obviously, several PALETTE services can be interconnected (it is even strongly advised!).*

#### 3. Methodology

*In this section, the "life cycle" of the scenario is described (see D.PAR.02., p. 40). Who participated in its elaboration, through which activities and when?*

#### 4. Scenarios' design and description

*Here, the scenario itself is described. The actors and the services offered by the PALETTE tools involved (see the Naudet's paper about the connections between "tools" and "services" at <https://bscw.ercim.org/bscw/bscw.cgi/d199002/Services%26Tools%20meta-model.pdf>) are firstly identified. Then a range of plausible scenarios are presented. Finally, summaries of the scenarios are presented in tables and the use cases integrating the different services used in the scenario are depicted.*

##### 4.1 Actors and services

*This section specifies:*

- *the actors of the community who are involved in the scenario*
- *the services and functions offered by the PALETTE tools that are invoked in the scenario (with the functional design considerations such as the need for login/passwords, the URLs, the location of the documents stored if any, etc.)*
- *the services/tools that the community already uses.*

##### 4.2 Description of plausible scenarios

*The scenarios are presented in a narrative form specifying the services offered by the PALETTE tools, the actors, the activities and their articulation. The description has to be as clear as possible indicating if need be a time line, locations, the relations (communication and collaboration) between the actors, the handling of the offered services, the connections between the services, etc. Several scenarios can be developed, one per group of needs. In addition, several variations or alternatives in one scenario can also be developed. It is also possible to propose pieces of scenarios without tools or services to be used! As the validated needs of the communities are related both to the uses of new tools and to the development of actions, it is possible to propose a scenario (or a part of it) that only deals with the development or the elaboration of actions or way to organize the work of the community.*

### 4.3 Scenarios' schematic representation

A summary of the scenarios and their alternatives are presented in tables. One table is required for each scenario or alternative. The goal is to quickly have in one table the time line of scenarios' sequence of events. Here are two examples of tables.

Scenario's sequence of events

	Step	Sequence of events	Expected Results and Evaluation Criteria	Tool Requirements	Notes
	0.0	opening a discussion			
	1.x...n.x	invite contributors			
	1...n	prepare resources			

Scenario's time line

	Event 1	Event 2	Event 3	Event 4	Event x
<b>Face-to-face</b>	Objectives of and actors involved in the event 1 of the scenario.				Objectives and actors of the event x that takes place face-to-face after 3 events at a distance.
<b>At a distance</b>		Objectives, actors and services of the event 2 that is organized at a distance.	Objectives, actors and services of the event 3 that is organized at a distance.	Objectives, actors and services of the event 4 that is organized at a distance.	

### 4.4 Comments on the use of services in the scenario

This section allows to add comments or additional information about the services or softwares described in the scenario. For example, how will the PALETTE's tools be connected together and be related to the community's existing tools? What does the user need to know about this? Another example could be the information that are not specifically chronological and that do not appear in the time lines above such as possible alternatives or choices that the users could make during the scenario while in progress.

### 5. Conditions of participation

This section is about the conditions required for implementing the scenario by the community: the specific technical skills required by the actors to use the PALETTE services, the competencies required to implement the scenario (for example communication or collaboration at a distance), the possible need for negotiation within the community for implementing the scenario, etc.

### 6. Validation procedures

This last section is about how the scenario will be evaluated / validated with the community. The evaluation procedure that will be negotiated and carried out with the community will be described here (questionnaires, questions of interviews, indicators of evaluation from the D.EVA.02).

This section will be to add after the validation to be held in February/March 2007.

