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.KNO.07 – KM evolution and evaluation services

Due date of deliverable: July 31st, 2008
Actual submission date: September 12th, 2008

Start date of project: 1 February 2006 Duration: 36 months

Organisation name of lead contractor for this deliverable: INRIA

| Project co-funded by the European Commission within the Sixth Framework Programme | | |
|---|---|---|
| **Dissemination Level** | | |
| **P** | Public | **PU** |

Keyword List: Ontologies, Knowledge Management
Responsible Partner: Rose Dieng-Kuntz, INRIA

| ***MODIFICATION CONTROL*** | | | |
|---|---|---|---|
| Version | Date | Status | Modifications made by |
| 1.0 | 30/06/2008 | Draft | CRP HT: chap. on Evaluation services |
| 1.1 | 24/07/2008 | Draft | Amira Tifous: chap. on Evolution service (Sections 2.2 to 2.7, and Conclusion) |
| 1.2 | 25/07/2008 | Draft | Olivier Corby, Amira Tifous: corrections on Evolution service chap. |
| 1.3 | 25/07/2008 | Draft | Alain Giboin: Summary, Gen. Introduction, Gen. Conclusion, chap Evolution Services (Introduction and Section 2.1). |
| 1.4 | 27/07/2008 | Draft | Amira Tifous: minor corrections and homogenisation of the document |
| 2.0 | 28/07/2008 | Version sent to evaluators | Alain Giboin |
| | 31/07/2008 | Feedback of Evaluator 1 | Etienne Vandeput (ULg) |
| 3.0 | 19/08/2008 | Corrections taking into account Evaluator 1's feedback | All |
| | 20/08/2008 | Feedback of Evaluator 2 | Aïda Boukottaya (EPFL) |
| 4.0 | 28/08/2008 | Corrections taking into account Evaluator 2's feedback | All |
| 5.0 | 02/09/2008 | Pre-final sent to the Steering Committee | Alain Giboin |
| 6.0 | 12/09/08 | Final | Alain Giboin |

**Deliverable manager**

- Rose Dieng-Kuntz †, INRIA
- Alain Giboin, INRIA
- Amira Tifous, INRIA

**List of Contributors**

- Pierre-Jean Barlatier (CRP HT)
- Olivier Corby (INRIA)
- Alain Giboin (INRIA)
- Yannick Naudet (CRP HT)
- Amira Tifous (INRIA)
- Alain Vagner (CRP HT)
- Géraldine Vidou (CRP HT)
- Marie-Laure Watrinet (CRP HT)

**List of Evaluators**

- Aida Boukottaya, EPFL
- Etienne Vandeput, ULg

**Summary**

This deliverable describes the research work done in order to design the two last KM services envisioned in the Palette project: (1) a service for managing the evolution of CoP knowledge and (2) a service for evaluating this knowledge. The work for (1) consisted in

(a) identifying the CoP evolution events originating the evolution of the CoP knowledge, and the types of knowledge evolution resulting from these events, and

(b) exploring two approaches to deal with the coherent evolution of semantic annotations in case of evolution of a CoP ontology, and implementing the approaches in a research prototype.

The work for (2) consisted in:

(a) designing a model and indicators of knowledge value from the perspective of a CoP member,

(b) elaborating an algorithm based on the model for computing the value of knowledge, and

(c) describing the architecture of the future evaluation service.

The deliverable also outlines the research work which remains to be done during the last months of the Palette project.

# Table of Content

# Table of Figures

# Foreword

*ROSE DIENG-KUNTZ, the leader of the Palette WorkPackage # 3, passed away on Monday June 30, 2008, after a long illness. In charge of managing this deliverable (D.KNO.07), Rose had prepared the plan, and should ensure the writing of Chapter 2 on the Knowledge Evolution Service, of the general introduction and of the general conclusion. She also should ensure the entire edition of the deliverable. Unfortunately she has not been able to do this.*

*In order that D.KNO.07 as imagined by Rose comes into being, we decided to write the deliverable parts that she had intended to write. Concerning Chapter 2 in particular, let us specify that the sections 2.2 to 2.6 have been written on the basis of the (Luong and Dieng 2007) paper, a paper written by Rose and her PhD student Phuc-Hiep Luong. Section 2.1 was written starting from the following passage of the Implementation Plan # 3 (IP3) as a writing instruction: "We will study how to make evolve the [CoP] memory after some important events in the life of the CoP: introduction of a new member, removal of a new member, constitution of sub-communities, introduction of a new document, introduction of new arguments in a discussion, memory maintenance phase, etc."*

*Together with all the partners of WP3, we would like to express our gratitude to our colleague and friend Rose for the important research and management work she has done for WP3.*

*Alain Giboin and Amira Tifous*

# Chapter 1: Introduction

The previous deliverable (D.KNO.06) reported the new developments of *SweetWiki* and *Bayfac* services (a service for collaborative knowledge creation, and a service for faceted classification), and introduced two new complex KM services: *SemanticFAQ* (a service for semi-automatic and semantic annotation of a corpus of e-mails, and for semantic retrieval from such e-mails), and *KM LinkWidget* (a service to semantically link resources stored in a repository with conversations posted in a discussion forum).

The current deliverable reports the research work done around the two last KM services envisioned for the Palette project: (1) *the "KM Evolution Service":* a service for managing the evolution of both the ontology and the annotations supporting the indexing and retrieval of the resources stored in a CoP "memory"; and (2) *the "KM Evaluation Service":* a service for assessing the value of the knowledge exchanged within a CoP, and stored in the CoP memory.

This deliverable details the work effectively done at this time for each service, precisely:

1. *Research work done for the KM Evolution Service*: where we study the CoP evolution events leading the evolution of the CoP knowledge (CoP memory) and the types of knowledge evolution resulting from these events, and where we explore two approaches enabling to manage the evolution of a CoP memory by taking into account the interconnection between its elements (cf. annotations may evolve if the ontology evolves, etc.) (Chapter 2).
2. *Research work done for the KM Evaluation Service:* the design of a model and of indicators of knowledge value from the perspective of a CoP member, the elaboration of an algorithm based on the model for computing the value of knowledge and the description of the architecture of the future service (Chapter 3).

In conclusion (Chapter 4), the deliverable recaps the research work done about the two services, and presents the research work which remains to be done during the last months of the Palette project.

# Chapter 2: Knowledge Evolution Service

*Foreword:* Let us inform the reader that the work reported in this chapter is a *research* work, not a development one— and moreover that this work deals more with the technical aspects of the service than with its usage aspects. This means that, when planning this work in the framework of the IP3, it has been decided that the Knowledge Evolution Service would not be put in the hands of the PALETTE CoPs' users, and that no user testing would be performed. User testing would have been premature at this time of the design of the service.

A *Knowledge Evolution Service* is a service which allows managing the evolution of the knowledge of a CoP, and more exactly the evolution of the "materialisation" of this knowledge in a *CoP Memory*.

A *CoP Memory* partially corresponds to the *shared repertoire* concept described in (Wenger, 1998) model of a CoP, *i.e.*, the "shared repertoire of communal resources (routines, sensibilities, artifacts, vocabulary, styles, etc.) that members have developed over time".

We define a *CoP Memory* (or *Community Memory*) as being the "persistent, explicit, disembodied representation of the knowledge and information of a community, in order to facilitate their access, sharing and reuse by the relevant members of the community, within the context of their tasks". This definition is in fact an adaptation  we made of the definition of the notion of a Corporate Memory (or Organisational memory) given by (Dieng et al., 2005)[1].

Among the knowledge materialised in the community memory are the ontologies and semantic annotations, i.e. "the knowledge characterised by complex formal representations that can be processed with complex inferences (e.g. deduction, induction, abduction), and not only with retrieval mechanisms" (Kuhn and Abecker, 1997). The Knowledge Evolution Service envisioned in Palette is a service aimed at managing the evolution of these two kinds of knowledge. The deliverable focuses on the following frequent scenario: the user in charge of the ontology management makes some changes in an existing ontology that thus evolves to a new version of the ontology; as a consequence, some semantic annotations based on the old version of the ontology become inconsistent; because this inconsistency may lead other users to meet problems when searching some information in the documents indexed with such inconsistent annotations, it is necessary to fix the inconsistencies. The Knowledge Evolution Service is intended to help users detect the inconsistent annotations and repair them. When implemented, the service would help CoP's members as users to make evolve the annotations contained in the CoP memory.

This chapter describes the research work done in order to design the Knowledge Evolution Service. We will successively report the research work done about the management of ontology evolution (Section 2.2) and the research work done about the management of annotation evolution (Section 2.3). A community memory evolving according to the evolution of the community itself, therefore, in Section 2.1, we will use CoP evolution models as analysis frameworks in order to touch on (a) the events that can originate the evolution of the

[1] We acknowledge that an organisation and a community have not necessarily the same characteristics. This doesn't however rule out that the memory of such collective entities as CoPs and structured organisations both encompass a material side. The very generic definition we give here has no other goal than emphasizing that we approach the CoP memory from the material side, and that we focus on the materialisation of knowledge. This definition doesn't foresee the way this materialisation is or will be instantiated within a CoP; we consequently admit that the instantiation of the knowledge materialisation within a CoP may be different from the instantiation of the materialisation within a structured organisation.

community knowledge, and (b) the types of knowledge evolution that can result from (a). This can help to understand knowledge evolution better and to support it better.

## 2.1. Evolution of a CoP and of its Knowledge

The question we asked when initiating this research work was: “How to make evolve the CoP memory after some important events in the life of the CoP?” To answer this question whatever is the best, it requires getting a precise enough idea of these events (or causes of the evolution). Using CoP evolution models as frameworks (Section 2.1.1), we can identify some of the events originating the evolution of the CoP knowledge and some of the resulting types of knowledge evolution (Section 2.1.2).

### 2.1.1. Evolution of a CoP

Several models of CoP evolution have been proposed in the literature. They can be divided into two categories: *Lifecycle* models and *Maturity-Capability* models (Gongla and Rizzuto, 2001).

#### 2.1.1.1. Lifecycle Models of CoP Evolution

(Wenger, 1998), (McDermott, 2000) and (Wenger et al., 2002) models of CoP evolution are *lifecycle* models: they “describe communities as developing through stages akin to birth, maturation, and death” (Gongla and Rizzuto, 2001).

- **(Wenger, 1998) initial model of CoP evolution.** In this model, communities of practice are seen as developing through five stages: potential, coalescing, active, dispersed, and memorable, with levels of interaction and types of activities varying across the stages (see Figure 1). Members’ interaction within the community generally increases through the active level and then declines through the dispersed stage, and disappears at the memorable level, although memories, stories, and artifacts of the community still remain.

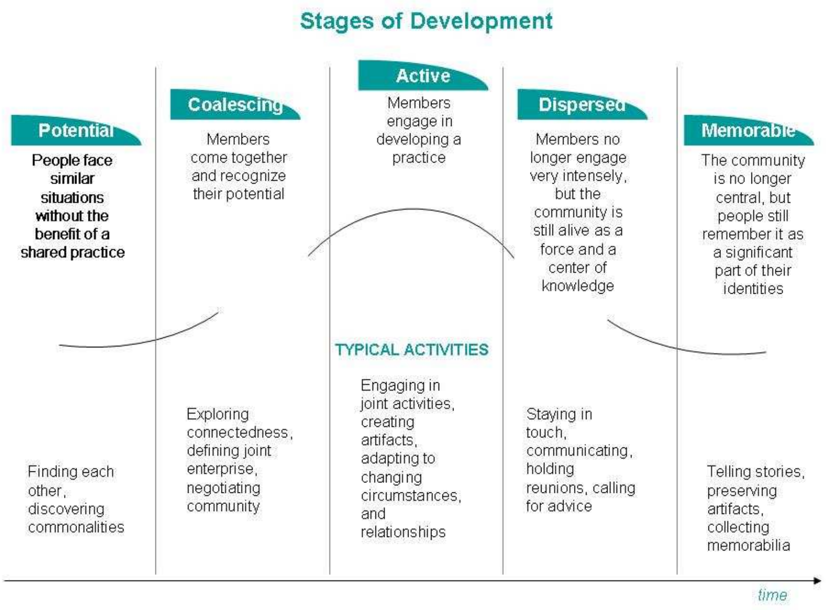

Stages of Development
Potential
People face similar situations without the benefit of a shared practice
Coalescing
Members come together and recognize their potential
Active
Members engage in developing a practice
Dispersed
Members no longer engage very intensely, but the community is still alive as a force and a center of knowledge
Memorable
The community is no longer central, but people still remember it as a significant part of their identities
TYPICAL ACTIVITIES
Finding each other, discovering commonalities
Exploring connectedness, defining joint enterprise, negotiating community
Engaging in joint activities, creating artifacts, adapting to changing circumstances, and relationships
Staying in touch, communicating, holding reunions, calling for advice
Telling stories, preserving artifacts, collecting memorabilia
time

*Figure 1 Stages of development of a Community of Practice (Figure of Wenger, 1998, redesigned by us)*

- **(McDermott, 2001) model of CoP evolution.** In this model, communities are viewed as living, human institutions that "form spontaneously, grow, mature, change, age and die;" hence the five-step lifecycle of the model: plan, start-up, grow, sustain/renew, and close (see Figure 2). This model is similar to the one of (Wenger, 1998), but details the tensions and challenges that stimulate the community to develop and renew itself, and eventually die.

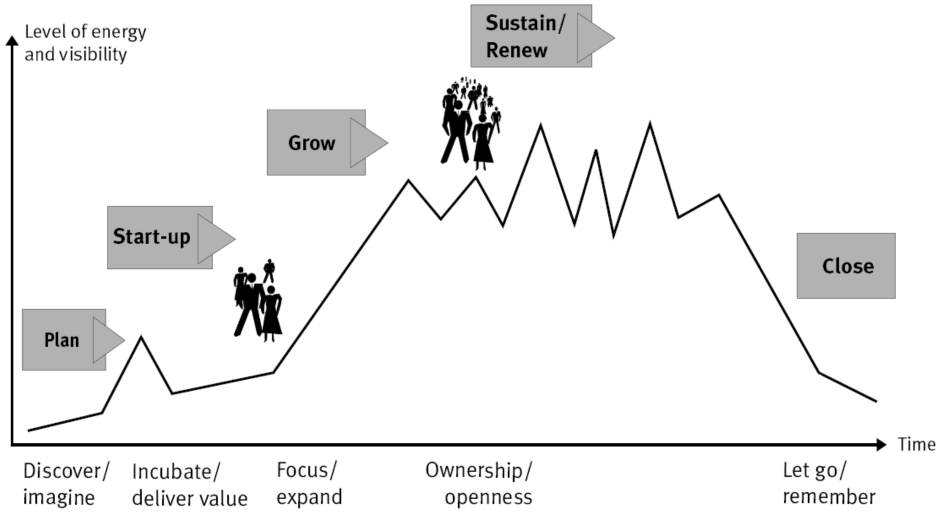


*Figure 2 The stages of community development (McDermott, 2000)*

- **(Wenger et al., 2002) model of CoP evolution.** In this model, Wenger, McDermott and Snyder combine the two previous models. Indeed, this model describes three life phases, which include five stages of community development representing the life cycle of a community: (1) *Formation (i/ potential and ii/ coalescing):* initial networks are discovered, common ground is formed and relationships are formed. The initial call (informally) is usually centred on the generation of value. (2) *Integration (iii/ maturing and iv/ stewardship):* focus upon particular topics and the admission of new members. Tools and methods are developed that are unique to the community. New ideas are continually welcomed as the community evolves. (3) *Transformation (v/ transformation):* the community may fade away or officially close, originate a new community, merge with other communities or become institutionalised as a formal unit. Figure 3 represents the average cycle of the life of a CoP community in terms of time and level of energy and visibility[2].

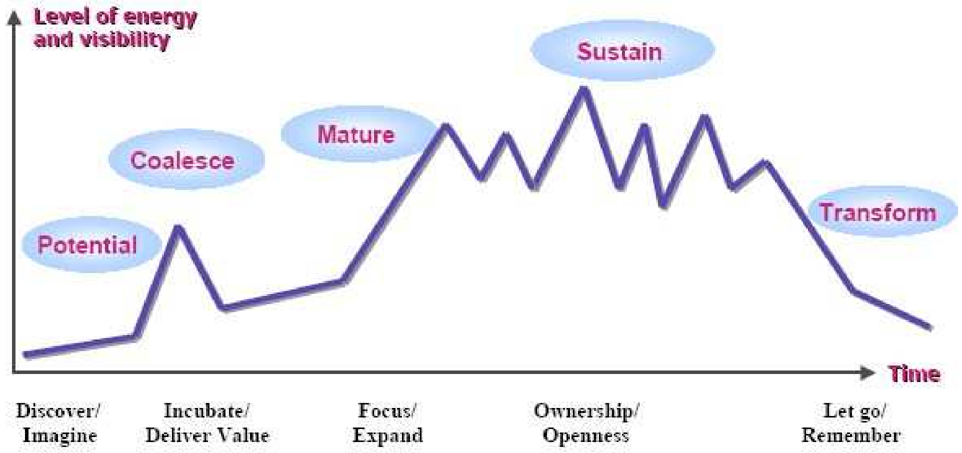
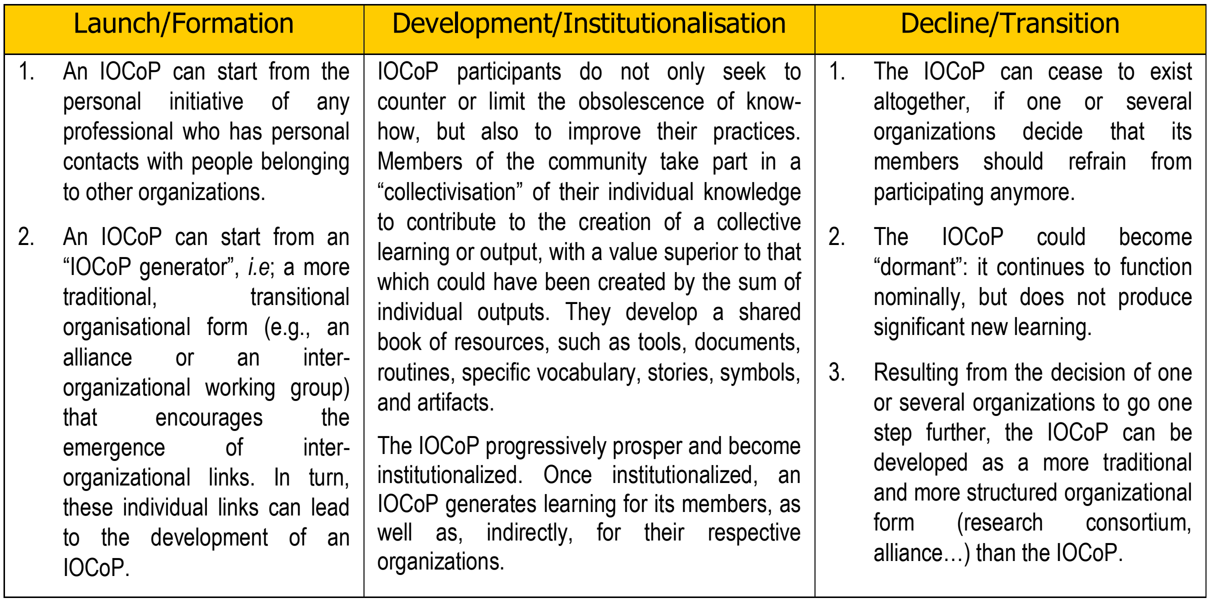


*Figure 3 Community life cycles related to time and level of energy and visibility (Wenger, McDermott & Snyder, 2002)*

- **(Moingeon et al., 2006) model of IOCoP evolution.** This model can be considered as complementing the (Wenger et al., 2002) model, in that it specifies the evolution of what the authors call an "Inter-Organisational Community of Practice" (IOCoP), *i.e.* "an organisational form having autonomous governance, gathering voluntary individuals from different organisations, with a common professional practice and aiming at developing their expertise on an individual basis". An IOCoP is viewed as evolving along three stages: launch/formation, development/institutionalisation and decline/transition (see Figure 4). .

[2] Note that the five stages of development are generally not substantiated with specific real-life examples. In this regard, "the purpose of presenting these stages is perceived to be academic" (Wenger, McDermott and Snyder 2002).