## 9.4 Appendix 4 – Questions related to integration issues

### 9.4.1 Discussion for ePrep

<b>Category of Need 1</b>	To support participation (Wikiprepas)
<b>Category of technological services</b>	Collaborative space and awareness
<b>Existing</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- No human and technical resources</li> <li>- Open Office and LaTeX documents (could this kind of documents be manipulated with identified services)</li> </ul>
<b>Identified services</b>	e-Logbook and SweetWiki services
<b>e-Logbook services: produced</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Hosting in EPFL during the PALETTE project and possible server to host after the project</li> <li>- Resources repository (what to use as repository)</li> </ul>
<b>Sweetwiki services: produced</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Use of ontology? Folksonomy?</li> <li>- How to build such ontology / folksonomy from Wikiprepas?</li> <li>- Conversion of Open Office and LaTeX document?</li> <li>- Where to store produced tags?</li> <li>- Do these tags enable information retrieval?</li> <li>- When tagging the document?</li> <li>- How the ontology (if any) will evolve?</li> <li>- What to tag? Granularity of tagging? (different for each project)</li> </ul>
<b>Interacting services</b>	<p>e-Logbook (awareness and organization), Sweetwiki, LimSee3 and Amaya services</p> <ul style="list-style-type: none"> <li>- retrieval of the documents produced in Wikiprepas, LimSee3 and Amaya ? (links between the services?)</li> <li>- Is information retrieval possible based on awareness?</li> <li>- Possibility to tag documents using Amaya, Sweetwiki (Use of a common ontology ? Where to store the tagging?)</li> </ul>
<b>Strategies</b>	<ul style="list-style-type: none"> <li>- To define the ontology with WP3 members.</li> <li>- Members begin to tag documents in SweetWiki</li> <li>- To think about the main objective of the resources retrieval process</li> <li>- First need : to develop the global platform (e-Logbook)</li> <li>- Wikiprepas : OpenOffice, LaTeX, ontology</li> <li>- Meeting with WP3 partners</li> <li>- First scenario with interactions</li> <li>- To have a common model (technical work)</li> <li>- Agenda + mailing list</li> </ul>

## 9.4.2 Discussion for @pretic

<b>Category of needs</b>	Knowledge management: share recycled information as well as newly-produced information; Mediation: help geographically sparse « personnes ressources (abbrev. PR) » (that's the generic name of @pretic members) to solve common problems.
<b>Category of technological services</b>	KM, sorting, retrieving, awareness, sharing, mediation, collaboration services
<b>Existing</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	R: Mailing-list archives; Courses written in heterogeneous formats (.doc, .ppt, .html, .odt, etc.); Other didactic material; E: Aging computer labs, soon to be renewed; U: Isolated PR's, who tend to be afraid to use the mailing-list. They first try to get things done by themselves or with the help of their (same-school) colleagues
<b>Identified services</b>	Services offered by Corese, SweetWiki, Amaya, ECCO
<b>Corese Services</b>	Pre-process existing documents to semi-automatically create an ontology <sup>2</sup> adapted to our target audience;
<b>SweetWiki Services</b>	Produce new knowledge; Organize knowledge; Write articles on previously discussed topics (via their mailing-lists); Social network creation/reinforcement; Provide awareness services so that people in the CoP do not forget its existence.
<b>Amaya Services</b>	Transition from using .doc/.ppt/etc. documents to more flexible .html documents; Publish these documents for students and for helping <i>extra-muros</i> teachers; Give the means to PR's to collaborate (write courses together).
<b>Interacting services</b>	SweetWiki and Amaya, both producing XHTML code <sup>3</sup> , interact quite easily. A document written with one tool can be used in the other one with virtually no human intervention.
<b>Strategies (tools)</b>	Create an ontology based on existing documents <sup>4 5</sup> ; Improve the ontology as new documents are produced; Make CoP members aware of the assets allowed by these innovating work strategies; Benefit from the tools-provided awareness.
<b>Strategies (social)</b>	Plenary meetings; Individual face to face meetings (they have proved to be vastly more efficient than plenary meetings); Organize trainings (plenary or individual) so that our CoPs members are able to use the tools in a quick and efficient way; Create a social network so that PR's stop 'fearing' debating in a public virtual space. <sup>6</sup>

### <sup>2</sup> Ontologies vs folksonomies

An ontology is a taxonomy presented as a set of concepts and relationships between these concepts. It is generally produced by experts. Its domain can be as large as needed (from the whole world to a very delimited part of it). A folksonomy is a user created taxonomy, i.e. a classification that might be (but not necessarily) hierarchical which is realized by means of 'tags' by simple readers (i.e. non-experts).

So, our 'ontology', when it begins, is more a folksonomy that becomes more and more of ontology as things go along. Experts and computer algorithms (in this case, Corese) allow for this transformation.