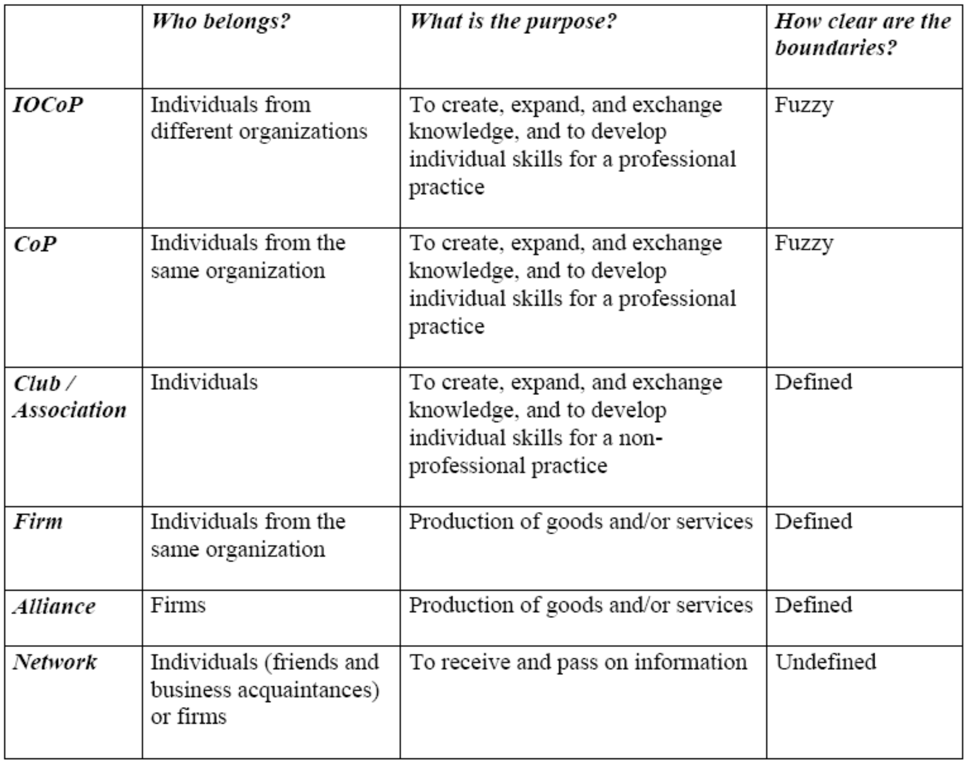


| Launch/Formation | Development/Institutionalisation | Decline/Transition |
|---|---|---|
| 1. An IOCoP can start from the personal initiative of any professional who has personal contacts with people belonging to other organizations.<br>2. An IOCoP can start from an "IOCoP generator", *i.e*; a more traditional, transitional organisational form (e.g., an alliance or an inter-organizational working group) that encourages the emergence of inter-organizational links. In turn, these individual links can lead to the development of an IOCoP. | IOCoP participants do not only seek to counter or limit the obsolescence of know-how, but also to improve their practices. Members of the community take part in a "collectivisation" of their individual knowledge to contribute to the creation of a collective learning or output, with a value superior to that which could have been created by the sum of individual outputs. They develop a shared book of resources, such as tools, documents, routines, specific vocabulary, stories, symbols, and artifacts.<br>The IOCoP progressively prosper and become institutionalized. Once institutionalized, an IOCoP generates learning for its members, as well as, indirectly, for their respective organizations. | 1. The IOCoP can cease to exist altogether, if one or several organizations decide that its members should refrain from participating anymore.<br>2. The IOCoP could become "dormant": it continues to function nominally, but does not produce significant new learning.<br>3. Resulting from the decision of one or several organizations to go one step further, the IOCoP can be developed as a more traditional and more structured organizational form (research consortium, alliance…) than the IOCoP. |

*Figure 4 The stages of evolution of an Inter-Organisational Community of Practice or IOCoP (Adapted from Moingeon et al., 2006*[3]*)*

This decomposition of the IOCoP lifecyle is based on the three-stage model of organisational forms (Quinn and Cameron, 1983; Jawahar and McLaughlin, 2001). Note that (Moingeon et al., 2006) defined an IOCoP by using comparisons with other well-known organizational forms. Doing this, they were leaded to review and expand the framework proposed by (Wenger *et al*., 2002): to analyse organisational forms (See Figure 5).

[3] We designed this figure by converting some parts of the text of Moingeon et al., 2006) into a table; we also made some adaptations to the selected parts.

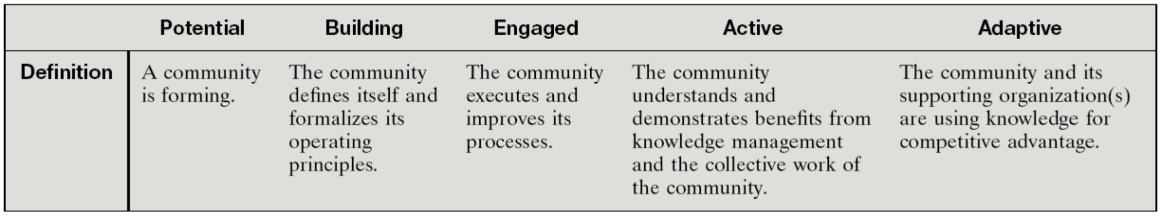


| | *Who belongs?* | *What is the purpose?* | *How clear are the boundaries?* |
|---|---|---|---|
| ***IOCoP*** | Individuals from different organizations | To create, expand, and exchange knowledge, and to develop individual skills for a professional practice | Fuzzy |
| ***CoP*** | Individuals from the same organization | To create, expand, and exchange knowledge, and to develop individual skills for a professional practice | Fuzzy |
| ***Club / Association*** | Individuals | To create, expand, and exchange knowledge, and to develop individual skills for a non-professional practice | Defined |
| ***Firm*** | Individuals from the same organization | Production of goods and/or services | Defined |
| ***Alliance*** | Firms | Production of goods and/or services | Defined |
| ***Network*** | Individuals (friends and business acquaintances) or firms | To receive and pass on information | Undefined |

*Figure 5 The (Wenger* et al.*, 2002) analysis framework of organisational forms revisited by (Moingeon et al., 2006).*

### 2.1.1.2. Capability-Maturity Models of CoP Evolution

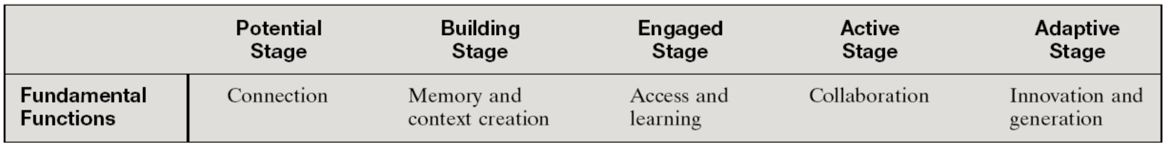
This second category of CoP evolution models is well represented by the **(Gongla and Rizzuto, 2001) model of community evolution**. In this model, communities of practice are seen as developing through five stages: potential, building, engaged, active, and adaptive (see Figure 6). This model "is more similar in overall intent to, for example, the capability maturity models developed by the Systems Engineering Institute for assessing software organisations than to the life-cycle-type development model for communities of practice, such as Wenger and McDermott describe" (Gongla and Rizzuto, 2001). In this model, "a community can mature and dissolve at any one of the evolution stages beyond the initial formation level. The model describes instead how communities transform themselves, becoming more capable at each stage, while at the same time maintaining a distinct, coherent identity throughout" (*Ibid.*).



| | Potential | Building | Engaged | Active | Adaptive |
|---|---|---|---|---|---|
| **Definition** | A community is forming. | The community defines itself and formalizes its operating principles. | The community executes and improves its processes. | The community understands and demonstrates benefits from knowledge management and the collective work of the community. | The community and its supporting organization(s) are using knowledge for competitive advantage. |

*Figure 6 Community evolution model definition (Gongla and Rizzuto, 2001)*

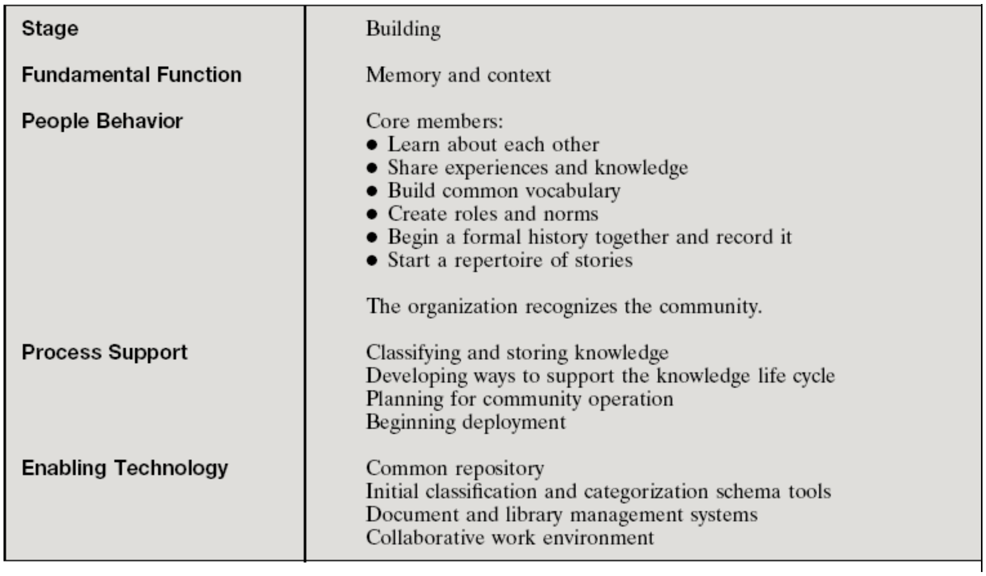
Each stage has its defining characteristics as well as an underlying fundamental function. Figure 7 shows the functions of the different stages.



| | Potential Stage | Building Stage | Engaged Stage | Active Stage | Adaptive Stage |
|---|---|---|---|---|---|
| **Fundamental Functions** | Connection | Memory and context creation | Access and learning | Collaboration | Innovation and generation |

*Figure 7 Fundamental functions for the stages of evolution (Gongla and Rizzuto, 2001)*

The main defining characteristics of each stage are : (1) the *behaviour of people* (people refers to “social individuals with their individual and group behaviours, as well as the larger organisational behaviour influence vis-à-vis a community” (*Ibid.*)), (2) the *degree and type of process support* (processes refer to the “sets of documented steps with clearly defined roles and activities for people to perform” (*Ibid.*)), and (3) the *types of technology encountered* (technology refers to “the application of science and the body of information system knowledge that we use to fashion tools, practice knowledge arts, and extract data and information” (*Ibid.*)). Figure 8 describes the characteristics of the Building stage of community evolution.



| | |
|---|---|
| **Stage** | Building |
| **Fundamental Function** | Memory and context |
| **People Behavior** | Core members:<br>● Learn about each other<br>● Share experiences and knowledge<br>● Build common vocabulary<br>● Create roles and norms<br>● Begin a formal history together and record it<br>● Start a repertoire of stories<br>The organization recognizes the community. |
| **Process Support** | Classifying and storing knowledge<br>Developing ways to support the knowledge life cycle<br>Planning for community operation<br>Beginning deployment |
| **Enabling Technology** | Common repository<br>Initial classification and categorization schema tools<br>Document and library management systems<br>Collaborative work environment |

*Figure 8 Building stage enablers that promote memory and context (Gongla & Rizzuto, 2001)*

### 2.1.1.3. Using the Models of CoP Evolution to account for the Evolution of CoP Knowledge

The models of CoP evolution presented above can be used to account for the evolution of the CoP knowledge we are interested in, *i.e.* the knowledge materialized in a CoP memory (among which are ontologies and semantic annotations). The idea is to identify in the models, or to infer, the types of knowledge evolution occurring during the evolution cycle of a CoP, and the events that have orginated or could originate the evolution of CoP knowledge.

For example, if we consider the transition from the stage *Potential* to the stage *Building* in the evolution cycle model of (Gongla and Rizzuto, 2001), we can identify a goal/process change: the change is here that the CoP members are no further trying to locate one another to form the community, but have to learn about one another, to share some experiences. “They must learn to talk to one another using words in the same way and build a common vocabulary and common understanding.” (Gongla and Rizzuto, 2001).

## 2.1.2. Evolution of the CoP Knowledge

We will now present some examples of the events that may originate the evolution of CoP knowledge, and some examples of knowledge evolution resulting from these events. To get more examples, it would be necessary to apply our eliciting approach in a systematic way. Note that our examples are “generic” or come from CoPs studied by the authors of the CoP evolution models. Getting more examples would be a question of finding instances of such changes in PALETTE CoPs, *e.g.*, through validating the existing examples directly with PALETTE CoPs’ members.

### 2.1.2.1. Events Leading to Community Knowledge Evolution

From a stage of the CoP evolution cycle to another, or within the same stage, the categories of events that may occur are, among others:

- *People (member or group) change.* (a) The composition of the CoP itself may change. Examples: a new member may enter the community, or an existing member may leave the community; the CoP may split into sub-communities. (b) The members themselves may change.
- *Behaviour change.* Example: CoP members may change their level of engagement towards the CoP, or their level of trust towards the other CoP members.
- *Goal/Process change.* Example: if we consider the transition from the stage *Potential* to the stage *Building* in the evolution cycle model of (Gongla and Rizzuto, 2001), the change is here that the CoP members are no further trying to locate one another to form the community, but have to learn about one another, to share some experiences. “They must learn to talk to one another using words in the same way and build a common vocabulary and common understanding.” (*Ibid.*).
- *Artefact change.* The artefacts (documents, software, etc.) handled by the CoP members may change: some existing artefact may evolve, or some new artefact can be introduced in the community.
- *Domain change.* Due to a re-examining of the community’s desired scope, the domain of practice/knowledge characterising the CoP may change.
- *Knowledge change.* A knowledge change may be the cause of some other knowledge change.

### 2.1.2.2. Types of Community Knowledge Evolution Resulting from the Events

Generally speaking, we can say that the different events reported above may lead, among others, to the following types of knowledge changes:

- *Knowledge content change.* To take the last event cited above, the change of some kind of knowledge may induce a change of some other kind of knowledge. For example, the change of a given ontology may lead to a change of the annotations previously done with this ontology (more details below). Another example is the arrival of a new member in the community who is a specialist of a sub-domain not covered by the current members of the CoP; this arrival may lead to the introduction of new knowledge in the CoP.
- *Knowledge value change.* Example: some low-valued knowledge (*e.g.*, a knowledge considered as useless for the community at time *t*) may acquire a higher value later on (*e.g.*, the same knowledge may be considered as useful by the community at time *t* + 1 because community's goals have changed). This may occur when a CoP member comes to have more confidence in some other CoP member, and to accept to give more value to the knowledge produced by this member, and as a result to accept to use this knowledge. (For details on the issue of knowledge value, see Chapter 3 "Knowledge Evaluation Service" of this deliverable.)
- *Knowledge management process change.* Example: the transition from the stage *Coalescing* to the stage *Maturing* in the CoP evolution model of (Wenger et al., 2002) implies to shift from sharing knowledge to organising and stewarding[4] knowledge.
- *Knowledge management tool change.* Example: to help CoP members wishing to share their experiences and learn to talk to one another using words in the same way, some tool can be introduced "for designing and maintaining whatever taxonomy is appropriate for the CoP domain of knowledge. If that taxonomy support can be linked with other organisational taxonomy efforts, all the better" (Gonga and Rizzuto, 2001).

#### 2.1.2.3. The Types of Community Knowledge Evolution Assisted by the Knowledge Evolution Service

The types of community knowledge evolution assisted by the Knowledge Evolution Service envisioned in the Palette project are:

- ontology evolution;
- annotations evolution.

Because the ontology may change to fit with the community's lifecycle, the need for the ontology evolution is unavoidable. (Stojanovic et al., 2002b) showed that a modification in one part of an ontology can impact the consistency of other parts of the same ontology, in the dependent ontologies and the applications using this modified ontology. In particular, changes in ontology can affect the consistency of semantic annotations which use the concepts or the properties defined in this modified ontology.

Inspired from researches on the database schema evolution (Roddick, 1996), on the ontology evolution (Stojanovic, 2004) and on the ontology versioning (Klein et al., 2002; Klein, 2004), we consider the *semantic annotation evolution* as "a process of adjustment of semantic annotation to the generated inconsistencies because of the changes on the ontology or the annotation itself" (Luong and Dieng-Kuntz, 2007).

[4] *Knowledge Stewarding* refers to a subset of knowledge management processes, such as selecting and configuring knowledge, as well as supporting its use in the practice of the community (our definition, adapted from a definition of the notion of "technology stewarding" provided on the Website http://technologyforcommunities.com/, a site about a forthcoming book by Wenger, White and Smith: *Stewarding Technology for Communities*).

## 2.2. Evolution of an Ontology

Some authors classify the causes of ontology changes according to the definition of an ontology. According to (Gruber, 1993), an ontology is a specification of a conceptualisation of a domain. Therefore, (Klein and Fensel, 2001) consider the following levels of causes for ontology changes:

- conceptual change: is a change in the conceptualisation;
- specification change: is a change of the specifications of a conceptualisation;
- representation change: is a change in the representation of the specifications of a conceptualisation.

As for (Flouris, 2006) in his work on the ontologies evolution, he identified two kinds of changes on an ontology: the changes on the conceptualisation and changes on the domain. These two types of changes are not rare. Indeed, the domain conceptualisation can change because of a new observation, a change of viewpoint, a change in the use of the ontology or yet the new access operations to information that were not known, etc. As well, the domain itself can change because the real world is rather dynamic and evolves over time.

The causes of ontology changes are also distinguished by the heterogeneity levels of an ontology (Klein, 2004):

- the language level heterogeneity: consists of the heterogeneity of the mechanisms used to define the classes and properties of an ontology;
- the ontology (model) level: concerns the domain described in the ontology. The heterogeneity can represent the different possible manners to model the domain.

These two levels may also explain the kinds of problems that are typically met and interfere with the combined use of ontologies that have been built independently.
In (Visser et al., 1997), the authors distinguish these two levels, under different designations, respectively non-semantic and semantic levels.

Moreover, the modular nature of the ontologies constitutes a factor of change and evolution. Indeed, the ontologies are usually built in a distributed and modularised way, making each asynchronous modification of a component of the ontology, likely to affect the consistency of the ontology as a whole.

Therefore, the ontology change has to be propagated to the dependent ontologies, the ontological instances and the software/applications using this modified ontology (Stojanovic et al., 2002b). In our approach, we are interested in the propagation of the ontological changes to the relying instances and semantic annotations.

### 2.2.1. Ontology Evolution Scenarios

In (Haase et al., 2005), the authors identify four situations where inconsistencies resolution methods are required to deal with evolving ontologies, these situations are:

- changing a consistent ontology can lead to potential inconsistencies in this ontology (this occurs mainly when we deal with ontologies that need maintenance during their evolution);
- reusing an inconsistent ontology (this occurs when we cannot check the consistency of the ontology source);
- evolving the model level and the instance level of the ontology separately, without

synchronising the changes;
- replacing an ontology version by an inconsistent version.

These situations can be considered as ontology evolution scenarios. There are also other scenarios of evolution for evolving ontologies when merging or aligning operations are performed (Bruijn et al., 2004).

### 2.2.2. Ontology Evolution Process

In (Stojanovic et al., 2002a), the authors present a cyclic ontology evolution process of six steps:

- *changes capturing*;
- *changes representation*, using suitable formats;
- *semantics of changes* (changes resolution) allowing to solve the changes in the ontology in a systematic way, preventing the ontology from turning inconsistent (structurally or semantically);
- *changes implementation*, where the required and derived changes are applied to the ontology;
- *changes propagation*, ensuring the consistency of the dependent parts after an ontology update (the dependent parts consist of the dependent ontologies, the instances, as well as the applications that use the initial ontology);
- *changes validation*.

In our work, we focus on the representation of changes (Section 2.3), semantics of changes (changes resolution) and the propagation of the ontology evolution to the dependent annotations (in Sections 2.3 to 2.6).

## 2.3. Changes Representation

During the evolution, changes must be identified and represented in suitable formats. (Stojanovic, 2004) classified the three levels of ontology changes:

- Elementary change: is an ontology change that adds or removes only one entity of the ontology model.
- Composite change: is an ontology change that creates, removes or changes the neighbourhood of an ontology entity.
- Complex change: is an ontology change that can be decomposed into any combination of at least two elementary and composite ontology changes. To illustrate, (Stojanovic, 2004) cites the example of moving a set of sibling concepts to a different location, which moves two or more siblings concepts in the concept hierarchy to a different parent in this hierarchy, thus they remain siblings, but under a different parent.

In the same way, the ontology operations are also divided according to two dimensions (Klein, 2004).

i. Atomic vs. composite operations: a composite operation performs several basic operations in one step. (Klein, 2004) considers the operation of moving a set of sibling concepts as being such a kind of composite operation.
ii. Simple vs. rich operations: a rich change incorporates information about the implication of the operation on the logical model of the ontology (Klein, 2004). For example, a rich change might specify that the range of a property is enlarged. Whereas a simple change can be detected merely by analysing the structure.

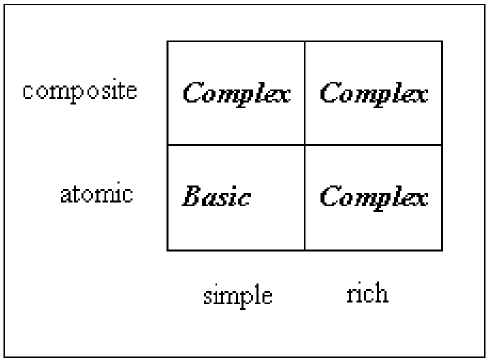
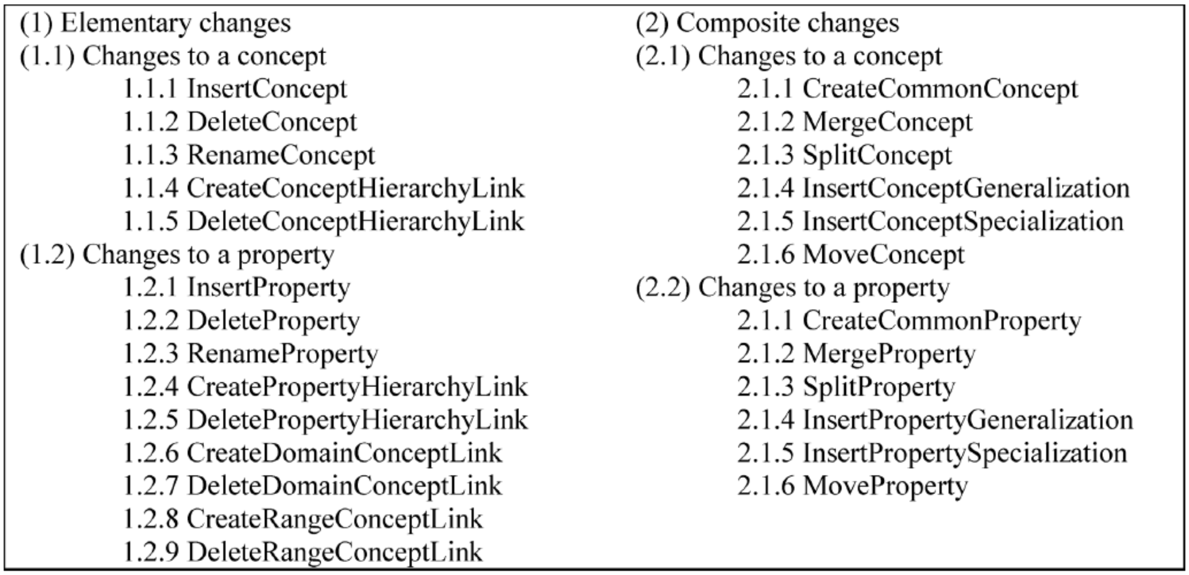


*Figure 9 The relation between atomic-composite and simple-rich operations (Klein, 2004)*

In our approach, we study all the changes in ontology which can affect the consistency of its dependent parts, of other dependent ontologies and particularly, the ontological changes affecting consistency on the concerned semantic annotations. We have built a list of the necessary changes for the process of ontology and semantic annotations evolution. We classify these changes in two types:

i. Elementary change: is an ontology change that modifies only one entity of the ontology model and are atomic (e.g., RenameConcept, DeleteConcept, DeleteDomain-ConceptLink, etc.);
ii. Composite change: is an ontology change that modifies several entities of the ontology model and can be broken up (e.g., MergeConcept, DivideConcept, etc.).

The detailed list of the changes as well as their definitions can be found in the Appendixes A and B of (Luong, 2007).



(1) Elementary changes
(1.1) Changes to a concept
1.1.1 InsertConcept
1.1.2 DeleteConcept
1.1.3 RenameConcept
1.1.4 CreateConceptHierarchyLink
1.1.5 DeleteConceptHierarchyLink
(1.2) Changes to a property
1.2.1 InsertProperty
1.2.2 DeleteProperty
1.2.3 RenameProperty
1.2.4 CreatePropertyHierarchyLink
1.2.5 DeletePropertyHierarchyLink
1.2.6 CreateDomainConceptLink
1.2.7 DeleteDomainConceptLink
1.2.8 CreateRangeConceptLink
1.2.9 DeleteRangeConceptLink

(2) Composite changes
(2.1) Changes to a concept
2.1.1 CreateCommonConcept
2.1.2 MergeConcept
2.1.3 SplitConcept
2.1.4 InsertConceptGeneralization
2.1.5 InsertConceptSpecialization
2.1.6 MoveConcept
(2.2) Changes to a property
2.1.1 CreateCommonProperty
2.1.2 MergeProperty
2.1.3 SplitProperty
2.1.4 InsertPropertyGeneralization
2.1.5 InsertPropertySpecialization
2.1.6 MoveProperty

*Figure 10 Taxonomy of ontology changes*

To have the explicit representation of changes, we developed the so-called *evolution ontology* enabling to formalise the ontology changes (see Figure 10) which can occur during the evolution. We have also created in this evolution ontology some necessary properties such as *hasVersionBefore*, *hasDate*, *hasAuthor,...*which allow to model and trace all the changes performed on an ontology (what changes, when, by whom and how).

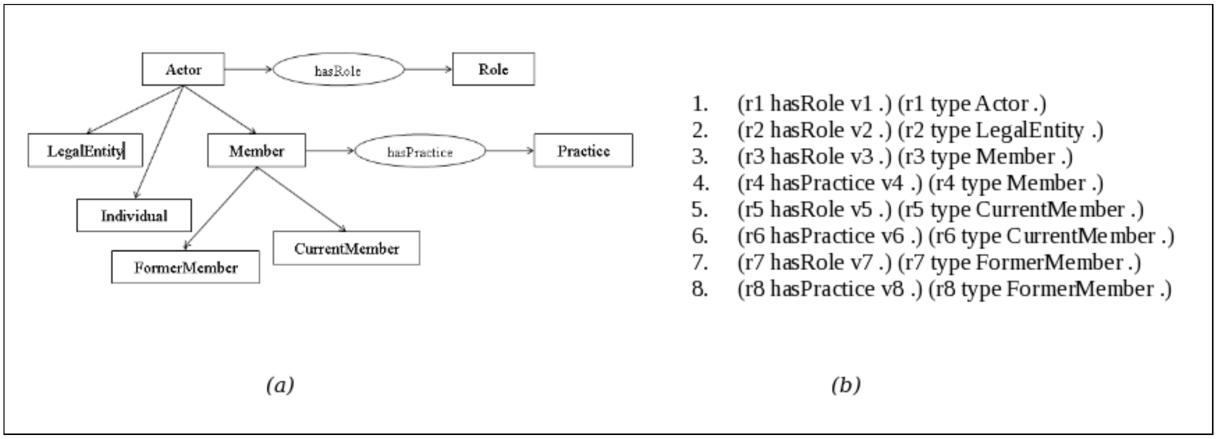
## 2.4. Evolution of Annotations

When we make some modifications on the ontology, not only parts of it might fall into inconsistent state but also its semantic annotations could be influenced. In these cases, the construction of evolution strategies allows us to control the inconsistencies that might have appeared.

### 2.4.1. Annotations Evolution Scenarios

Now, we identify some scenarios that can impact the consistency of semantic annotations.

- *Scenario 1*: User makes some modifications on her underlying ontology results in a new ontology version. Because of these ontological changes, semantic annotations may be affected leading to the inconsistent state with respect to the new ontology version.
- *Scenario 2*: User changes her annotations without referring to the underlying ontology which is used by these annotations. The modified annotations can be inconsistent with respect to the ontology.
- *Scenario 3*: User imports annotations from another source. These imported annotations and the old annotations are based on the same ontology. There would be similar annotations (or triples) which describe the same resource.
- *Scenario 4*: User makes migration of the annotation base from another representation formalism (for example, migration of RDF(S) toward OWL-Lite). There would be inconsistencies of syntax on these annotations.



1. (r1 hasRole v1 .) (r1 type Actor .)
2. (r2 hasRole v2 .) (r2 type LegalEntity .)
3. (r3 hasRole v3 .) (r3 type Member .)
4. (r4 hasPractice v4 .) (r4 type Member .)
5. (r5 hasRole v5 .) (r5 type CurrentMember .)
6. (r6 hasPractice v6 .) (r6 type CurrentMember .)
7. (r7 hasRole v7 .) (r7 type FormerMember .)
8. (r8 hasPractice v8 .) (r8 type FormerMember .)

*Figure 11 A part of ontology (a) and semantic annotations based on this ontology (b)*

However, we find that the first scenario is likely to be the most encountered in reality. In this deliverable, we will then focus on this particular context: changes in underlying ontology can impact to the consistency of the semantic annotations which are using the defined terms in this underlying ontology. We use the RDFS[5] language to model the ontology and to describe a triple *(s p v.) (subject property value)* of the annotation in RDF[6]. This triple represents a

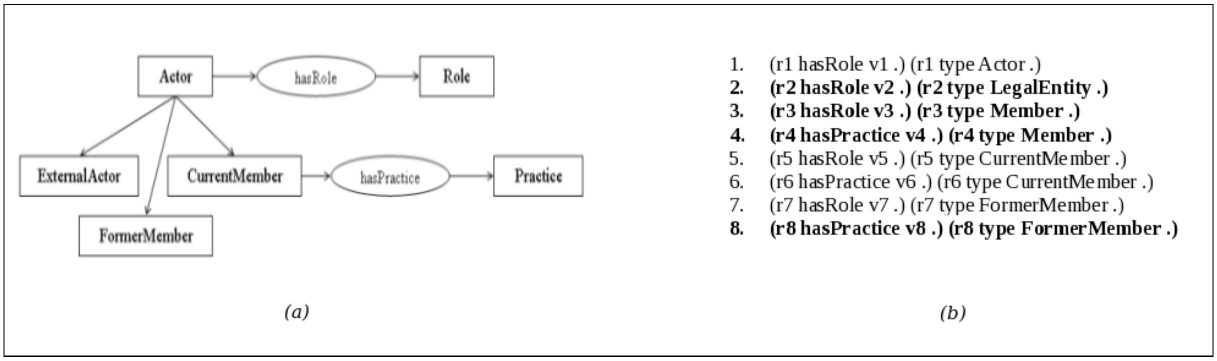
[5] RDF Vocabulary Description Language 1.0: RDF Schema (http://www.w3.org/TR/rdf-schema/)

[6] Resource Description Framework (http://www.w3.org/RDF/)

statement on the resource which can be expressed as a subject *s* has a property *p* whose value is *v*.

We examine an example with a part of O'CoP ontology described in the deliverable D.KNO.02 (Tifous et al., 2007) (see Figure 12a) containing the concept *Actor* which is domain of the property *hasRole* and is also father concept of its sub-concepts *LegalEntity*, *Individual* and *Member*. The *Member* concept, having two sub-concepts *CurrentMember* and *FormerMember*, is domain of the property *hasPractice*. The *Role* and *Practice* concepts are ranges of the corresponding properties *hasRole* and *hasPractice*. In addition, we have also some triples in semantic annotations based on this part of ontology (see Figure 12b).

Let us assume that this part of ontology was modified by removing the *Member* concept, its two sub-concepts are reconnected to the concept *Actor* and the property *hasPractice* receives from now on the *CurrentMember* concept as its domain. Moreover, the concepts *LegalEntity* and *Individual* are merged and replaced by the new *ExternalActor* concept.
After having applied these changes, we obtain a new ontology version (see Figure 10a) in which some elements were changed compared to its old version. Some triples become inconsistent now (see Figure 10b) because of the loss of the reference links toward the corresponding concepts in the ontology before its modification. We will analyse the causes and its solutions for the inconsistent triples (e.g., triples in lines 2, 3, 4 and 8) later in the section.



1. (r1 hasRole v1 .) (r1 type Actor .)
2. **(r2 hasRole v2 .) (r2 type LegalEntity .)**
3. **(r3 hasRole v3 .) (r3 type Member .)**
4. **(r4 hasPractice v4 .) (r4 type Member .)**
5. (r5 hasRole v5 .) (r5 type CurrentMember .)
6. (r6 hasPractice v6 .) (r6 type CurrentMember .)
7. (r7 hasRole v7 .) (r7 type FormerMember .)
8. **(r8 hasPractice v8 .) (r8 type FormerMember .)**

*Figure 12 The modified ontology (a) and the semantic annotation become inconsistent (b)*

### 2.4.2. Semantic Annotation Evolution Process

After having applied changes to the ontology, this ontology evolves to a new version. We distinguish two cases of ontology evolution which can influence the consistent state of semantic annotation:

- ontology evolution with trace;
- ontology evolution without trace.

Trace concerns the changes which were carried out between two versions of the ontology.

For both cases, we have proposed a process allowing us to find the annotations related to the modified ontology in an annotation base and particularly the inconsistent annotations affected by the ontological changes (cf. Figure 13). This process contains two main steps: (i) inconsistency detection and (ii) inconsistency resolution of annotations.

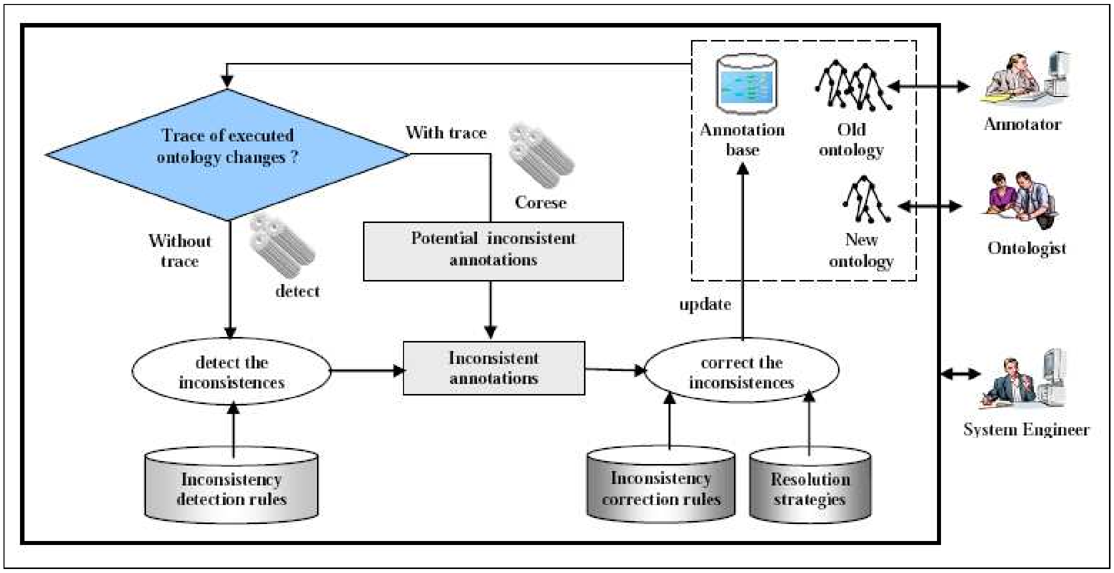

Trace of executed ontology changes ?
With trace
Corese
Without trace
detect
Potential inconsistent annotations
Annotation base
Old ontology
New ontology
update
Annotator
Ontologist
System Engineer
detect the inconsistences
Inconsistent annotations
correct the inconsistences
Inconsistency detection rules
Inconsistency correction rules
Resolution strategies

*Figure 13 Process of ontology change propagation*

### Annotation inconsistency detection

If we know the history of changes that were carried out, we can use Corese, a semantic search engine (Corby et al., 2004), that allows us to query the annotation base taking into account the concept hierarchy and the relation hierarchy defined in the ontologies. Corese can retrieve from the existing annotation base the annotations related to the modified ontology as well as the potential inconsistent annotations (they may include both related consistent and inconsistent annotations) (Luong and Dieng-Kuntz, 2007). A potential inconsistent annotation means that it relates to the ontological change but its consistency constraint has not been verified. An annotation is inconsistent if it violates the consistency constraints defined for the annotations (Luong and Dieng-Kuntz, 2007).

Otherwise, if the history of change was not kept, we apply inconsistency detection rules in order to detect the actual inconsistent annotations.

### Annotation inconsistency resolution

After having determined the inconsistent annotations, they will be repaired by applying inconsistency correction rules in case that we know how to apply an evolution strategy corresponding. So, we can restore the consistent state for the influenced semantic annotations (Luong and Dieng-Kuntz, 2007). However, we cannot always choose a way to correct automatically inconsistent annotations. For example, if a concept is used in an annotation but has been removed from the new ontology version, a decision should be made concerning the related annotations: either to delete the annotations that are related to the removed concept or to replace the removed concept in the related annotation triples by another concept of the ontology; in this case, the system should be able to propose a list of available and relevant concepts of the new ontology version. Thus, the process of solving inconsistencies can be done with the user intervention to choose a suitable solution for completing the inconsistency resolution. In order to help user in this task, we have established several possible solutions to solve the propagation of ontological changes to their semantic annotations in order to keep consistency status.

## 2.5. Procedural Approach for Ontology Change Propagation on Semantic Annotations

The procedural approach aims at detecting and solving the inconsistencies on the semantic annotations, generated from the ontology evolution, when the trace of the ontology changes is preserved.

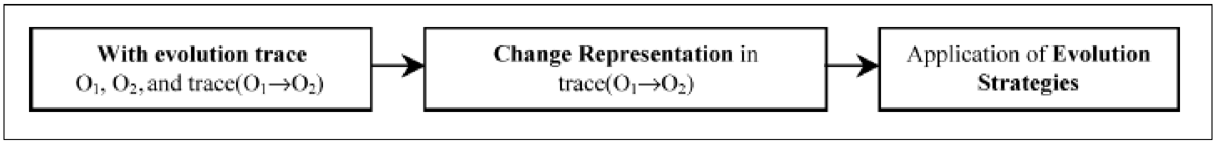

**With evolution trace**
$O_1$, $O_2$, and trace($O_1 \rightarrow O_2$)
**Change Representation** in
trace($O_1 \rightarrow O_2$)
Application of **Evolution**
**Strategies**

*Figure 14 Ontology evolution with a trace between two versions*

The ontology changes trace consists of all the executed changes, as well as the results of operations between two versions O1 and O2 of the ontology. This information is preserved in a log file of changes, which contains the trace of changes *trace(O1 → O2)* carried out between these ontology versions. These executed changes are represented in a more formal way according to our classification of changes and they are expressed in terms of the *evolution ontology*.

This log of changes is quite similar to the evolution log presented in (Stojanovic, 2004) which tracks the history of an ontology as an ordered sequence of ontology changes. Then, we apply **evolution strategies** corresponding to each ontological change to restore the consistent state for the influenced semantic annotations.

### 2.5.1. The Evolution Trace

In order to represent ontology changes in a formal way, we have built the so-called *evolution ontology* that defines formally the change classification and the ontological relations between these changes and the entities (concepts and properties) of the evolving ontology. This ontology also formalises information regarding the process of ontology evolution and semantic annotations evolution.

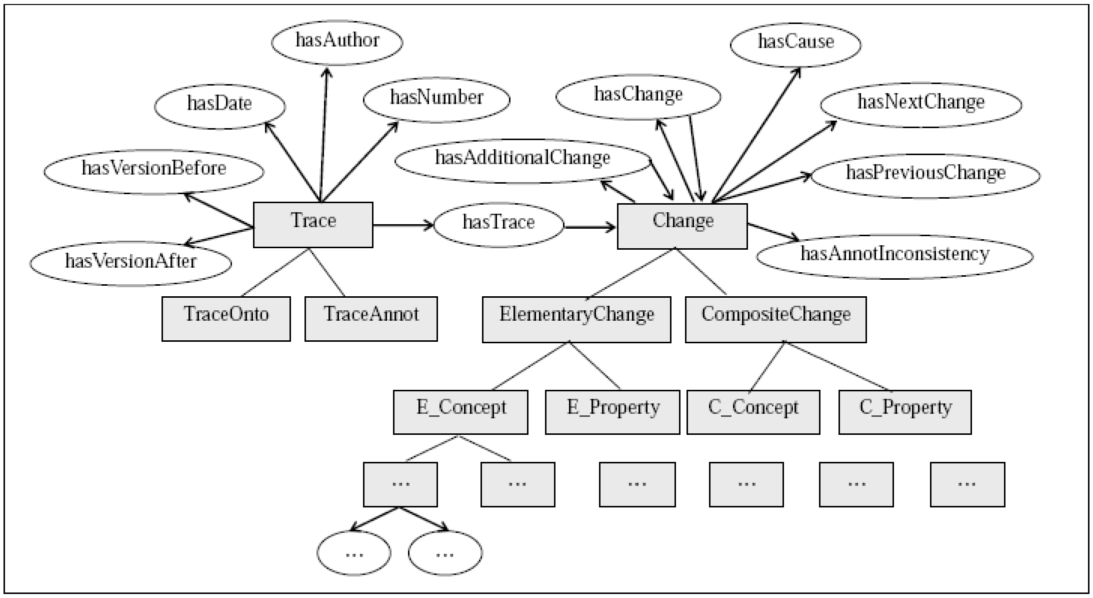


*Figure 15 A part of the Evolution Ontology*

Figure 15 illustrates a part of this *evolution ontology*. In order to model the trace of executed changes in ontology and the trace of updated semantic annotations, we created the concept *Trace* and its sub-concepts *TraceOnto* and *TraceAnnot*.

For each kind of trace, we will save relevant information describing the process of evolution, using properties of the evolution ontology such as the author who made changes (thanks to the property *hasAuthor*), the identification and the date of executed trace (thanks to the property *hasDate*), etc. We also distinguish the types of different changes using the hierarchy of the concept *Change*, such as elementary changes (*ElementaryChange* concept), composite changes (*CompositeChange* concept). Moreover, we can distinguish changes that can cause inconsistencies in semantic annotations by the values assigned to the property *hasAnnotInconsistency*.

To describe each change on an element of the ontology, we have created properties to represent relations between this change and the modified element. For example, the change *CreateHierarchyConceptLink(c1,c2)* aims at creating a hierarchy relation between the concepts c1 and c2. This change is made more explicit using the properties *hasSuperConcept* and *hasSubConcept*.

This changes formalisation through the evolution ontology is used to generate an evolution trace file, in which the evolution trace is represented as semantic annotations based on the terms defined in the *evolution ontology*.

## 2.5.2. Evolution Strategies

(Stojanovic, 2004) proposes some evolution strategies for ontology in which she defines resolution point and elementary strategies for each case of ontology change. These evolution strategies ensure that the ontology and other dependent parts of it will remain consistent after having applied some changes to the ontology. Moreover, they are also responsible for avoiding the illegal changes. However, her evolution strategies only cover some effects of simple changes and she did not mention the evolution strategies for semantic annotations.

In this section, we present a complete set of evolution strategies which tries to solve the inconsistencies caused by all the two types of ontological changes: *simple* and *composite* changes. For each ontological change having an impact on the consistency of annotations, we have built an equivalent strategy to correct the inconsistencies appearing in the semantic annotations. To illustrate our evolution strategies, we examine one of the cases of change: suppression of a middle concept *c* in the ontology.