The original plan was to data-mine @pretic's original documents thanks to Corese, allowing us to have a base ontology (which would have grown as users were tagging documents) but time and content issues (the mailing-list archive contents was too informal (too semantically poor) for Corese to produce useful results) prevented us to do so.

### <sup>3</sup> Amaya/SweetWiki integration

At present time, the integration between Amaya and SweetWiki is a matter of copy & paste. 'Easier' ways could be found in the future, but copying and pasting, be it from the source code view or the WYSIWYG view (both software have both modes) is already really easy. Except for mass-exporting documents, there's no practical need to enhance the integration.

### <sup>4</sup> Ontology creation, who is in charge?

ECCO, the ontology editor soon to be implemented in SweetWiki (working with Corese, also a part of SweetWiki), will allow virtually anybody to classify concepts. In a more practical way, CoPs core members, with the probable help of their mediators, will be in charge of the ontology. Corese will also be able to do some of the classification automatically thanks to its AI, reducing the human work load.

### <sup>5</sup> Technical side of ontology management

These ontologies are stored server-side in XML files, using W3C's Resource Description Framework (RDF).

*'The Resource Description Framework (RDF) integrates a variety of applications from library catalogs and world-wide directories to syndication and aggregation of news, software, and content to personal collections of music, photos, and events using XML as an interchange syntax. The RDF specifications provide a lightweight ontology system to support the exchange of knowledge on the Web.'* (quoted from <http://www.w3.org/RDF/>)

### <sup>6</sup> Social aspects and awareness concerns

It has to be emphasized that in @pretic, major concerns are not computer concerns, but social concerns:

CoP members ask for a recognition of their roles as PR;

CoP members don't know each other very well (or in some case at all) and need to be more 'virtually' close than they are now to 'dare' sharing knowledge;

### 9.4.3 Discussion for Form@HETICE

<b>Category of Needs</b>	Production and updating of training and self-learning resources. Capitalization of existing ICTE practices and their dissemination inside the Form@HETICE network
<b>Category of technological services</b>	Sharing, debate, collaboration, KM, multimedia authoring.
<b>Existing</b> <ul style="list-style-type: none"> <li>• <b>Resources</b></li> <li>• <b>Environment</b></li> <li>• <b>Uses</b></li> </ul>	<ul style="list-style-type: none"> <li>- A mailing list</li> <li>- A wiki</li> <li>- A website</li> <li>- Plenary face-to-face meetings</li> <li>- Electronic mails</li> </ul>
<b>Identified services</b>	BayFac, Amaya, LimSee3 and Sweetwiki offered services
<b>BayFac services</b>	<ul style="list-style-type: none"> <li>- Possible researching</li> <li>- Structuration of the website documents</li> </ul>
<b>Sweetwiki services</b>	<ul style="list-style-type: none"> <li>- Produce new knowledge;</li> <li>- Write articles on previously discussed topics (via their mailing-lists);</li> <li>- Organize knowledge;</li> <li>- Social network creation/reinforcement;</li> <li>- Provide awareness services so that people in the CoP do not forget its existence.</li> </ul>
<b>Amaya Services</b>	<ul style="list-style-type: none"> <li>- Transition from using .doc/.ppt/etc. documents to more flexible .html documents;</li> <li>- Publish these documents for students;</li> <li>- Publish these documents for helping <i>extra-muros</i> teachers;</li> <li>- Make sense to “resource person” to collaborate (write courses together).</li> </ul>
<b>LimSee3 services</b>	Create multimedia presentation to support courses based on existing and future templates
<b>Interacting services</b>	SweetWiki and Amaya, both producing XHTML code, interact quite easily. A document written with one tool can be used in the other one with virtually no human intervention.
<b>Strategies</b>	<ul style="list-style-type: none"> <li>- Participation to Form@HETICE plenary meetings in order to follow the CoP’s needs evolution.</li> <li>- Presentation of the different services/tools during a plenary meeting.</li> <li>- Workshop on Amaya with experimented “resource people” during the Form@HETICE summer school (August 2006).</li> <li>- Workshop on Amaya and LimSee at the Form@HETICE conference (December 2006).</li> <li>- Presentation of the BayFac tool to the network animators in charge of its sustainability (next June 2007).</li> <li>- Organization of trainings (plenary or individual) so that our CoP members are able to use the tools in a quick and efficient way.</li> </ul>

Awareness tools, in @pretic's tools and services, are the one of the keys (if not the only key) needed to make @pretic work as a CoP.