Let us assume that *c2* is father concept of the middle concept *c*, *c0* is root concept, *p* is a property which can receive the concepts *c*, *c2*, *c0* as its domain/range. The evolution strategies for this case are described in the table below (see Table 1):

***Table 1 Evolution Strategies for Ontology and Semantic Annotations in Case of Deletion of a Middle concept in the Ontology***

| **Evolution Strategies: For the Dependent Elements of Ontology** | **Evolution Strategies: For the Related Annotations** |
|---|---|
| **SO-1**: To process the instances of the deleted concept **c**: 4 options | **SA-1**: To process the triple containing instances of the deleted concept **c**: 4 options |
| (1) Delete all instances of the concept **c** | (1) Delete this triple |
| (2) Attach all instances of the concept **c** to its father concept **c2** | (2) If the father concept $c2 \in \text{domain}(p)$ or $c2 \in \text{range}(p)$ then replace the name of type **c** for its instances in triple by the name of type **c2**. Else, remove this triple. |
| (3) Attach all instances of the concept **c** to its root concept **c0** | (3) If the root concept $c0 \in \text{domain}(p)$ or $c0 \in \text{range}(p)$ then replace the name of type **c** for its instances by the name of type **c0** in triple. Else, remove this triple. |
| (4) Attach all instances of the concept **c** to any concept **cx** indicated by user | (4) If the indicated concept $cx \in \text{domain}(p)$ or $cx \in \text{range}(p)$ then replace the name of type **c** for its instances by the name of type **cx** in triple. Else, remove this triple. |
| **SO-2**: To process the subconcepts of **c**: 4 options | **SA-2**: To process the triples containing the resources of type **c**: |
| (1) Delete all subconcepts of **c** | (1) Apply SA-1 |
| (2) Attach all subconcepts of **c** to its father concept | (2) (3) (4) No changes |
| (3) Attach all subconcepts of **c** to its root<br>(4) Attach all subconcepts of **c** to any concept **cy** indicated by user | *Note: The evolution strategies must be repeated for all sub-concepts of* ***c*** *and for all the semantic annotations related to these sub-concepts.* |
| **SO-3**: To process the properties related to the deleted concept **c**: 2 options | **SA-3**: To process the triples containing the resources of type **c** and the property **p**: 2 options |
| (1) Remove **c** from domain (resp. range) of the property **p** | (1) Delete all triples containing the resource of type **c** |
| (2) If $c \in \text{domain}(p)$ (resp. range(p)) and there | (2) Replace the name of type **c** of resources by |

| | |
|---|---|
| does exist a concept **c2** which is **ancestor of c** such as $c2 \in \text{domain}(p)$, then remove **c** from domain of the property **p** | the name of its father concept type **c2** in annotation. |

## 2.6. Rule-based Approach for Ontology Change Propagation on Semantic Annotations

In the dynamic and distributed context of Semantic Web, it is not always that one can keep the trace between ontology versions. We can reuse the results of existing research on the ontology versioning allowing us to find the similarities and the differences *diff(O1, O2)* between two ontology versions O1 and O2 as well as the changes carried out between these versions (Klein et al., 2002; Klein, 2004). According to this approach, we can detect some executed ontological changes; we can then follow the procedures of resolution of the inconsistencies to repair the inconsistent semantic annotations.

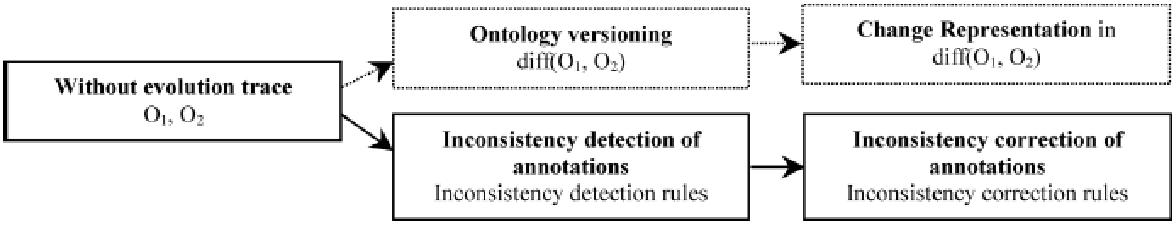

Without evolution trace
$O_1$, $O_2$
Ontology versioning
diff($O_1$, $O_2$)
Change Representation in
diff($O_1$, $O_2$)
Inconsistency detection of annotations
Inconsistency detection rules
Inconsistency correction of annotations
Inconsistency correction rules

*Figure 16 Ontology evolution without a trace between two versions*

However, we have proposed a rule-based approach, constructed from some consistency constraints that must be satisfied for any annotation model. Consistency is an attribute of a (logical) system that is constituted so that none of the facts deductible from the model contradicts another (Stojanovic, 2004). Therefore, we have proposed some **consistency constraints** that can be considered as an agreement among semantic annotations entities with respect to their underlying ontology.

Based on these consistency constraints, we have created some **inconsistency detection rules**, using the syntax of Corese rule language, to detect the real inconsistent annotations from a set of potential ones. A real inconsistent annotation means that it violates the consistency constraint defined for the annotation.

After having determined real inconsistent annotations, these will be repaired by applying **correction rules**. We have established all possible solutions that solve the propagation of ontological changes to their semantic annotations to keep consistency status.

### 2.6.1. Consistency Constraints

To describe the inconsistency of semantic annotation, we define what a *consistency constraint* is and what an *inconsistent semantic annotation* is. We also give a definition of an *annotation model* that is based on the data model RDF presented in (Miller and Manola, 2004).

*Definition 1.* A semantic annotation is defined to be *inconsistent* with respect to its ontology model if it violates the consistency constraints defined for annotation model.

*Definition 2.* A *consistency constraint* ensures a consistent agreement among semantic annotations entities with respect to their underlying ontology.

*Definition 3.* An *annotation model* is a t-uple SA := ($R_A$,$C_A$,$P_A$,L,$T_A$) where:

- $R_A$ : set of resources
- $C_A$ : set of concept names defined in ontology $(C_A \subset R_A)$
- $P_A$ : set of property names defined in ontology $(P_A \subset R_A)$
- L : set of literal values
- $T_A$ : set of triples (s, p, v.) where $s \in R_A$, $p \in P_A$ and $v \in (R_A \cup L)$

To express consistency constraints, we take the notation from (Miller and Manola, 2004) to describe an RDF triple in annotation as a triple *(s p v.)*. This triple makes statement about a resource and can be expressed as a subject *s* has a property *p* whose value is *v*. We use the primitive *rdf:type* to indicate a resource as instance of specific types or classes (e.g., resource has type Class or Property) and other primitives with prefix *rdfs:* to describe classes or relationship among these classes in the ontology.

1. *Constraint on concept*: all the concepts used in the annotation must be defined before in the ontology.
   *(s rdf:type c)* $\Rightarrow$ *(c rdf:type rdfs:Class)*

2. *Constraint on property*: all the properties used in the annotation must be defined before in the ontology.
   *(s p v.)* $\Rightarrow$ *(p rdf:type rdf:Property)*

3. *Constraint on property domain*: the resource which is the domain of a property in the annotation must be compatible with the domain of the corresponding property defined in the ontology.
   *(p rdf:type rdf:Property)* $\wedge$ *(p rdfs:domain d)* $\wedge$ *(s p v.)*
   $\Rightarrow$ *(s rdf:type d)* $\vee$ *(* $\exists$ *dx, (dx rdfs:subClassOf d)* $\wedge$ *(s rdf:type dx))*

4. *Constraint on property range*: the resource which is the range of a property in the annotation must be compatible with the range of the corresponding property defined in the ontology.
   *(p rdf:type rdf:Property)* $\wedge$ *(p rdfs:range r)* $\wedge$ *(s p v.)*
   $\Rightarrow$ *(v rdf:type r)* $\vee$ *(* $\exists$ *rx, (rx rdfs:subClassOf r)* $\wedge$ *(v rdf:type rx))*

5. *Constraint on datatype*: The data type of a value of property in the annotation must be compatible with the value of the corresponding property defined in the ontology.
   *(p rdf:type rdf:Property)*$\wedge$ *(p rdfs:range r)*$\wedge$ *(r rdf:type rdfs:Datatype)*$\wedge$ *(s p v.)* $\Rightarrow$ *(v rdf:type r)*

## 2.6.2. Inconsistency Detection Rules

We have established several inconsistency detection rules that we apply in this step to detect the real inconsistent annotations (that are obsolete with respect to a modified ontology considered as a reference), from the set of potential inconsistent annotations. These rules are based on the **consistency constraints** described above.

We have constructed some groups of rules allowing us to detect inconsistencies related to concepts, properties and domain/range. In the following paragraph, we present some rules which can be used to illustrate the above example. We use the notation from (Miller and Manola, 2004) to express these rules in RDF.

- *Group 1 (detection rules for a concept resource)*: If a concept is used in an annotation but it has not been defined in ontology, then this annotation leads to inconsistent state and is marked “inconsistent”.

  R-1: $\forall$ *(s p v.), (s rdf:type c)* $\wedge$ $\neg$ *(c rdf:type rdfs:Class)* $\Rightarrow$ note(inconsistent)
  R-2: $\forall$ *(s p v.), (s rdf:type c)* $\wedge$ *(c rdf:type rdfs:Class)* $\Rightarrow$ note(OK)

- *Group 2 (detection rule for a property resource)*: If a property is used in an annotation but it has not been defined in ontology, then this annotation leads to inconsistent state and is marked “inconsistent”.

  R-3: $\forall$ *(s p v.),* $\neg$*(p rdf:type rdf:Property)* $\Rightarrow$ note(inconsistent)
  R-4: $\forall$ *(s p v.), (p rdf:type rdf:Property*$\Rightarrow$ note(OK)

- *Group 3 (detection rule for a resource which is a domain of property)*: If a property p takes a resource of concept type c as its subject in annotation, but c is not compatible with the domain of p in ontology, then this annotation leads to inconsistent state and is marked “inconsistent”.

  R-5: $\forall$ *(s p v.), (s rdf:type c)* $\wedge$ *(p rdfs:domain d)* $\wedge$ $\neg$ *(c rdfs:subClassOf d)*
  $\Rightarrow$ note(inconsistent)
  R-6: $\forall$ *(s p v.), (s rdf:type c)* $\wedge$ *(p rdfs:domain d)* $\wedge$ *(c rdfs:subClassOf d)*
  $\Rightarrow$ note(OK)

- *Group 4 (detection rule for a resource which is a range of property)*: If a property p takes a resource of concept type c as its value in annotation, but c is not compatible with the range of p in ontology, then this annotation leads to inconsistent state and is marked “inconsistent”.

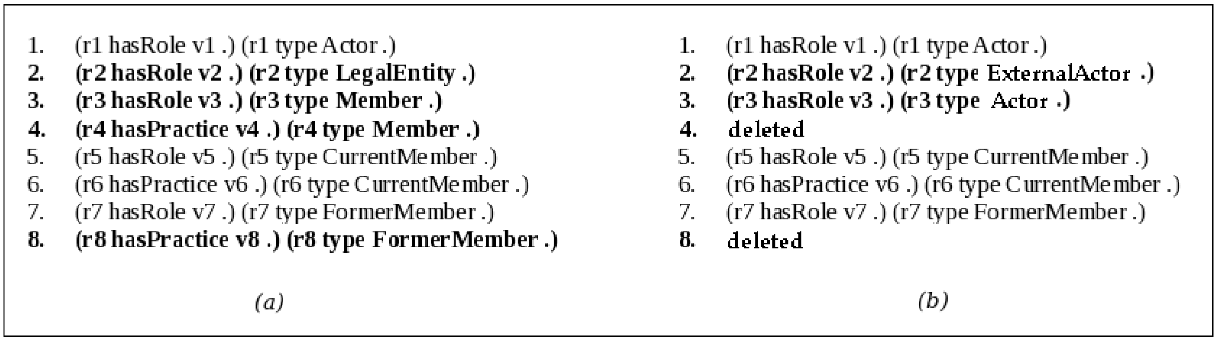
  R-7: $\forall$ *(s p v.), (v rdf:type c)* $\wedge$ *(p rdfs:range d)* $\wedge$ $\neg$ *(c rdfs:subClassOf d)*
  $\Rightarrow$ note(inconsistent)
  R-8: $\forall$ *(s p v.), (v rdf:type c)* $\wedge$ *(p rdfs:range d)* $\wedge$ *(c rdfs:subClassOf d)*
  $\Rightarrow$ note(OK)

With the constructed detection rules, we applied these detection rules on the set of annotations related to the modified ontology. With the rule R-1, we can detect the triples in lines 2, 3, and 4 (see Figure 17a) which become inconsistent because the loss of concept reference toward the concepts *LegalEntity* and *Member*. The rule R-5 detects the triples in line 8 inconsistent because the domain relation between the concept *FormerMember* and the property *hasPractice* was deleted in the new ontology version.

### 2.6.3. Inconsistency Correction Rules

After having collected all inconsistent annotations from a set of potential inconsistent annotations, we need to correct these inconsistencies by applying some correction rules on these annotations. These rules guide the execution of evolution strategies in which we specified how to propagate the change resolution to inconsistent annotations to keep an overall consistency (see Table 1).

Still taking the example of the O'CoP ontology and the dependent annotations respective changes: after having detected inconsistencies, we find that triples in lines 2, 3, 4, and 8 are in the inconsistent state (see Figure 17). Knowing that the change *DeleteConcept(Member)* leads to the loss of the concept reference of the resources in triples in lines 3 and 4 toward their ontological corresponding concepts, we can apply some below rules:



1. (r1 hasRole v1 .) (r1 type Actor .)
2. **(r2 hasRole v2 .) (r2 type LegalEntity .)**
3. **(r3 hasRole v3 .) (r3 type Member .)**
4. **(r4 hasPractice v4 .) (r4 type Member .)**
5. (r5 hasRole v5 .) (r5 type CurrentMember .)
6. (r6 hasPractice v6 .) (r6 type CurrentMember .)
7. (r7 hasRole v7 .) (r7 type FormerMember .)
8. **(r8 hasPractice v8 .) (r8 type FormerMember .)**

*(a)*

1. (r1 hasRole v1 .) (r1 type Actor .)
2. **(r2 hasRole v2 .) (r2 type ExternalActor .)**
3. **(r3 hasRole v3 .) (r3 type Actor .)**
4. **deleted**
5. (r5 hasRole v5 .) (r5 type CurrentMember .)
6. (r6 hasPractice v6 .) (r6 type CurrentMember .)
7. (r7 hasRole v7 .) (r7 type FormerMember .)
8. **deleted**

*(b)*

*Figure 17 Some inconsistent triples before (a) and after (b) update*

-RC-1: If SO-3 (2) is applied for ontology, then apply SA-3 (2) for the annotation. This rule will replace the name of concept type *Member* by the name of the concept type *Actor* in triple in line 3 because the relation between the property *hasRole* and the concept *Actor* is still kept.

-RC-2: If SO-3 (3) is applied for ontology, then apply SA-3 (2) for the annotation. This rule will remove the triple in line 4 because it does not exist relation between the property *hasPractice* and the concept *Actor* which is father of the deleted concept *Member*.

## 2.7. Implementation

### 2.7.1. Architecture of the Knowledge Evolution Support Prototype

We have developed the CoSWEM prototype (Corporate Semantic Web Evolution Management), a Web based prototype for supporting the semantic annotation evolution when its underlying ontology changes. Its architecture consists of the following main components:

- User component: manages the diverse roles of humans interacting with the system. To each role (profile) corresponds a view, we distinguish the following users' profiles:
  - User: it is the person who uses the system to manipulate her domain knowledge. She needs evolution management functionalities enabling to be aware of the changes;
  - Ontologist: she is the ontology provider, the person(s) who knows the domain and makes a formal representation of it. During the evolution, ontologists may have to modify some parts of the ontology in order to comply with the new needs of the community;
  - Annotator: she uses the provided ontologies to annotate the resources of the community;

  - System engineer: she is the administrator of the system; she controls the system and ensures the consistency and maintenance of the system. She is responsible for inconsistencies detection and correction mechanisms on which the system relies.
- Intermediate component:
  - Interface module: this module deals with human-machine interaction aspects. It allows the users to visualise and access the ontologies and semantic annotations, as well as the differences between an ontology versions, inconsistent annotations, etc. It also enables to use the correction of the annotations functionality;
  - Changes representation module: where the ontological changes are formally represented, based on the *evolution ontology*, described in Section 2.5.1;
- Evolution component:
  - Ontology evolution module: it allows to work with different versions of an ontology. The changes occurring from a version of the ontology to another are captured and formalised, so that they can be reused by the annotations evolution module;
  - Annotations evolution module: uses the information about the ontology changes in order to detect the semantic annotations that have become obsolete regarding the new ontology version. These inconsistencies can than be corrected with respect to the solution chosen by the user;
  - Evolution journal: enables to keep a trace and the history of the changes occurring between ontology versions. Once this information is saved, it is transformed into semantic annotations, relying on the *evolution ontology* and enabling to easily manipulate them and discover useful information about the evolution process.

### 2.7.2. Illustration

In our evolution management prototype, a function for the comparison of the differences between two ontology versions is implemented. Concretely, this function detects the concepts or properties that have been modified. Then, it can retrieve the possibly inconsistent annotations relying on the last version of the ontology.

Figure 18 shows two versions of a part of the O'CoP ontology. The left hand column represents the old version of the ontology, the right hand one represents the new version of the ontology. The CoSWEM prototype detects the differences between these two versions. It shows that the concepts *ACTIVITY*, *technicians* and *RESOURCE* have been modified such that they does not appear in the new version of the ontology (at least, they do not appear exactly the same way), this is illustrated using the red font colour. On the other hand, the concepts *Activity,_Coordination_task* and *Animation_task* have been added to the new version of the ontology, which is illustrated using the blue font colour.

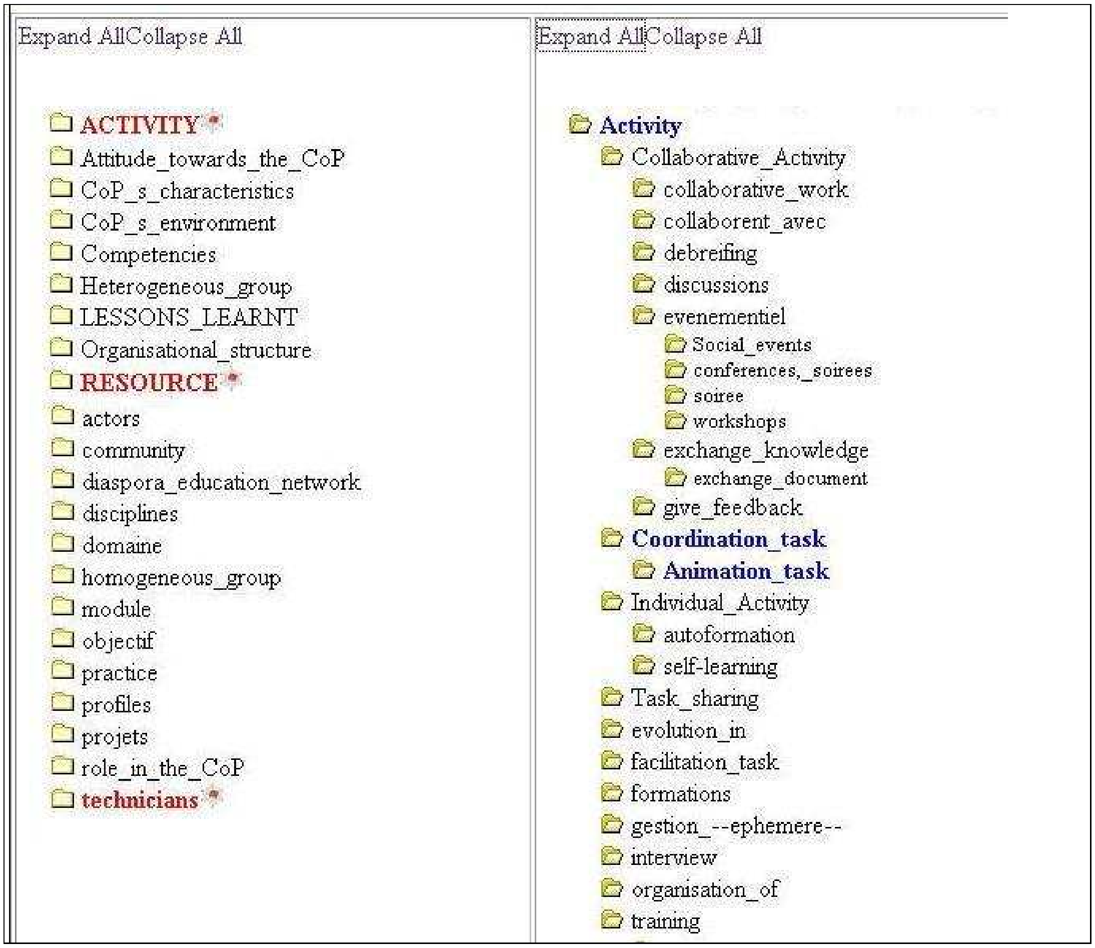
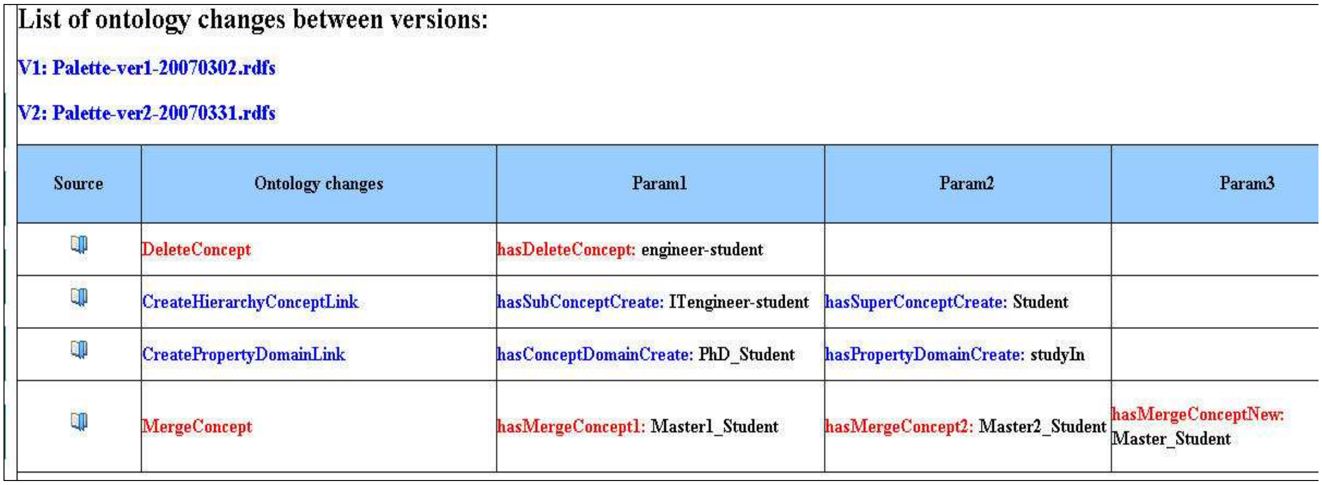


*Figure 18 Comparison of the concepts between two ontologies versions, without evolution trace, in the CoSWEM prototype*

To compare between two ontology versions, CoSWEM relies on the Corese search engine, which queries the two versions, then it compares the results of the queries on the versions, which enables to determine the structural differences (*i.e.* the different concepts and properties) between the versions of the ontology.

If the changes trace is kept, CoSWEM can give more details about the nature of the changes and how the concerned entities (concepts and properties) have been modified (see Figure 19), as well as the semantic annotations influenced by these ontological changes.

Then, CoSWEM provides an interface for the semi-automatic inconsistency resolution: it can propose, for inconsistent annotations, the set of potential and available consistent resolution strategies, in which the user will choose the most relevant one.



List of ontology changes between versions:

V1: Palette-ver1-20070302.rdfs

V2: Palette-ver2-20070331.rdfs

| Source | Ontology changes | Param1 | Param2 | Param3 |
|---|---|---|---|---|
| 📖 | DeleteConcept | hasDeleteConcept: engineer-student | | |
| 📖 | CreateHierarchyConceptLink | hasSubConceptCreate: ITengineer-student | hasSuperConceptCreate: Student | |
| 📖 | CreatePropertyDomainLink | hasConceptDomainCreate: PhD_Student | hasPropertyDomainCreate: studyIn | |
| 📖 | MergeConcept | hasMergeConcept1: Master1_Student | hasMergeConcept2: Master2_Student | hasMergeConceptNew: Master_Student |

*Figure 19 Ontology Evolution Trace*

## 2.8. Conclusions

In this chapter, we have described the research work done in order to design the Knowledge Evolution Service envisioned for CoPs in the Palette project, a service aimed at supporting the evolution of two kinds of knowledge materialised in what we have called a “CoP Memory”. These two kinds of knowledge are: ontologies and semantic annotations.

A community memory evolving according to the evolution of the community itself, to help understand knowledge evolution better and to support it better, we have identified —using CoP evolution cycle models as analysis frameworks— (a) the events that can originate the evolution of the community knowledge and (b) the types of knowledge evolution that can result from (a).

Because the ontology may change to fit with the community's evolution cycle, we studied some ontology evolution issues related to the ontology evolution process and its consequences on annotation evolution.

Thus, we studied two evolution scenarios: with trace and without trace of ontology changes carried out during its evolution. These two scenarios often lead to inconsistencies of the annotations semantics using this modified ontology. For this reason, we have developed two approaches, a procedural approach and a rule-based approach, corresponding to these above scenarios, in order to manage semantic annotations evolution. These approaches allow us to detect inconsistent annotations and to guide the process of solving these inconsistencies.

These propositions are implemented in the CoSWEM prototype (Corporate Semantic Web Evolution Management) which facilitates the evolution management of the changes. It enables to carry out automatically or semi-automatically inconsistency detection and correction of the semantic annotations. Moreover, CoSWEM can highlight visually the different entities between two ontology versions according to the concept or property hierarchy.

The Corese search engine is also enhanced with some of our propositions. These consist of some of the inconsistency detection rules presented in this chapter. They enable to run the type checking rules on the ontologies or annotations loaded in Corese.

As further work, the rule-based approach will be refined and some effective algorithms on the process of correction and validation for semantic annotations changes will be studied.

# Chapter 3: Knowledge Evaluation Service

A Knowledge Evaluation Service is a service allowing to measure and assign a value to knowledge circulating within a CoP. This service is based on different criteria, qualitative and quantitative, identified through a model of knowledge evaluation within CoPs.

The concept of Communities of Practice (CoPs) has been quickly identified as a powerful social vector for organisational learning enhancement. Nevertheless, the evolution of this social phenomenon has faced during the last decade an exponential need of communication tools sustained by ICT development, leading to a "progressive virtualisation" of CoPs. Nowadays the use of various, complex Web 2.0. services represents a common feature for virtual CoPs members, and one arising issue is therefore the design of efficient knowledge management devices dedicated to such communities, able to support knowledge creation mechanisms *via* ICT solutions in order to enhance the creation and the development of collective, social knowledge.

. We argue in this research that KM services dedicated for virtual CoPs can be built on the basis of a reflection about the value of knowledge and information flows exchanged within virtual CoPs through collaboration technologies, by detecting and interpreting traces and signs about their use and their content. Hence, this work takes up the challenge of developing a comprehensive model and insights of suitable indicators for knowledge evaluation within virtual CoPs.
These indicators are elaborated on the basis of a model of knowledge value tracking within virtual CoPs using Web 2.0 solutions. This model of knowledge evaluation is built upon a study of participation and reification processes within virtual CoPs and outlines the conditions of knowledge value creation able to support the different virtual CoP's dimensions of knowledge creation, diffusion and storing.

This service could be used for different purposes; we can also imagine a lot of diverted uses for this service. Some use-cases are presented below.

Given the fact that the service allows to attribute a value to knowledge, and consequently to resources circulating in the CoP, we can propose to users some recommendations on the most valued knowledge inserted since their last connection. After an absence or inactivity period, the CoP's members are generally interested in the last most important resources concerning the CoP activity. The knowledge evaluation service could be used as an informant.

The service could be used as support of search engine. Indeed, KECES could be integrated to such services in order to sort results by value, and by this way could improve the results of the search engine and propose more pertinent results.

The service could also detect the obsolete knowledge, hence it will permit to maintain and update the CoP's database. Indeed the less valued knowledge, or below a specific threshold, could be deleted of the database.

Thanks to the knowledge evaluation service, the virtual CoP administrator will be able to monitor both participation and reification efficiency of its own CoP. Indeed, he will be able to detect exchange process issues such as a lack of interaction about specific topics or, on the other hand, the most popular ones. In a similar way, the administrator will be able to track inputs of reificated knowledge coming from members' experience. The possibility to monitor and detect popular topics or valuable knowledge as well as uninteresting topics or useless knowledge represent powerful triggers for virtual CoP's animation. Therefore, the evaluation service can represent a useful tool for virtual CoP's administrators who want to foster participation and reification processes.

## 3.1. Design of the Knowledge Value Model and Indicators

### 3.1.1. Assumptions about Knowledge and Learning within CoPs

Knowledge is a protean concept (tacit/explicit; individual/collective…) that has become prominent during these last years in the organisational learning literature (Nonaka and Takeuchi, 1995, Kogut and Zander, 1996). The realisation of knowledge as embedded and created from and through social relationships and interactions (Nahapiet and Ghoshal, 1996) has led some KM researchers to focus on the importance of communal resource (Von Krogh, 2003) and the notion of evolving communities within and without organisational boundaries. Resulting from a social and situated perspective of learning and cognition, the concept of CoP has been certainly one of the most developed and used ones. As (Amin and Roberts, 2006) noted, a large body of literature has developed concerning CoPs since the original use of the term (Lave and Wenger 1991).

CoPs facilitate an environment of "structured informality" supported by knowledge, knowledge owners (i.e. knowers), and CoPs infrastructure. CoPs own a vast base of knowledge ranging from theoretical concepts to practical experiences; they are the engines of learning for its members. Socially, CoPs are the fabrics of knowing as members of CoPs acquire communal identity around shared relationships, roles and ways of intermingling common knowledge, practices and approaches (Scarbrough et al., 1999). The importance of these practice and person-based networks has been acknowledged in a number of seminal works on: sense making (Weick, 1979), CoP (Lave and Wenger, 1991; Brown and Duguid, 1991), storytelling (Orr, 1990), knowing in practice (Cook and Brown, 1999), and communities of knowing (Boland and Tenkasi, 1995).

However, this social conception of situated learning and cognition has its own set of assumptions and focus (Wenger, 1998). From this perspective, we put forward some premises about the underlying conception of knowledge, knowing and knowers in the CoP concept.

- Individuals are social beings, and even if this fact appears as being trivial, it represents a central aspect of learning (Wenger, 1998).

- We must distinguish knowledge from knowing (Cook and Brown, 1999). The noun "knowledge" draws a static concept that implies knowledge as a thing that can be located and manipulated as an independent object or stock; it seems possible to "capture" knowledge, to distribute, measure and manage it.

- The gerund "knowing" suggests instead a process, the action of knowers being inseparable from them and from their context. If it may be possible to promote, motivate, nurture or guide knowing, the idea of capturing, distributing or even measuring it seems difficult, if not senseless… (Cohen, 1998);

The activity of learning must produce meaning, *i.e.* the (changing) ability of individuals to experience the world and their engagement (Wenger, 1998). Furthermore, from a socio-constructivist point of view, to learn means to participate to a process of co-construction of meaning (Vygotsky, 1978). In a CoP, knowledge and its articulation are social and context-related. Cognitive productions resulting from interactions between members of a CoP are not only attributed to individuals but also to the group itself (Dillenbourg et al., 1996).

### 3.1.2. Challenging Knowledge Value Measurement Issues

Even if during these last years knowledge has been widely recognised as a vital (if not the vital) source of competitive advantage and of production, both academic and practitioners seems to fail in developing acknowledged efficient methods for measuring knowledge. According to (Siesfeld,1998): "*Measuring knowledge is still a whole new area of development. It is clear that the traditional input/output approach to determining whether and to what extent a firm's assets are working do not work with knowledge*". Moreover, KM experiences show that good knowledge measures integrate qualitative and quantitative elements: "*Milestones and metrics define what you are trying to accomplish and whether you are succeeding, but 'crude and fuzzy' measures capture knowledge value more effectively than inappropriately precise ones*" (Cohen, 1998).

The nature of our issue of knowledge measurement leads us to adopt a socio-organisational view instead of an economy-level view. Hence, we focus our analysis on specific aspects of knowledge value. In the virtual CoPs framework, we can associate the "value" of knowledge with the proxy concept of "usefulness". Indeed, within such communities, knowledge generates value when it is used to satisfy a need; it represents here inputs for CoPs member's actions. So, the aim is not to determine the exact "objective" value of specific knowledge, but rather a "subjective", i.e. community-related value of knowledge within the CoP. For instance, in knowledge-intensive organisations such as CoPs, great importance is attached to the perceived value of knowledge by the community members (value of knowledge for individuals) as well as stored knowledge, as a collective good, element of the socially shared cognition (value of knowledge for the CoP). We then consider knowledge getting into the community (which implies clear representations of CoPs boundaries) that flows within the community and benefits for the CoP and/or its members.

As a consequence, we will focus our attention on a model able to underline indicators that provide information about a perception of the "value-added" by the knowledge of the CoP and its members (perceived outcome for members), instead of *ex post* or *ex ante* value indicators of knowledge. From this perspective, given our highly context-related, specific nature of knowledge and value, traditional input/output models of value measurement are not relevant. Both qualitative and quantitative indicators should be considered.

### 3.1.3. The Methodological Adjustments

Talking about measurement leads irremediably to consider performance measurement. If we have seen that developing an effective system for measuring and managing knowledge performance will require new ways of thinking, we cannot nevertheless ignore general properties of all measures. (Meyer and Gupta 1994) think that effective management requires multiple, uncorrelated and changing measures of performance. Applied to the virtual CoPs, this means that simple and static measures loose information contents over time – the knowledge useful today will not be so tomorrow, and unless the virtual CoPs change the measure, the value of knowledge is likely to decay. We notice five general properties related to effective measures (Meyer and Gupta, 1994).

i. Reliability: a reliable measure is one which returns the same value for the same performance, regardless the time of measurement, the form or nature of the observation (or observer), and the conditions under which these observations are made.

ii. Validity: a valid measure "measures what the measurer intends it to measure". For a measure to be valid, we need to be clear on what the objective of the measure is and what the assumptions about the relationship between the phenomenon and the measure are.

iii. Comparability: a single measure conveys little information in and on itself. The information comes when the single measure is compared to some other standard, like a base line. Providing information for comparison (if necessary) allows knowing whether a measured value is good or bad.

iv. Variability: a lack of variation among measurements makes it impossible to tell whether something is good or bad.

v. Time: performance measures tend to run down over time, through learning (homogenisation of human behavior and performance to maximise the measure), perverse learning (opportunistic appropriation of the measure in order to maximise it, but with diminishing performance) and selection (if over time individuals who perform well are retained and others are not, then the measure will no longer convey any new information as the pool grows in homogeneity).

Of course, measures should not be frozen. When a measure does run down, it has to be replaced by another; as well as the more the phenomena we study are complex, the more measures we need. From this perspective, as every virtual CoP is different (in its nature, its focus, its used ICT solutions or even its level of maturity), it is therefore both difficult and inappropriate to propose a predefined battery of indicators for measuring knowledge within virtual CoPs, such as in a toolkit for instance.

We propose instead a model of knowledge value tracking within virtual CoPs and insights about criteria to be taken into account for the elaboration of suitable indicators in accordance with the specific nature and properties of the considered virtual CoP.

Therefore this model (see Figure 18 in 3.1.4.) is composed by several theoretical propositions of knowledge value tracking within virtual CoPs that represent the framework and the limits from which the measure will deal with. For each theoretical proposition, corresponding objectives are defined in order to give sense to the measure.

Then we define relevant criteria according to every objective. These criteria might allow, by observing their evolution, an interpretation of the current situation according to the objectives. These criteria will be also broken down in several parameters that provide qualitative and quantitative measures that strongly influence the relevance of the exploitation of the measure.

Then the parameters of the chosen criteria must be transcribed in data, which can be combined in order to provide a global indicator.

However, considering the idiosyncratic nature of every virtual CoP and the complexity of knowledge evaluation, we present in this research some basic, generic criteria, that might be completed in the case of empirical application of the model according to specificities of the studied virtual CoP. Following up research carried by Barlatier et al. (2007), we describe this model as well as insights about main parameters and resulting indicators in the following section.

### 3.1.4. Towards a Knowledge Value Model for Virtual CoPs

This section describe the knowledge value tracking model dedicated for virtual CoPs, articulated through several theoretical propositions. As mentioned before, the complexity of this task implies both a deep reconsideration of the CoP's basic concepts as well as some appropriated methodological adjustments, presented in the previous section From this perspective, we explain here the theoretical layout that will guide our analysis, based on a study of participation and reification processes within virtual CoPs.

Commitment, participation and exchanges are important concepts intervening in a CoP. They occur in face-to-face meetings, but are also supported by ICT solutions. Nowadays people exchange a lot of information by emails or via forums, using a lot of different means to communicate, and consequently participating in the CoP's life.

Considering our objective of giving elements for measuring CoPs knowledge value supported by ICT tools, we will use the term "knowledge" as an umbrella term gathering explicit knowledge and information. For virtual CoPs, inputs of knowledge are pieces of explicit knowledge and information (able to circulate via ICT solutions), brought by CoPs members from CoPs environment *via* different exchange objects as tools, rules, methodological support, demonstrations, references and vignettes or cases (Daele, 2006). Then, the CoP will act as a system, *i.e.* as a method for collecting and processing knowledge inputs, specific to each CoP, and as a consequence, giving different results for different CoPs.

Hence, we consider CoPs as self-organised, autonomous systems, with strong identities, creating their own values and references system as well as their own sense making. In other terms, CoPs are autopoietic systems (Varela, 1989).

However, despite of the CoPs' autopoietic characteristics, virtual CoPs members need to use ICT solutions to capture external information in order to develop their individual and collective knowledge. From this perspective, Web-based services such as web specialised search engines or RSS feeds represent efficient means of linking and collating disparate information sources and data streams, according to chosen topics. Virtual CoPs members should also propose knowledge sources such as documents to develop the CoP's knowledge base.

The main criteria that should be considered here in order to define knowledge value indicators about knowledge incomes are: (i) the reputation of the knowledge provider that shows the viability of the source; (ii) the reputation of the author; and (iii) the relevance of knowledge regarding the topics of the CoP.

> *Proposition 1: CoPs are autopoietic, self-referencing systems. CoPs members provide inputs of knowledge to the community. These inputs are required to perform a task, to answer a need and to effect a change in members' daily activities.*

The primary focus of the CoPs conception is on learning as social participation (Wenger, 1998). Participation represents in Wenger's conception of CoPs a core element since it is through participation that communities' characteristics and practice are developed: "*Participation here does not just refer to local events of engagement in certain activities with certain people, but to a more encompassing process of being active participants in the practices of social communities and constructing identities in relation to these communities*" (Wenger, 1998).

Participation is an active process that conveys the possibility to mutual recognition and the ability to negotiate meanings, but does not necessarily entail equality or respect, or even collaboration (Wenger, 1998). If CoP's members have repeated exchanges about knowledge freely flowing within the community, we can consider that the most collective exchanges a piece of knowledge generates, the more potential value it has. If knowledge cannot be measured, its impact can be. Indeed, knowledge lies here in the flows, and it is in these flows, *i.e.* in the combination of community member's experiences and insights, that knowledge is created and applied (Siesfeld, 1998).

Thus, it is logically through this participation process within virtual CoPs that collaboration technologies impact information and knowledge exchanges. The design of adequate solutions of collaboration technologies represents the core, the key factor of success of any virtual CoP. Web 2.0 solutions will often provide several different communication and collaboration spaces for supporting both tacit and explicit knowledge transfer, shared and individual activities, synchronous (real-time chat…) and asynchronous (e-mailing…) communication. For instance, collaboration technologies can support tacit knowledge transfer by offering the possibility to convey an unbound number of discourse types such as ideas, comments and notes, *i.e.* the personal interactions required for its sharing (e.g. videoconferencing).

Such technologies should also ensure the possibility of managing (*i.e.* adding, deleting, updating…) exchange objects in order to foster interactions within virtual CoPs. The main criteria that should be considered here in order to define knowledge value indicators about participation and knowledge exchanges are linked with the quality and the quantity of the exchange objects (*i.e.* length and frequency of reactions/consultations on objects; number of downloads, exchanges of an object; notation and comments about an object…). A smart combination of such indicators according to the collaboration technologies used within the virtual CoP must provide information about dynamics of knowledge exchange among its members. In addition, the use of such technologies often requires user registration, identification and authorisation. These functionalities imply *de facto* a reflection about the boundaries of the virtual CoP, *i.e.* its identity. Indeed, to which extent are guest members and lurkers allowed to get into the virtual CoP? As emphasised by Wenger (1998), to participate in a CoP implies an *action* of participation within this CoP. Therefore, such issues about virtual CoPs boundaries must be also considered in the study of its participation dynamics.

> *Proposition 2: High levels of knowledge exchanges and interactions within the CoP strengthen the participation process and reveals knowledge with high potential value.*

Within a virtual CoP, it is quite easy to determine levels of interactions (number of emails exchanged around a subject, number of clicks on a link…); nevertheless, it is more complex to determine their interest. Indeed, people can interact around knowledge with low value, in order to demonstrate that it is not interesting or not proved. By contrast, high value knowledge that could be very interesting for CoPs members can be overlooked due to the important number of information contained in such tools (lots of topics in forums, to many emails exchanged with not enough time to read them…). Anyway, thanks to these interactions, CoPs members may be able to anticipate the created value by the use of this knowledge, integrating and combining it (Kogut and Zander, 1996) in order to mobilise it in a personal knowing process.

From this perspective, the use of web 2.0 collaborative solutions can be a lever for both, explicit and implicit decision-support making, for a single or a group of members of the virtual CoP. This support to decision-making means that users can retrieve information about past events or decisions, documents or objects linked to this decision (or even topic) with suitable visualisations devices. For instance, in the case of a collective decision-making process, voting algorithms can be considered.

Hence, the main criteria that should be considered here in order to define knowledge value indicators about the potential value of information / knowledge are directly linked with the design of participation and interactions of knowledge objects indicators (see P2 above), focusing on receptivity and feedbacks of virtual CoPs members.