## 9.4.4 Discussion for Did@ctic

<b>Category of CoP need</b>	To constitute common resources (To archive common resources and to make them retrievable and reusable )
<b>Category of technological service</b>	Capitalisation of CoP' resources
<b>Identified services</b>	Template-based restructuring services and LinkWidget services
<b>Existing</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- What type of resources (which format, structure, content)?</li> <li>- Any category of document has been identified?</li> <li>- Existent documents are in Word, ppt and HTML (we will focus on HTML documents for the moment)</li> <li>- Some class of documents could be identified</li> </ul>
<b>Existing environment and uses</b>	Word, PowerPoint, moodle repository
<b>Template-based restructuring services</b>	
<b>Produced resources</b>	<ul style="list-style-type: none"> <li>- Templates describing identified classes of documents are produced (manually or using an extension of Amaya).</li> <li>- CoP's documents will be restructured according to these templates (possible automation for the instances belonging to the same class of documents).</li> </ul>
<b>Produced environment and uses</b>	<ul style="list-style-type: none"> <li>- CoP members use predefined templates to produce "structured" documents (e.g., Amaya editor)</li> <li>- How to make templates evolve? What impact on the instances?</li> <li>- Where to save produced resources? Moodle repository? New repository?</li> <li>- Web-based application (installed on CoP server or distant server)</li> </ul>
<b>LinkWidget (tagging service)</b>	
<b>Produced resources</b>	<ul style="list-style-type: none"> <li>- Tagging structured documents (possibility to tag templates as well as instances).</li> <li>- How to tag documents (whole documents or parts of documents)? What does mean parts of document?</li> <li>- When tagging document is possible (when producing the document)?</li> <li>- Any ontology/thesauri could be used/created?</li> <li>- Such ontology could evolve?</li> </ul>
<b>Produced environment and uses</b>	<ul style="list-style-type: none"> <li>- Used editors should allow tagging when producing templates/ instances</li> <li>- Restructuring service should allow tagging when producing structured instances</li> <li>- Where to save produced resources (tags)? Moodle repository? New repository?</li> </ul>
<b>Interacting services</b>	<ul style="list-style-type: none"> <li>- Restructuring service should invoke LinkWidget</li> <li>- Restructuring service and LinkWidget should manipulate a common repository</li> </ul>
<b>Established working plan (strategy)</b>	<p><b>Phase 1 – short term:</b></p> <p><u>Categorization of documents (producing class of documents)</u></p> <ul style="list-style-type: none"> <li>- <b>Participants:</b> Didactic mediator/members, restructuring service team (EPFL)</li> <li>- <b>Expected result:</b> A set of document classes, examples of templates/instances.</li> </ul> <p><u>Validation of a first proposed ontology</u></p> <ul style="list-style-type: none"> <li>- <b>Participants:</b> Didactic mediator/members, LinkWidget (Inria-Sophia)</li> <li>- <b>Expected result:</b> a validated ontology</li> </ul> <p><b>Phase 2 – long term:</b></p> <ul style="list-style-type: none"> <li>- First restructuring tests (Didactic + EPFL)</li> <li>- Evolving produced ontology based on the previous analysis of CoP documents (Didactic + Inria-Sophia)</li> </ul>