> *Proposition 3: The potential value of knowledge circulating within CoPs depends on interaction levels, but also simultaneously on the members' ability to assimilate it and to anticipate the created value by its use.*

After having appreciated the potential value of knowledge, it is now relevant to examine how this potential value can be achieved. Knowledge is not separable from its context, especially within CoPs which origin is rooted, let us not forget, to situated learning (Lave and Wenger, 1991; Wenger, 1998). Knowledge is here a lever for action, and its value is very context-dependant. In addition, CoP's members use CoP's knowledge in the framework of their practice. Therefore, this process of knowing is a human act.

From this perspective, using CoP's knowledge refers to the personal knowledge-creation abilities of the CoP's member (*i.e.* his abilities to detect, assimilate, combine and experiment this knowledge). As (McDermott, 1999) wrote "*… professionals piece information together, reflect on their experience, generate insights, and use those insights to solve problems*".

Collaboration technologies should therefore support virtual CoPs members to achieve outstanding results in their everyday practice. Anyway, the technology can be used in virtual CoPs' individual and collective decision-making processes, insofar as such collaborative functionalities are compatible with the tools used by CoPs members in their daily practice. The more these solutions are closely linked with CoPs members' action devices, the more they can enhance knowing capabilities.

Hence, the definition of knowledge value indicators about the use of collective knowledge in everyday, individual practice implies the consideration of individual, personal social characteristics and interests that stick out our framework of analysis. However, technologies linked to 3G telephony or Blackberry devices for instance, as easy-to-carry technological devices, can develop user-proximity, community-oriented facilities that can be explored.

> *Proposition 4: The value of CoP's knowledge in practice relies on the "knowing" capabilities of CoP's members, i.e. their personal abilities to use knowledge in their daily practice.*

The (personal) use of knowledge circulating within CoPs would be valueless for the community if members do not share and exchange it. These outcomes of knowledge in motion have to be "crystallised" by CoPs members and re-injected in the community in order to be shared, evaluated and acknowledged by the whole CoP. This refers to the concept of "reification" defined by Wenger as: "*the process of giving form to our experience by producing objects that congeal this experience into 'thingness'*" (Wenger, 1998). According to this, applied knowledge generates value if virtual CoPs members formalise their experiences, *i.e.* give a form to their own understanding of their practice by writing and exchanging emails and messages, or producing electronic documents and books.

Hence, virtual CoPs members produce exchange objects, shaped by their experiences. But, as Wenger emphasised: "*these objects… are only the tip of an iceberg, which indicates larger contexts of significance realised in human practices*" (Wenger, 1998). Once produced, these objects can be introduced to the virtual CoP by different ways: either directly to some other virtual CoPs members or put in the virtual CoP electronic document memory, *i.e.* the virtual CoP knowledge base. Nevertheless, these objects represent as many points of focus around which the negotiation of meaning becomes organised (Wenger, 1998). In most cases, less-formalised objects are directly submitted to other members, and then the negotiation of meaning process will be collective and often achieve the articulation of the object. But virtual CoP members can place more formalised objects directly in the virtual CoP knowledge base. In this case, the collective negotiation process is rather focused on the pertinence of the existence of this document within the virtual CoP's knowledge base instead of the collective achievement of its formalisation. If this newly re-injected knowledge generates interesting interactions within the virtual CoP, it will then generate value for the whole community itself.

We propose to label these outcomes of the virtual CoP's reification process "realised value" of knowledge, *i.e.* value from knowledge experience feedback., in contrast with the potential value of knowledge previously defined. However, we must emphasise that in the framework of this research we make a distinction between reification of knowledge from direct or indirect experience feedback. In other words, we distinguish "*reification by interacting*", which refers to the reification of knowledge by perceiving, interpreting, reacting from CoP's members previous experience (*i.e.* related to participation in P2) and "*reification by doing*", which refers to the reification of knowledge by directly using, making, reusing knowledge exchanged within the framework of the CoP (*i.e.* related to direct experience feeback).

From this perspective, collaboration technologies should support experience feedback and production of meaning for virtual CoPs members. This means that collaborative solutions may allow of course the possibility to share experience, personal standpoints and documents in collaboration spaces' topics but also require both individual expertise management (linked with the user's profiles and the identification of experts) and collective expertise management (linked with collegial agreement and acknowledgement) for the validation of the reificated objects.

Hence, the main criteria that should be considered here in order to define knowledge value indicators about the reification of knowledge are both function of the capabilities of CoP's members to use knowledge in practice (see P4 above) as well as their capabilities of reification, *i.e.* of formalisation of their own experience. Such indicators can be built according to the implementation of, for instance, ROE (return of experience) annotations or links on comments about used knowledge. Virtual CoPs members may inter-relate new reificated knowledge objects with older and / or used ones, still available in the virtual CoP's knowledge base.

> *Proposition 5: The virtual CoPs member's capabilities of reifying outcomes of knowledge in motion and of diffusing them within the community generate value for a virtual CoP.*

Anyway, the reification of "realised" knowledge leads virtual CoPs members to use collective knowledge storing ICT solutions, such as a shared database, in order to make it available to other members. Afterwards, these objects of knowledge are submitted to the judgment of the other virtual CoP members, which validate or not the considered object. Once validated, knowledge can be stored and accessible to the virtual CoP. In order to be an efficient ICT solution, the knowledge base must be organised and indexed so as to be convenient to usual requests as well as specific demands. In addition, the base must propose links between tasks and roles to pertinent documents or knowledge objects. This structures the presentation and storing of knowledge to virtual CoPs members.

Moreover, the accumulation of the same knowledge yields no extra value (Siesfeld, 1998). Indeed, if there is value in reproducing knowledge, there is no value in acquiring the same knowledge again: "*More is not better, new is better*" (Siesfeld, 1998). Knowledge value may reside more in trying to discover relationships among distinctive ideas, via argumentation and negotiation of points of view, than in embracing sameness (Cohen, 1998).

From this perspective, collaboration technologies dedicated to virtual CoPs exchanges should be compatible with the collective knowledge storing solution, or at least allow the retrieving of knowledge objects (*i.e.* documents, notes, ideas, comments…) and their reuse in different collaboration spaces (according to the access rights). The virtual CoP knowledge base, as a knowledge repository, must structure and present knowledge efficiently, allow an easy access to CoPs members and avoid proposing accumulation of the same knowledge. The virtual CoP's knowledge base may be dynamic and updated in order to prevent the virtual CoP from inertia. It may be accessed in multiple ways and its content combined, restructured, and presented in a variety of new contexts depending on how the knowledge base has been designed and the mechanisms for presenting and distributing its content (Zack and Serino, 2000).

Hence, the different criteria that should be considered here in order to define knowledge value indicators about the management of the virtual CoP's knowledge base are related to the availability and accessibility of knowledge objects as well as to the coherence of the knowledge base (*i.e.* interactions and cumulativeness of knowledge objects). Indicators based on searching options need to be implemented (keywords, topics, knowledge object, combination of them and links between them…) as well as clear indications about most valuable and / or popular content. A “search the Web” option can also be implemented but should be linked with P1 related functionalities and indicators in order to add value (in targeting specific search engines to expand research, for instance…).

> *Proposition 6: The virtual CoP’s knowledge base, as a knowledge repository, must structure and present knowledge efficiently, allow an easy access to virtual CoPs members and avoid proposing accumulation of the same knowledge.*

Once knowledge has been reified and proposed to the virtual CoP, members exchange, share their experiences and debate about it. If knowledge is acknowledged as useful for the group, it is henceforth implemented in the virtual CoP knowledge base. The process of negotiation of meaning will collectively evaluate, validate and attribute categories to the stored knowledge. This collective process will also update the virtual CoP knowledge base.

However, usefulness is difficult to evaluate. Some documents, e.g. a basic process, may be very useful for a novice member and have less value for an expert. Moreover, for a virtual CoP gathering members from different organisations, some knowledge may also be evaluated as very useful for one, and have less value for another. In this context, usefulness refers to the subjective value of knowledge. It depends on the potential use of the stored knowledge object (Marwick, 2001). Thus, great importance is dedicated to stored knowledge that generates high levels of interaction and experiences accumulation within the virtual CoP.

Evaluating the usefulness could be done after having described the different groups of members composing the virtual CoP: novice versus expert, intra-organisation versus inter-organisation, etc. Sometimes, virtual CoP’s identified sub-groups can evaluate the usefulness of a virtual CoP’s knowledge. As knowledge captured by a CoP is an element of the collective construction, linked to exactly defined social situations, it is normal that this knowledge and its usefulness evolve with the continuous collective interactions.

Furthermore, knowledge is a specific resource that has a specific life cycle and degree of obsolescence. Actually, knowledge can have a great value at a certain time, and can drop to zero if this stock of knowledge becomes obsolete. This means that, as the timing of obsolescence is highly uncertain, there are no schedules of depreciation. In this case, a maintenance service could be useful to sort knowledge contained in mails for instance, or to sort the old posts or documents contained in a forum.

From this perspective, virtual CoPs members are both users and suppliers of the knowledge base that consequently requires high degrees of structuring and viewing flexibility appropriated to various, particular types of content (*i.e.* knowledge and information).

Hence, the main criteria that should be considered here in order to define knowledge value indicators about knowledge storing are related to the validation and the modification of knowledge objects. Both quantitative and qualitative indicators may be developed in this context. For instance, metrics and measurement instruments’ regarding number of downloads, reactions, comments, about knowledge objects should be jointly developed with positive or negative feedback and notation about these objects. It is not obvious that the most downloaded documents are the more valuable ones.

> *Proposition 7: The virtual CoP, through a collective process of negotiation of meaning, evaluates, validates and attributes categories to the stored knowledge.*

*Hence, the knowledge base may be dynamic and updated in order to prevent the virtual CoP from inertia.*

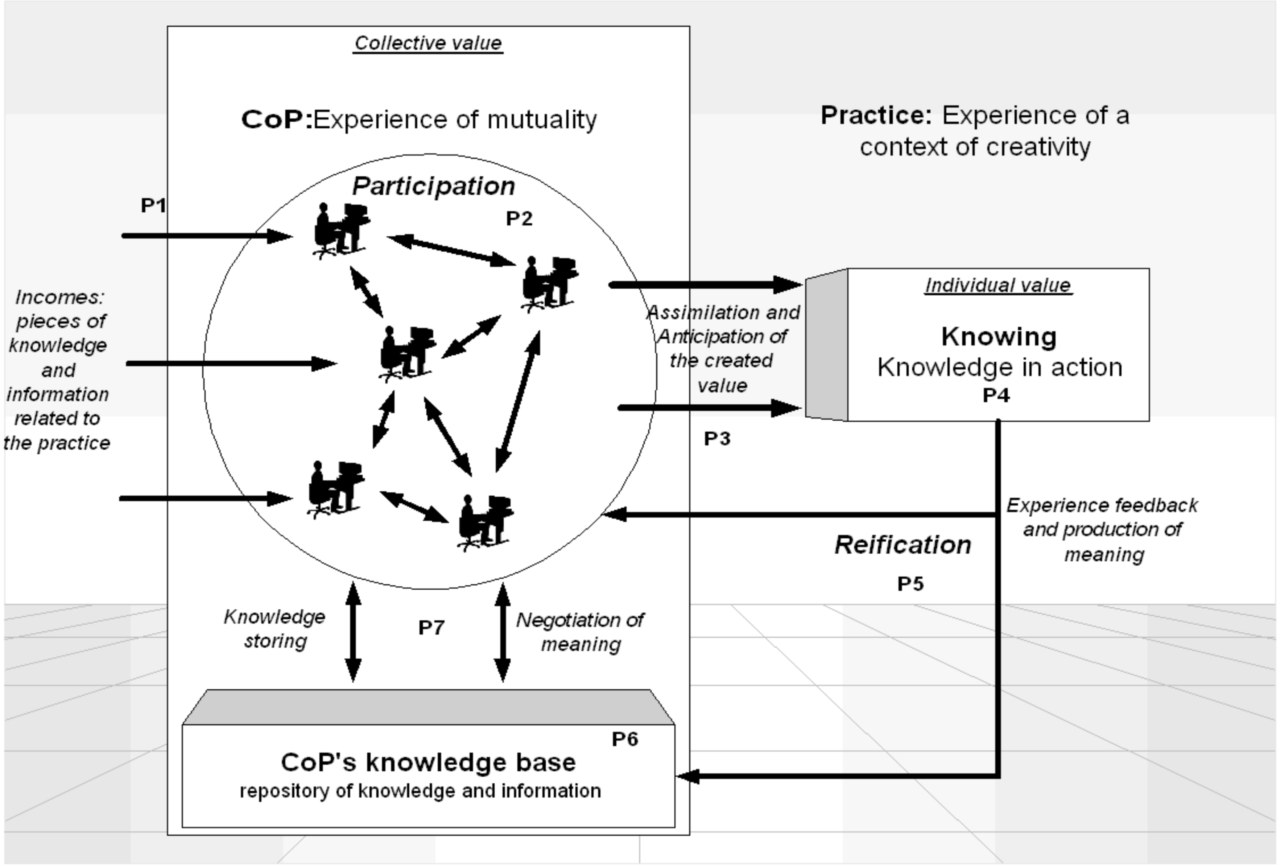

Collective value
CoP: Experience of mutuality
Participation
P1
P2
Incomes: pieces of knowledge and information related to the practice
Practice: Experience of a context of creativity
Assimilation and Anticipation of the created value
P3
Individual value
Knowing
Knowledge in action
P4
Experience feedback and production of meaning
Reification
P5
Knowledge storing
P7
Negotiation of meaning
P6
CoP's knowledge base
repository of knowledge and information

*Figure 20. The Knowledge Value Model*

The above Figure 20 synthesises our theoretical construction through a model of knowledge value tracking within virtual CoPs. This model reveals insights about knowledge evaluation within virtual CoPs through the analysis of the participation / reification dialectic. The comprehension of this participation / reification duality appears as the key to analysing knowledge value creation within CoPs. Moreover, participation and reification are self-feeding processes (Wenger, 1998). Indeed, participation implies interactions, identifiable commitment in CoPs activities that leverage actions in CoPs' members practice. Then, reification gives form to theses actions, and generates interactions within CoPs through mainly the negotiation of meaning processes.

Hence, reification strengthens commitment and participation within CoPs, with the negotiation of meaning as catalyst. As the dual system participation / reification is relatively less explored in the literature, our quest for reliable measures of knowledge value within virtual CoPs identify and analyse knowledge value creation vectors within this system. From this perspective, this model allows the identification of pertinent knowledge measurement indicators, but we point out once again that such indicators should be implemented with regard to the used (both actual and future) ICT solutions within the virtual CoP, as well as to its own nature, objectives and maturity. According to these issues, knowledge value indicators related to parameters described in the theoretical propositions of the previous model can be developed. However, we can notice that the idiosyncratic and complex nature of every CoP may lead to the construction of specific, idiosyncratic indicators as well.

## 3.2. Knowledge Value Processing

The knowledge value-tracking model depicted above (see Figure 20), designed through the articulation of several theoretical propositions (P1 to P7), has given insights about generic criteria to be considered. As mentioned in 3.1.3., we now focus on relevant parameters and indicators for every generic criterion.

From this perspective, the Figure 21 below shows the diagram of the set of proposed criteria, as well as their related parameters and indicators. Hence, we propose in this section to develop a mathematical framework for knowledge value processing.

As explained above in the model presentation and explanation on the proposals, the proposition 4 is not included in our processing. Indeed this proposition includes factors intervening in the member daily practice, *i.e.* outside the community.

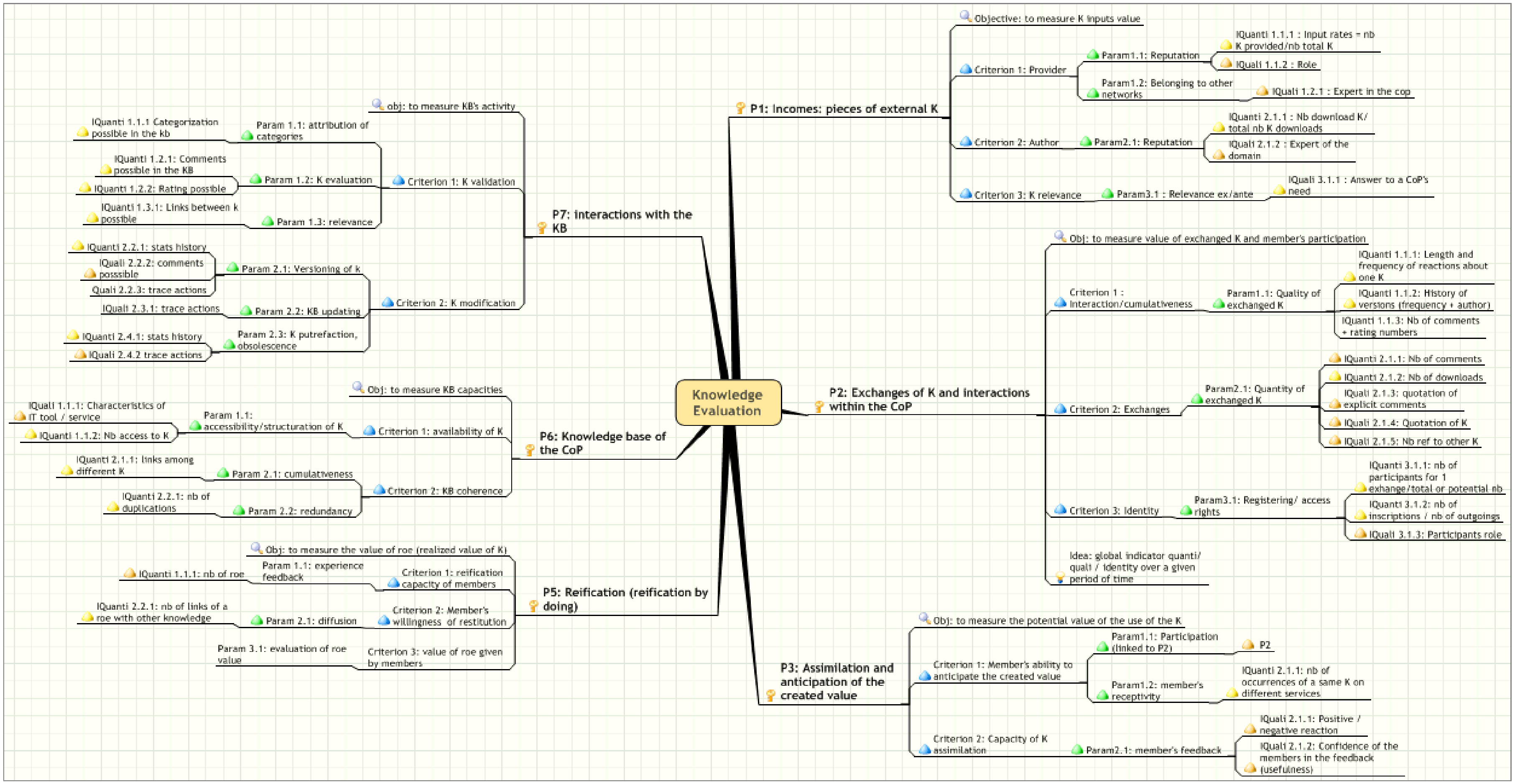

Knowledge Evaluation
P1: Incomes: pieces of external K
Objective: to measure K inputs value
Criterion 1: Provider
Param1.1: Reputation
IQuanti 1.1.1 : Input rates = nb K provided/nb total K
IQuali 1.1.2 : Role
Param1.2: Belonging to other networks
IQuali 1.2.1 : Expert in the cop
Criterion 2: Author
Param2.1: Reputation
IQuanti 2.1.1 : Nb download K/ total nb K downloads
IQuali 2.1.2 : Expert of the domain
Criterion 3: K relevance
Param3.1 : Relevance ex/ante
IQuali 3.1.1 : Answer to a CoP's need
P2: Exchanges of K and interactions within the CoP
Obj: to measure value of exchanged K and member's participation
Criterion 1 : Interaction/cumulativeness
Param1.1: Quality of exchanged K
IQuanti 1.1.1: Length and frequency of reactions about one K
IQuanti 1.1.2: History of versions (frequency + author)
IQuanti 1.1.3: Nb of comments + rating numbers
Criterion 2: Exchanges
Param2.1: Quantity of exchanged K
IQuanti 2.1.1: Nb of comments
IQuanti 2.1.2: Nb of downloads
IQuali 2.1.3: quotation of explicit comments
IQuali 2.1.4: Quotation of K
IQuali 2.1.5: Nb ref to other K
Criterion 3: Identity
Param3.1: Registering/ access rights
IQuanti 3.1.1: nb of participants for 1 exhange/total or potential nb
IQuanti 3.1.2: nb of inscriptions / nb of outgoings
IQuali 3.1.3: Participants role
Idea: global indicator quanti/ quali / identity over a given period of time
P3: Assimilation and anticipation of the created value
Obj: to measure the potential value of the use of the K
Criterion 1: Member's ability to anticipate the created value
Param1.1: Participation (linked to P2)
P2
Param1.2: member's receptivity
IQuanti 2.1.1: nb of occurrences of a same K on different services
Criterion 2: Capacity of K assimilation
Param2.1: member's feedback
IQuali 2.1.1: Positive / negative reaction
IQuali 2.1.2: Confidence of the members in the feedback (usefulness)
P5: Reification (reification by doing)
Obj: to measure the value of roe (realized value of K)
Criterion 1: reification capacity of members
Param 1.1: experience feedback
IQuanti 1.1.1: nb of roe
Criterion 2: Member's willingness of restitution
Param 2.1: diffusion
IQuanti 2.2.1: nb of links of a roe with other knowledge
Criterion 3: value of roe given by members
Param 3.1: evaluation of roe value
P6: Knowledge base of the CoP
Obj: to measure KB capacities
Criterion 1: availability of K
Param 1.1: accessibility/structuration of K
IQuali 1.1.1: Characteristics of IT tool / service
IQuanti 1.1.2: Nb access to K
Criterion 2: KB coherence
Param 2.1: cumulativeness
IQuanti 2.1.1: links among different K
Param 2.2: redundancy
IQuanti 2.2.1: nb of duplications
P7: interactions with the KB
obj: to measure KB's activity
Criterion 1: K validation
Param 1.1: attribution of categories
IQuanti 1.1.1 Categorization possible in the kb
Param 1.2: K evaluation
IQuanti 1.2.1: Comments possible in the KB
IQuanti 1.2.2: Rating possible
Param 1.3: relevance
IQuanti 1.3.1: Links between k possible
Criterion 2: K modification
Param 2.1: Versioning of k
IQuanti 2.2.1: stats history
IQuali 2.2.2: comments posssible
Quali 2.2.3: trace actions
Param 2.2: KB updating
IQuali 2.3.1: trace actions
Param 2.3: K putrefaction, obsolescence
IQuanti 2.4.1: stats history
IQuali 2.4.2 trace actions

*Figure 21. The Knowledge Value Diagram*

Logically, Figure 21 speaks about knowledge. From a technical point of view, knowledge in a CoP refers to any document or information manipulated by the members and stored in the knowledge base of the community. It comprises also explicit (*i.e.* formalised) comments, ratings, return on experience; in any electronic form and from any tool, comprising e-mails, forum posts, etc.