## 9.4.5 Discussion for Learn-Nett

<b>Category of Need 1</b>	To support participation – debate, confrontation (decision making : to identify the problem, to keep traces)
<b>Category of technological services</b>	Sharing, debate, awareness
<b>Existing</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Discussion about cases of tutoring groups at a distance</li> <li>- Discussion during phone meeting, with preparation before the meeting (document for each participant : excel, attached in messages in the Moodle forum, then collected in one document by the coordinator)</li> <li>- Min Human resources</li> </ul>
<b>Identified services</b>	CoPe_it! and e-Logbook services
<b>CoPe_it!! Services – produced</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Provides a structure to the discussion</li> <li>- Allows documents or URL sharing</li> <li>- Tagging and annotation of discussions</li> <li>- Retrieve information from past discussions</li> <li>- It is possible to perform information retrieval using CoPe_it!?</li> <li>- Do CoPe_it! helps for document assembly? (documents related to a discussion)</li> <li>- Where to store structured discussions? (Using already existing moodle repository?)</li> <li>- Semantic tag? Manual, automatic? Before, during or after the discussion?</li> <li>- Use of a predefined ontology (link with KM services)?</li> <li>- Possible interaction with external resources (google calendar)?</li> </ul>
<b>e-Logbook Service produced</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> </ul> <b>Uses</b>	<ul style="list-style-type: none"> <li>- Provide awareness in the CoPe_it! Environment</li> <li>- Just for the navigation – context aware view</li> <li>- Need of another repository?</li> <li>- Dynamic changes of the context aware view (synchronization between Cope_it! and e-Logbook)</li> </ul>
<b>Interacting services</b>	CoPe_it!+ e-Logbook (synchronization issues) CoPe_it! + KM services (for tagging) <ul style="list-style-type: none"> <li>- Who will define the ontology and how it will be incorporated to CoPe_it!?</li> </ul> The same repository used for both services?
<b>Strategies</b>	<ul style="list-style-type: none"> <li>- To identify types of documents – (excel + themes)</li> <li>- To identify the need of tag</li> <li>- To create ontology</li> <li>- What about the integration with the platforms to be used by Learn-Nett in 2007-2008?</li> </ul>
<b>Category of Need 2</b>	To constitute common resources - Capitalization of resources (of ten years of working)
<b>Category of technological services</b>	Capitalization of resources (awareness), and information retrieval
<b>Existing</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Documents and text-based discussions disseminated in different platforms on different servers of the Learn-Nett partners</li> <li>- At the moment, a student is gathering all the Learn-Nett archives for classification.</li> </ul>
<b>Identified services</b>	SweetWiki, LinkWidget, ECCO
<b>Service SweetWiki – produced</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Tagging – Annotations</li> <li>- Management of folksonomies</li> <li>- Queries and Search</li> </ul>

<b>Service LinkWidget produced</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Simple annotation of static resources and conversations</li> <li>- Association of resources and conversations</li> <li>- Retrieving information in resources and conversations</li> </ul>
<b>Service ECCO</b>	<ul style="list-style-type: none"> <li>- Creation and management of ontologies</li> </ul>
<b>Interacting services</b>	SweetWiki-ECCO + LinkWidget-ECCO + SweetWiki-LinkWidget + LinkWidget-Learn-Nett platforms
<b>Strategies</b>	<ul style="list-style-type: none"> <li>- Create a table of contents for the documents stored on SweetWiki</li> <li>- Create categories of documents</li> <li>- Create an ontology describing the contents of the documents (reports of students groups, research documents and practices/actions of tutors)</li> <li>- Create pages in SweetWiki (from old documents and past text-based conversations)</li> <li>- Link documents with conversations and tag/annotate them</li> <li>- Retrieve information from the Learn-Nett platform and/or SweetWiki with LinkWidget</li> </ul>
<b>Category of need 3</b>	To have a common environment for all the activities of Learn-Nett (coordination, training of the tutors, training of the students, etc.)
<b>Category of technological services</b>	No PALETTE service but need to think about the interactions between this new environment and the PALETTE services (see category of need 1 and 2)
<b>Existing</b> <ul style="list-style-type: none"> <li>• Resources</li> <li>• Environment</li> <li>• Uses</li> </ul>	<ul style="list-style-type: none"> <li>- Two platforms in use: Moodle for the tutors and the coordination and GALANET for the training of the students</li> <li>- One external service: CENTRA for the synchronous discussion between the tutors, the coordination team, the tutors training, the internal meetings of the students groups, etc.</li> </ul>
<b>Identified services</b>	
<b>Strategies</b>	<ul style="list-style-type: none"> <li>- To have an environment which could be used for different purposes: <ul style="list-style-type: none"> <li>o Public web site</li> <li>o Easy to update</li> <li>o Hosted by one Learn-Nett partner in the long term and requiring a minimum of maintenance</li> <li>o Managing different levels of access</li> <li>o Providing Learn-Nett with communication, collaboration, repository tools and possibly interoperable with PALETTE services</li> </ul> </li> <li>- This need will be taken into account in the future. Learn-Nett has no sufficient resources for the moment.</li> </ul>