Let $m \in M$ be a CoP's member of the set $M$ of members. Let $k \in K$ be a piece of knowledge of the set $K$ stored in the knowledge base (KB) of the CoP. The value of $k$ in terms of a perceived outcome for the CoP's members can be assimilated to the realised value of $k$, as defined in the previous section. It is obtained by two different values, the first being the potential value it has at the moment it enters the CoP, $val_{pot}(k)$, and the second being the value in term of reification, which we note $val_{reif}(k)$. The latter is calculated according to the realised value formula, which then can be written:

$$\boxed{val(k) = f(val_{pot}(k), val_{reif}(k)) = f(val_{pot}(k), g(val(K_{reif}(k))))},$$

where $K_{reif}(k)$ represents any knowledge reified from $k$.

According to our knowledge evaluation model, seven propositions have been enunciated in the previous section, all linked to important steps in the knowledge life-cycle in the CoP. Unless a piece of knowledge $k$ has been assimilated by the CoP (*i.e.* validated and stored in the KB), its value corresponds to its potential value, which is a function of the values corresponding to propositions P1, P2 and P3. Then, the final value of $k$ as a realised value is its potential value, plus the value of reifications it has generated, corresponding to proposition P5. We write:

$$\boxed{\begin{array}{l} val_{pot}(k) = \mathrm{f}_1(val_{P1}, val_{P2}, val_{P3}) \\ val_{reif}(k) = val_{P5} \end{array}},$$

with $val_{Px}$ being the value corresponding to the proposition $Px$. It is to note that the value corresponding to the proposition P4 does not appear here as it is hard to process and is roughly included in $val_{P5}$. Propositions P6 and P7 can be treated separately, as they concern the KB. The value derived from these propositions influences the value of knowledge. However, whether it should be added in the calculus of the knowledge value or not can be discussed. Indeed, the influence of the KB characteristics will already implicitly influence most of the indicators used to obtain the different $val_{Px}$. For example, a KB having a bad design or difficult to use comments/ratings system will lead to a low-level number of comments/ratings, which influences the value of knowledge ($val_{P2}$). Finally we thus consider separately a value for the KB, knowing it influences the value of knowledge, but we let the formalisation of the link between both for a further research. We define the value of the CoP's KB as:

$$\boxed{val(KB) = \mathrm{f}_3(val_{P6}, val_{P7})}.$$

According to the model, each proposition value $val_{Px}$ is a function of $n_x$ criteria $c_{x.i}$, which are themselves functions of $m$ parameters $par_{x.i.j}$:

$$val_{Px}(k) = \mathrm{fc}_x(\mathbf{c}_x(k)),$$

where $\mathbf{c}_x=[c_{x.1}, \ldots, c_{x.n}]^t$ is a vector of criteria:

$$c_{x.i}(k) = \mathrm{fp}_{x.i}(\mathbf{par}_{x.i}(k)),$$

where $\mathbf{par}_{x.i}=[par_{x.i.1}, \ldots, par_{x.i.m}]^t$ is a vector of parameters, which can be calculated according to some indicators $ind_{x.i.j}$:

$$par_{x.i.j}(k) = \mathrm{fi}_{x.i.j}(\mathbf{ind}_{x.i.j}(k)),$$

with $\mathbf{ind}_{x.i.j}=[ind_{x.i.j.1}, \ldots, ind_{x.i.j.l}]^t$.

Before going into the details of each function, we define some generic functions that will be reused multiple times:

- When $n$ variables are assumed to be independent, their influence can be calculated by a weighted sum. We note wS() this function, defined as:

$$\boxed{\mathrm{wS}(\mathrm{f}_1,\ldots,\mathrm{f}_n) = \frac{\sum_{i=1}^{n} \mathrm{sgn}(\alpha_i).|\alpha_i|.\mathrm{f}_i}{\sum_{i=1}^{n} |\alpha_i|}},$$

  where $\alpha_i$ , with $|\alpha_i| \in [0,1]$, is a weight that can be either positive or negative depending on $\mathrm{f}_i$ and the influence it has in the global function wS.

- The relative value of a function f($x$), where $x$ is an element of a set $X$, is the value of f($x$) related to the values obtained for the other elements of the set $X$. We write:

$$\boxed{\mathrm{f}^{rel}(x) = \%(\mathrm{f}(x), \mathrm{f}(X))},$$

  where the %() represents a comparison function, which can have different form according to how the distinction between $x$ and the other elements of $X$ needs to be made. In order to homogenise the different values (e.g. given by criteria or their parameters) that will need to be combined (in a wS function for example), it seems clever to use the so-called z-value, which centers and reduce the variables. This gives:

$$\boxed{\mathrm{f}^{rel}(x) = \frac{\mathrm{f}(x) - \overline{\mathrm{f}(x)}}{\sigma_{\mathrm{f}(x)}}},$$

  where $\overline{\mathrm{f}(x)} = \underset{x}{\mathrm{E}}\{\mathrm{f}(x)\}$ is the average of f($x$) other the set $X$, and $\sigma_{\mathrm{f}(x)}$ denotes the standard deviation.

- Last, we define the average of a function f($x$) other the set $X$ of all $x$:

$$\boxed{\underset{x}{\mathrm{E}}\{\mathrm{f}(x)\} = \underset{x \in X}{\mathrm{E}}\{\mathrm{f}(x)\} = \frac{1}{\dim(X)} \sum_{i=1}^{i=\dim(X)} f(x_i)}.$$

### 3.2.1. Potential Value of Knowledge

The potential value is obtained by considering propositions 1, 2 and 3. The first two concern an inputted knowledge, while the last one is related to capacities of the set of members in the CoP. Considering P1 and P2 are independent, and both influenced by P3, we write:

$$\boxed{val_{pot}(k) = \alpha.val_{P3}.\mathrm{wS}(val_{P1}(k), val_{P2}(k))},$$

where $\alpha \in [0,1]$ is a weighting coefficient that can be used to tune the influence of $val_{P3}$, and all the $\alpha_i$ in wS are positive. Since P3 influences P1 and P2, we choose to multiply P1 and P2 values by P3. We detail in the following the formula for each proposition value involved here. In order to get a value that is meaningful for CoP's members, we standardise on the [0,1] interval, which finally gives:

$$\boxed{val_{pot}(k) = \frac{val_{pot}^{nstd}(k) - \min\{val_{pot}^{nstd}(k)\}}{\max\{val_{pot}^{nstd}(k)\} - \min\{val_{pot}^{nstd}(k)\}}},$$

where $val_{pot}^{nstd}(k) = \alpha.val_{P3}.\mathrm{wS}(val_{P1}(k), val_{P2}(k))$ is the non-standardised potential value. Note that the different values calculated from the propositions, as defined hereafter, have values that have been homogenised and thus can be negative or positive. They should also be standardised between 0 and 1 if they are to be given directly to CoP's members for some purpose.

### P1: Incomes - pieces of knowledge and information

**Objective:** measure knowledge and information input in the CoP.
As it would not be in the interest of the CoP, we neglect any influence link that could exist between a knowledge provider and an author (e.g. a provider almost always provides knowledge coming from the same set of authors). With this postulate, the three criteria defined for measuring proposition 1 value are considered as being independent. We can write:

$$\boxed{val_{P1}(k) = \mathrm{wS}\left(c_{1.1}(k), c_{1.2}(k), c_{1.3}{}^{rel}(k)\right)},$$

where $c_{1.1}$ is the value related to the provider of $k$, $c_{1.2}$ is the value related to the author of $k$ and $c_{1.3}$ is the value related to the pertinence of $k$ regarding the CoP. Here, all the $\alpha_i$ in wS are positive.

## *Criterion 1: knowledge provider*

$c_{1.1}$ is obtained from 2 parameters: the provider's $m$ reputation in the CoP (*repuP*($m$)), and his belonging to other communities (*connect*($m$)). If we neglect the influence the reputation in the CoP can have in the other communities he is member of and vice-versa, we can write:

$$\boxed{c_{1.1}(k) = \mathrm{wS}\left(repuP^{rel}(m), connect^{rel}(m)\right)}, \text{ with } m \text{ provider of } k,$$

where all the $\alpha_i$ in wS are positive. If we make the assumption that the reputation is more important than the fact of belonging to other communities, we have $\alpha_1 > \alpha_2$. *repuP*$^{rel}$($m$) and *connect*$^{rel}$($m$) are respectively the reputation and connection functions for $m$ related to the set of members $M$.

- The reputation of the knowledge provider is indicated by his knowledge input rate, *inKrate*, balanced by a coefficient associated to his role in the CoP at the time the knowledge is provided, $\lambda_R \in [0,1]$, where $R$ is a role. The reputation of a member $m$ as a knowledge provider is then calculated by the sum of knowledge provided under each of his roles balanced by a role coefficient:

  $$\boxed{repuP(m) = \sum_{R_i} \lambda_{R_i}(m).inKrate_{R_i}(m)},$$

  with $inKrate_{R_i}(m) = nb(inputed\_k, m \mid role(m) = R_i)$

  Note that since the role is a qualitative indicator, there should be a hierarchy in roles allowing affecting a value or a range in the [0,1] interval to each role (the highest value being of course 1).
  Note also that a member can have a unique role within a CoP's tool, even if he has multiple roles within a CoP. Indeed, a member can have a unique tool's role which sums the roles he has on a CoP (which becomes rights within the tool). In this case, there might be no way to distinguish the different roles a member has.
- The good reputation of a provider tends to increase the potential value of his provided knowledge. The influence of membership in other networks is indicated by the reputation of the knowledge provider in these networks. We take the mean value over all communities $co_i \in Co(m)$ to which $m$ belongs:

$$\boxed{connect(m) = \underset{co_i}{\mathrm{E}}\{repuP(m, co_i)\}}.$$

Note that though reputation is a qualitative indicator, it can be estimated with the formula we have defined just before.

### *Criterion 2: knowledge author*

$c_{1.2}$ is obtained from one parameter, which is the reputation of the knowledge's author in the CoP. This reputation can be indicated globally by the download rate of knowledge he is author of (*dlrate*), to which we add a coefficient $\varepsilon_{Dk} \in [0,1]$ formalising the recognised expertise of the author in the domain of *k*. We consider the expertise in *k*'s domain has no influence on the download rate, though the inverse is probably true. An author can have a high download rate, without necessarily being considered as an expert. Consequently, we write:

$$\boxed{c_{1.2}(k) = \mathrm{wS}\left(\varepsilon_{D\,k}(m), dlrate^{rel}(m)\right)},$$

where all the $\alpha_i$ in wS are positive, and $dlrate(m) = \sum_{k_i \in K} dlrate(k_i) \mid m \text{ Author of } k_i$,

with $dlrate(k_i)=nb\_downloads(k_i)$, being the number of downloads of a particular $k_i$ since it has been entered in the CoP. The recognised expertise is a qualitative indicator, and like for the role, a hierarchy of expertise needs to be defined, which will need to be digitised. We could have for example: High $\Rightarrow \varepsilon_{Dk}=1$, Good $\Rightarrow \varepsilon_{Dk}=0.5$, Low $\Rightarrow \varepsilon_{Dk}=0.25$. This kind of digitisation can be used for almost all qualitative indicators.

### *Criterion 3: knowledge pertinence*

$c_{1.3}$ is obtained from one parameter, which is the relevance of the knowledge according to the CoP's domain and objectives. The relevance of a knowledge *k* can be indicated by a coefficient $\beta(k,n) \in [0,1]$ representing the matching between *k* and a need $n \in N$ explicitly formulated. In a simplistic form, $\beta$ could be binary: 1 if there is a match, 0 if not. But as a piece of knowledge most always answers only partially a need, we keep matching values in a continuous range. We postulate that $c_{1.3}(k)$ increases with two parameters: the number of needs *k* satisfies and the fact that a need satisfied by *k* was or was not previously unsatisfied. Consequently, we write:

$$\boxed{c_{1.3}(k) = \underset{n_i \in N}{\mathrm{E}}\{\beta(k, n_i)(1 - \underset{k_i \in K, k_i \neq k}{\mathrm{E}}\{\beta(k_i, n_i)\})\}}.$$

Hence, $c_{1.3}(k)$ is defined as the average of the matching values between *k* and the needs it satisfies, balanced by the mean of matching values for all the other pieces of knowledge in the CoP already satisfying each need.

### P2: Exchanges of knowledge and interactions within the CoP

**Objective:** measure exchanged knowledge value and members participation.

Proposition 2 concerns the value of a piece of knowledge k regarding the exchanges and interactions it generates and can be valuated according to three criteria:

- $c_{2.1}$: interaction and cumulativity which means quality of exchanged knowledge
- $c_{2.2}$: exchanges which is quantity of knowledge exchanges
- $c_{2.3}$: identity: access level to knowledge

The way knowledge can be accessed globally influences interactions and exchanges. Then $c_{2.3}$ balances both $c_{2.1}$ and $c_{2.2}$. We write:

$$\boxed{val_{P2}(k) = \alpha_3.c_{2.3}(k).wS(c_{2.1}(k), c_{2.2}(k))},$$

where $c_{2.1}$ is the value related to the quantity of interactions generated by *k*, $c_{2.2}$ is the value related to the quality of exchanges generated by *k* and $c_{2.3}$ is the accessibility of k. Here, all the $\alpha_i$ in wS are positive.

### *Criterion 1: Interaction / cumulativity*

$c_{2.1}$ is obtained from one parameter, which is the quality of knowledge exchanged based on k. This can be indicated by the length and frequency of reactions on *k*, the history of *k* through its different versions, and the number of comments on *k*. Each indicator do not influence the others. We write:

$$\boxed{c_{2.1}(k) = \mathrm{wS}\left(reac(k), hist^{rel}(k), nb\_comments^{rel}(k), nb\_ratings^{rel}(k)\right)},$$

where all the $\alpha_i$ in wS are positive. The number of comments (*nb_comments*) and number of ratings (*nb_ratings*) can be obtained directly. No distinction is made between good or bad feedbacks. The other indicators are influenced by different variables, which we detail above.

- The coefficient *reac*(*k*) takes into account the length and frequency of reactions to *k*, which can be e.g. a comment or a rating. We consider the average period of reactions: $\overline{Tr}(k) = \underset{l}{\mathrm{E}}\{time(r_k(l+1)) - time(r_k(l))\}$, where $r_k(l)$ is the $l^{st}$ reaction on *k* and $time(r_k(l))$ is the time at which it as happened. The corresponding average frequency can then be calculated as:

$$\boxed{\overline{fr}(k) = \frac{1}{\overline{Tr}(k)} = \frac{nr-1}{\sum_{l=1}^{nr-1}\left(time(r_k(l+1)) - time(r_k(l))\right)} = \frac{nr-1}{time(r_k(nr)) - time(r_k(1))}},$$

  where *nr* is the number of reactions.

  Considering the length of reactions makes sense only for reactions persisting a certain amount of time, like conversations (chats), a sequence of emails or forum posts on the same subject. In this case, we can take into account the average length of reactions $r_k$ over the set of all non-instantaneous reactions, $R_k^{NI}$ :

$$\boxed{\overline{lr}(k) = \underset{r_k \in R_k^{NI}}{E}\{tend(r_k) - tstart(r_k)\}},$$

  where $tend(r_k)$ and $tstart(r_k)$ are respectively the time of the end and of the beginning of the reaction $r_k$.

  If we postulate that a knowledge generating high frequency or long duration reactions implies that the CoP implicitly prizes it, then we can write *reac*(*k*) as:

$$\boxed{reac(k) = \mathrm{wS}\left(\overline{fr}^{rel}(k), \overline{lr}^{rel}(k)\right), \quad \text{with } \alpha_1, \alpha_2 > 0 \text{ and } \alpha_2 = 0 \text{ if } \overline{lr}(k) = 0}.$$

  As we do not give more importance to one of persisting or instantaneous reactions, forcing $\alpha_2$ to zero when there is no persistent reactions allows to avoid penalising knowledge having generated only instantaneous reactions.

- We compute the coefficient *hist*(*k*) by considering the frequency at which new versions of a knowledge have been created. This can be balanced if we also take into account the importance of modifications (this can be formalised by using a hierarchy of versions, like with software when considering beta versions, release candidates and releases). We write *fv*(*k*) the mean frequency of *k*'s versioning:

  $$\boxed{\overline{fv}(k) = \frac{nv - 1}{time(v_k(nv)) - time(v_k(1))}},$$

  where $v_k(i)$ is the $i^{st}$ version of *k* and *nv* is the total number of versions of *k*. Then we write hist(k) as:

  $$\boxed{hist(k) = \underset{v_k}{E}\{vv(v_k)\}.\overline{fv}(k)},$$

  where $vv(v_k)$ is a value given to a version $v_k$ according to a versioning scale.

### *Criterion 2: Exchanges*

$c_{2.2}$ is obtained from one parameter, which is the quantity of knowledge exchanged based on k. This can be indicated by the number of comments on *k* associated to explicit ratings on these comments, the number of downloads of *k*, the explicit rating of *k*, and the number of pieces of knowledge referencing *k*. Since comments added to *k* either adds value to it (in case of positive comments), either diminish its value, the number of comments is relied to comments ratings that will help to see if a comment has a positive impact in calculation. If *k* has been downloaded a certain number of times, it increases its value, as if a certain number of pieces of knowledge reference *k*. If we consider that the explicit rating of *k* balances its value, and is linked to other indicators listed before, we write:

$$\boxed{c_{2.2}(k) = \mathrm{wS}\left(cmtrate^{rel}(k), dlrate^{rel}(k), nb\_kref^{rel}(k)\right).rating(k)},$$

where all the $\alpha_i$ in wS are positive. As in proposition 1, *dlrate(k)=nb_downloads(k)*. The number of references to *k* (*nb_kref*) is obtained directly as soon as the CoP's IT system allows retrieving it. The knowledge rating, *rating*(*k*), is a qualitative parameter that needs to be quantified with a given value scale. The indicator related to comments, which we write *cmtrate*(*k*) is defined above:

- The calculus of *cmtrate*(*k*) depends on the value scale used to rate knowledge. It can be defined as the ratio of highrated comments, on the set of comments on *k*: *cmtrate*(*k*)=(*nb_cmt*(*k*) | *cmt*(*k*) = good)/*nb_cmt*(*k*). We can also consider the average on comment ratings:

  $$\boxed{cmrate(k) = \underset{cmt(k)}{\mathrm{E}}\{rating(cmt(k)\}}.$$

### *Criterion 3: Identity*

$c_{2.3}$ is obtained from one parameter, which is the level of access to knowledge. This can be indicated by the ratio of members participating to exchanges on *k* ($r_{part}(k)$), the ratio of new members in the CoP and the leaving ones, $r_{in/out}$, and the role of members participating to exchanges ($c_R(k)$). The ratio $r_{in/out}$ does not influence the two other indicators. Then, we postulate that the number of participants in an exchange is independent from their specific role in the CoP. However, this can be discussed since there can be specific roles who might be required to participate to an exchange; in this case they should not be taken into account in the value calculation. We write:

$$\boxed{c_{2.3}(k) = \mathrm{wS}\left(r_{in/out}, r_{part}{}^{rel}(k), c_R{}^{rel}(k)\right)},$$

where all the $\alpha_i$ in wS are positive. The three indicators are defined as follows:

- The indicator $r_{in/out}$ is simply given as the ratio between the number of subscriptions minus the number of cancellation (*i.e.* members leaving the CoP) and the number of subscription taken since the beginning of the CoP. This will give us a ratio of members staying in the CoP compared to the total number of subscriptions in the CoP:

$$\boxed{r_{in/out} = \frac{nb\_subscriptions - nb\_cancellations}{nb\_subscriptions}}.$$

- Access to knowledge can be measured observing the ratio of participants to some collaboration, interaction or exchange. Let note $exch_i(k)$ an exchange between members generated by a piece of knowledge $k$. The value of $k$ is increased when the number of participants to such exchanges is high regarding the number of CoP members or the members that could potentially have been participants. We can take the average over all exchanges generated by $k$ and write:

$$\boxed{r_{part}(k) = \frac{\underset{exch_i(k)}{\mathrm{E}}\left\{nb\_part(exch_i(k))\right\}}{nb\_members}},$$

  where $nb\_exch(k)$ is the number of exchanges generated by $k$ and $nb\_part(exch_i(k))$ is the number of participants to one such exchange. $nb\_members$ is the total number of CoP members, but can be replaced by the number of members potentially interested by the exchange. This could be obtained for example having profiles of members allowing to perform matchmaking with the subject of an exchange and thus determine if an exchange can be of interest for a member.

- The coefficient linked to the roles of exchanges participants, $c_R(k)$, depends on the role distribution in each exchange. For example, we can assume that exchanges in which multiple roles participate have more value than exchanges concerning one single role, because they cover a better diversity of members. Additionally, some role might not have an added value in an exchange, in the case their presence is mandatory according to the CoP's rules. Assuming a value can be calculated for each exchange generated by $k$, we consider the average:

$$\boxed{c_R(k) = \underset{exch_i(k)}{\mathrm{E}}\left\{c_R(exch_i(k))\right\}}.$$

### P3: Assimilation and anticipation of the created value

**Objective:** measure the knowledge use potential value.

Proposition 3 concerns the CoP's members and their capacity to anticipate the value of knowledge or assimilate it. It basically states that the value of *k* is influenced by the capacity of members to be conscious of and to use the value generated by *k*. Hence the value derived from this proposition balances both P1 and P2 as written in section 3.2.1. P3 can be valuated according to two criteria:

- $c_{3.1}$: capacity of members to anticipate value created by knowledge
- $c_{3.2}$: capacity of members to assimilate knowledge

If we consider that the assimilation of knowledge is independent from any anticipation concerning its potential value, we can write:

$$\boxed{val_{P3}(k) = \mathrm{wS}\left(c_{3.1}{}^{rel}(k), c_{3.2}{}^{rel}(k)\right)},$$

where all $\alpha_i$ are positive.

### *Criterion 1: Anticipation capacity of members*

$c_{3.1}$ is obtained from two parameters: exchanges and interactions generated by knowledge, *i.e.* $val_{P2}(k)$, $\forall k \in K$, and the receptivity of members: *recep*(*M*). The latter obviously influences the member's participation, but we consider this influence is already implicitly contained in $val_{P2}$. Then, we write:

$$\boxed{c_{3.1}(k) = \mathrm{wS}\left(\underset{k}{\mathrm{E}}\left\{val_{P2}(k)\right\}, recep(M)\right)},$$

where all the $\alpha_i$ in wS are positive.

- Receptivity can be indicated by the repartition of knowledge on all the IT tools or services used by the CoP, managing knowledge and making it accessible to users. We write:

  $$\boxed{recep(M) = \underset{K}{\mathrm{E}}\left\{serv\_repart(k)\right\}},$$

  *serv_repart*(*k*) being a function of the repartition of a piece of knowledge *k* on the CoP's IT system. In a simplified form, this function could consider the number of occurrences of a same knowledge on different services.

### *Criterion 2: Assimilation capacity of members*

$c_{3.2}$ is obtained from one parameter which is the feedback given on knowledge. This can be indicated by the ratio of positive versus negative reactions (if the system can distinguish the positive feedback from the negative ones), balanced by explicit confidence coefficient that members can put on their reactions quantifying the usefulness they think their input has. Let *reac*(*k*,*m*) be a reaction from a member *m* concerning a knowledge *k* and $conf(m, reac(k,m)) \in [0,1]$ be a confidence coefficient provided by *m* for his reaction concerning *k*. The global feedback *fbk*(*k*,*m*) of *m* concerning *k* is the ratio of positive reactions versus negative ones:

$$\boxed{fbk(k,m) = \frac{\sum\limits_{reac(k,m)\,is\ positive} conf(m, reac(k,m))}{\sum\limits_{reac(k,m)\,is\ negative} conf(m, reac(k,m))}}.$$

The capacity of assimilation of members can be obtained by averaging first on all the knowledge on which each user has given feedback, and second on all the members:

$$\boxed{c_{3.2}(k) = \underset{m}{\mathrm{E}}\left\{\underset{k}{\mathrm{E}}\left\{fbk(k,m)\right\}\right\}}.$$

## 3.2.2. Realised Value of Knowledge

The global or realised value of *k* is obtained taking into account proposition P5. We write the non-standardised value of knowledge as:

$$\boxed{\begin{aligned} val^{nstd}(k) &= \mathrm{wS}\left(val_{pot}(k), val_{P5}(k)\right) \\ &= \mathrm{wS}\left(\alpha . val_{P3} . \mathrm{wS}\left(val_{P1}(k), val_{P2}(k)\right), val_{P5}(k)\right) \end{aligned}},$$

where all the $\alpha_i$ in wS are positive. As we will see in the details of the formula for P5 hereafter, *val*(*k*) is a recursive function because $val_{P5}$ depends on the value of knowledge it generates, which is itself calculated using *val*(*k*). As for the potential value, the standardised version for the realised value is finally written as:

$$\boxed{val(k) = \frac{val^{nstd}(k) - \min\{val^{nstd}(k)\}}{\max\{val^{nstd}(k)\} - \min\{val^{nstd}(k)\}}}.$$

### P5: Reification

**Objective:** measure the value of knowledge in terms of reification, centered on expressed returns on experience.

Proposition 5 concerns the value of a piece of knowledge *k* in terms of reification, which is a function of CoPs members reification capacity and willingness to contribute. It can be valuated according to three criteria:

- $c_{5.1}$: members capacity of reification
- $c_{5.2}$: capacity and willingness of members to restitute knowledge
- $c_{5.3}$: value of knowledge reified from *k*

The first two criteria can be considered independently, as the capacity of a person to use a piece of knowledge is independent from the fact he will actually share or not experiences. Then, the realised value of knowledge reified from *k* depends on these two parameters, but as this influence will be taken into account in the calculus of $c_{5.3}$, we propose:

$$\boxed{val_{P5}(k) = \mathrm{wS}\left(c_{5.1}, c_{5.2}, c_{5.3}(k)\right)},$$

where all the $\alpha_i$ in wS are positive, $val_{P5}$ is positive.

### *Criterion 1: reification capacity of members*

$c_{5.1}$ is obtained from one parameter, which is the return on experience provided by members. For a member *m*, we define the reification capacity *rei*(*m*) (in the form of returns on experience explicitly provided to the CoP) by the ratio between the number of reified knowledge (*nb_roe*(*m*)) and the number of knowledge *m* has accessed (e.g. viewed, commented, downloaded) (*nb_accessed_k*(*m*)):

$$\boxed{rei(m) = \frac{nb\_roe(m)}{nb\_accessed\_k(m)}}.$$

Then, $c_{5.1}$ is defined as the average reification capacity of all the members of the CoP:

$$\boxed{c_{5.1} = \underset{m}{\mathrm{E}}\left\{rei(m)\right\}}.$$

### *Criterion 2: capacity and willingness of members to restitute knowledge*

$c_{5.2}$ is obtained from one parameter, which is the diffusion of a member's return on experiences *roe*(*m*). This can be indicated by the number of links to other knowledge explicitly given by a member when he diffuses a return on experience. Taking the average on all the members of the average number of links they have provided for each *roe*(*m*), we write:

$$\boxed{c_{5.2} = \underset{m}{\mathrm{E}}\left\{\underset{roe(m)}{\mathrm{E}}\left\{nb\_links(roe(m))\right\}\right\}}.$$

### *Criterion 3: reified knowledge value*

The value of knowledge globally fluctuates according to the number and value of all the generated returns on experiment. Then, we define $c_{5.3}$ as the average of the values of such reifications:

$$\boxed{c_{5.3}(k) = \underset{roe(k)}{\mathrm{E}}\left\{val(roe(k))\right\}},$$

where $roe(k)$ is an explicit return on experience given on $k$. $c_{5.3}$ is positive.

## 3.2.3. Value of the Knowledge Base

As said, we can estimate the value of the KB handling the knowledge of the CoP with proposition P6 and P7. As both can be considered as independent because they are related to different functionalities of the tools supporting the KB, we write, in a non-standardised form:

$$val^{nstd}(KB) = \mathrm{wS}\left(val_{P6}(KB), val_{P7}(KB)\right),$$

where all the $\alpha_i$ in wS are positive, val(KB) is positive. In a standardised form, we have:

$$\boxed{val(KB) = \frac{val^{nstd}(KB) - \min\{val^{nstd}(KB)\}}{\max\{val^{nstd}(KB)\} - \min\{val^{nstd}(KB)\}}}.$$

**P6: Knowledge base in the CoP**

**Objective**: to measure the KB capacities

Proposition 6 concerns the value of the KB in terms of its capacities. For the calculation, two criteria are taken into account. We write:

$$\boxed{val_{P6}(KB) = \mathrm{wS}(c_{6.1}(KB), c_{6.2}(KB))},$$

where $c_{6.1}$ is the value related to the availability of knowledge, and $c_{6.2}$ is the value related to the KB coherency. All the $\alpha_i$ in wS are positive.

### *Criterion 1: knowledge availability*

$c_{6.1}$ is obtained from one parameter: the accessibility of knowledge in the KB, which can be indicated by the characteristics of the services or IT tools used in the CoP and the quantity of knowledge actually accessed. As a CoP uses different services with different functionalities, the knowledge used and produced by the CoP is often distributed in each service's own KB. In the following, we refer to CoP's services or their managed KB indifferently as $kb_i$, with $kb_i \subset KB$. We define $c_{6.1}$ as:

$$\boxed{c_{6.1}(KB) = \mathrm{wS}\left(val\_func(KB), access(KB)\right),}$$

where all the $\alpha_i$ in wS are positive.

- Let $func(kb_i)$ be a functionality of a $kb_i$. We have identified the following non-exhaustive list: access, search functionality, notification, and collaborative edition. In the simplest manner, we can quantify the presence or absence of each functionality: the more are present, the more the value of KB increases. For the whole KB, we write:

  $$\boxed{val\_func(KB) = \underset{kb_i}{\mathrm{E}}\left\{\underset{func(kb_i)}{E}\left\{val\_func(kb_i)\right\}\right\}},$$

  where $val\_func(kb)$ is the value given to a functionality of a $kb$, which might be binary as explained.

- We define *access*(*KB*) as the mean number of access to a piece of knowledge stored in KB:

$$\boxed{access(KB) = \underset{k}{\mathrm{E}}\{nb\_access(k)\}}.$$

### Criterion 2: KB coherency

$c_{6.2}$ is obtained from two parameters which are the cumulativeness and redundancy of knowledge in the KB. From a general point of view, redundancy can be considered as a bad thing. However, redundancy might generate specific links between pieces of knowledge , thus increasing cumulativeness. This duality will be considered in the redundancy formula. Then we write:

$$\boxed{c_{6.2}(KB) = \mathrm{wS}(rdcy(KB), cumul(KB))},$$

where all the $\alpha_i$ in wS are positive. *rdcy*(KB) and *cumul*(KB) are respectively measures of the KB's redundancy and cumulativeness. The two parameters are defined hereafter.

- Redundancy can be indicated by estimating the number of redundant pieces of knowledge present in the KB:

  $$\boxed{rdcy(KB) = \underset{k}{\mathrm{E}}\{nbdble\_p(k) - nbdble\_n(k)\}},$$

  where *nbdble_p*(*k*) is the number of doubles of *k* that exist in the KB having links to other knowledge that are different from those of *k*, and *nbdble_n*(*k*) is the number of doubles of *k* having no links to other knowledge or links with knowledge already linked with *k*. The first set of doubles is considered to be an added value as a link generator, while the second is considered negatively.
- An indicator of the cumulativeness is the number of links existing between stored pieces of knowledge. We can take the average number for one piece *k*:

  $$\boxed{cumul(KB) = \underset{k}{\mathrm{E}}\{nblinks(k)\}},$$

  where *nblinks*(*k*) is the number of links implying *k* (*i.e.* from *k*, or to *k*). This simple form might be extended, by taking into account the repartition of links among the stored knowledge: we can assume that a wide coverage of links on the whole knowledge has more value than if only a few pieces are linked together.

## P7: Interaction with the KB

**Objective**: measure the KB activity

Proposition 7 concerns the KB activity. It can be calculated through two criteria:

- $c_{7.1}$: the knowledge validation in the KB
- $c_{7.2}$: the knowledge modification in the KB

As these two criteria are clearly independent, the activity of the KB can be calculated as following:

$$\boxed{val_{P7}(k) = \mathrm{wS}(c_{7.1}(k), c_{7.2}(k))},$$

where $c_{7.1}$ is the value related to the <u>validation of *k*</u> and $c_{7.2}$ is the value related to the <u>modification of *k*</u>. All the $\alpha_i$ in wS are positive.

### Criterion 1: knowledge validation

$c_{7.1}$ objective is to measure how all different *kbs* allow to validate knowledge. It can be defined as the average on the $c_{7.1}(kb_i)$ which are the value of $c_{7.1}$ for all *kbs*:

$$\boxed{c_{7.1}(k) = \underset{kb_i}{\mathrm{E}}\{c_{7.1}(kb_i)\}}.$$

$c_{7.1}(kb_i)$ is obtained from three parameters: the possibility to categorise knowledge, to evaluate it and to determine its relevance:

$$\boxed{c_{7.1}(kb_i) = \mathrm{wS}(categ(kb_i), eval(kb_i), relev(kb_i))},$$

where all the $\alpha_i$ in wS are positive.

The different indicators are detailed hereafter.

- Categorisation of *k* in *kb*: *categ(kb<sub>i</sub>)*. The possibility to categorise *k* in the tool is an added value of the *kb*. This indicator is a boolean: it is possible or not to categorise *k* in the *kb*.
- Evaluation of k in the kb: $eval(kb_i)$. As with the categorisation, the possibility to evaluate *k* in a *kb* is an added-value. This indicator is linked to the possibility of making comments (=*comment(kb)*), and on attributing ratings (=*rating(kb)*) in the *kb*. These indicators are booleans, and can be considered together:

$$\boxed{eval(kb_i) = \frac{comment(kb_i) + rating(kb_i)}{2}}$$

- Relevance of knowledge in the *kb*: $relev(kb_i)$. The possibility to make links between different pieces of knowledge is a facilitator for providing relevant knowledge. In fact, if a piece of knowledge is linked to others, it tends to show that this piece of knowledge deals with the CoP domain, and is not a free electron. As for the previous indicators, the relevance is a boolean: the feature is present or not.

### *Criterion 2: knowledge modification*

$c_{7.2}$ objective is to measure how all *kb*s enable modifications of knowledge. It can be defined as the average on the $c_{7.2}(kb_i)$ which represents the ability, a particular *kb* has to modify knowledge. We write:

$$\boxed{c_{7.2}(k) = \underset{kb_i}{\mathrm{E}}\{c_{7.2}(kb_i)\}}$$

$c_{7.2}(kb_i)$ takes into account three parameters: knowledge versioning ($vers(kb_i)$), knowledge updating ($upd(kb_i)$), and knowledge obsolescence ($obs(kb_i)$), measuring how a *kb* enables to identify obsolete or putrefied knowledge. We can write:

$$\boxed{c_{7.2}(kb_i) = \mathrm{wS}(vers(kb_i), upd(kb_i), obs(kb_i))},$$

where all the $\alpha_i$ in wS are positive.

Since all parameters have the same importance, we can suppose that $\alpha_1 = \alpha_2 = \alpha_3 = 1$. $vers(kb_i)$, $upd(kb_i)$ and $obs(kb_i)$ are respectively the ability of the *kb* to version, update and make it obsolete.

- The ability of a *kb* to version a knowledge ($vers(kb_i)$) can be indicated by: historical statistics ($stath\_v(kb_i)$), the possibility to add comments to a knowledge version ($comm\_v(kb_i)$), and the possibility to trace actions made on a knowledge ($traca\_v(kb_i)$). $stath\_v(kb_i)$, $comm\_v(kb_i)$ and $traca\_v(kb_i)$ are all booleans indicating respectively whether a *kb* proposes historical statistics of modifications on knowledge, enables to add comments to a version, gives the possibility to trace actions. We write:

$$\boxed{vers(kb_i) = \mathrm{wS}(stath\_v(kb_i), comm\_v(kb_i), traca\_v(kb_i))}$$

- The ability of a *kb* to update a knowledge (*upd(kb$_i$)*) can be indicated by the possibility to trace actions made on a knowledge (*traca_u(kb$_i$)*). *traca_u(kb$_i$)* is a boolean indicating whether a *kb* gives the possibility to trace actions. We write:

$$\boxed{upd(kb_i) = traca\_u(kb_i)}$$

- The ability of a *kb* to make a knowledge obsolete (*obs(kb$_i$))* can be indicated by: historical statistics (*stath_o(kb$_i$)*), and the possibility to trace actions made on a knowledge (*traca_o(kb$_i$)*). *stath_o(kb$_i$)* and *traca_o(kb$_i$)* are booleans indicating respectively whether a *kb* proposes historical statistics of modifications realised on knowledge, and gives the possibility to trace actions. We write:

$$\boxed{obs(kb_i) = \mathrm{wS}(stath\_o(kb_i), traca\_o(kb_i))}$$

## 3.3. Description of the Architecture of the Future Service

### 3.3.1. Architecture Schema

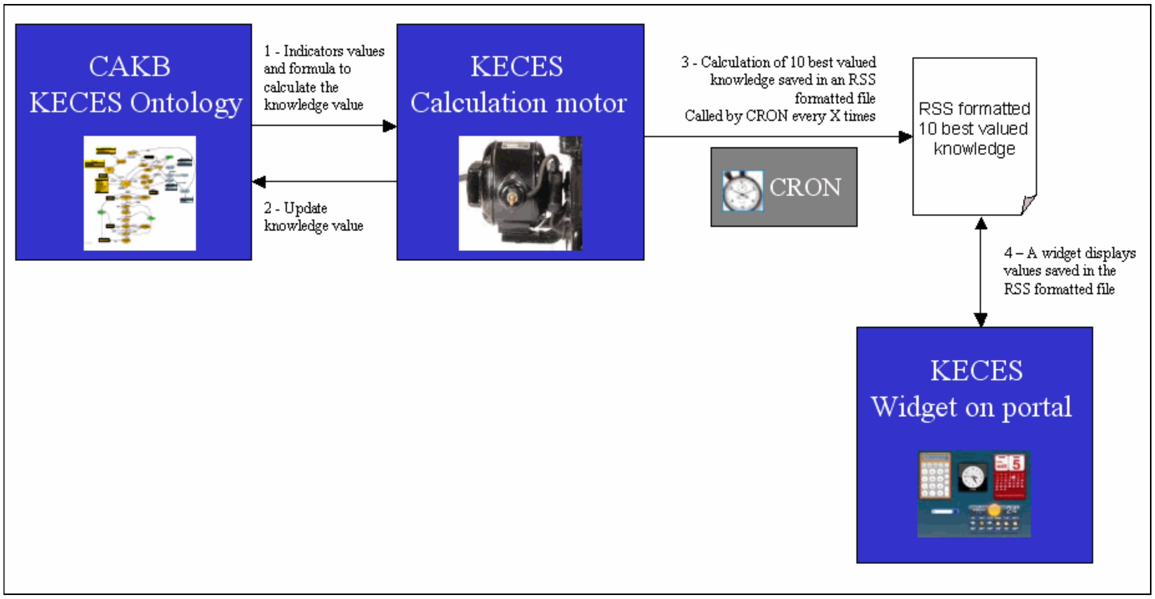

CAKB
KECES Ontology
1 - Indicators values and formula to calculate the knowledge value
2 - Update knowledge value
KECES
Calculation motor
3 - Calculation of 10 best valued knowledge saved in an RSS formatted file
Called by CRON every X times
CRON
RSS formatted 10 best valued knowledge
4 – A widget displays values saved in the RSS formatted file
KECES
Widget on portal

*Figure 22. KECES architecture schema*

### 3.3.2. CAKB KECES Ontology

The Knowledge Exchanged in a Cop Evaluation Service (KECES) architecture is based upon a Cross Awareness Knowledge Base (CAKB) ontology that will store indicators values, formulas enabling to calculate knowledge value, and the value of each knowledge.

#### Ontology description

The ontology, centralised on knowledge, aims at representing in one side all indicators linked together (the upper part), and in another side knowledge values.

The part concerning the indicators is inspired from the model presented in the 3.1.4 section: “Towards a knowledge value model for virtual CoPs” (Figure 20): the different criteria listed to determine the knowledge value are linked to the concept of knowledge.

We can notice that in this part of the ontology, some concepts derive from the O’CoP ontology, namely the community, actor, member, role and collaboration concepts.

In addition to that, the author and provider of the knowledge, as well as their reputation value are represented. Moreover, the feedback, positive and negative, the rating and comment done on the knowledge by a CoP’s member are present.

As explained in the propositions 6 and 7, the knowledge is stored in a knowledge base that possesses, or not, certain functionalities, such as notification, search, access form (public/private), collaborative edition, and possibility of rating, making comment, and versioning.

Furthermore knowledge has some properties also represented: its publication date, its version history, number of comments, downloads, references, and relations with others knowledge in the KB, mean of its notes, and its classification.

We can notice that the concept of Knowledge is linked to the concept of Knowledge Value Facet, which represents the propositions presented above that determine the list of indicators used to attribute a value to knowledge.

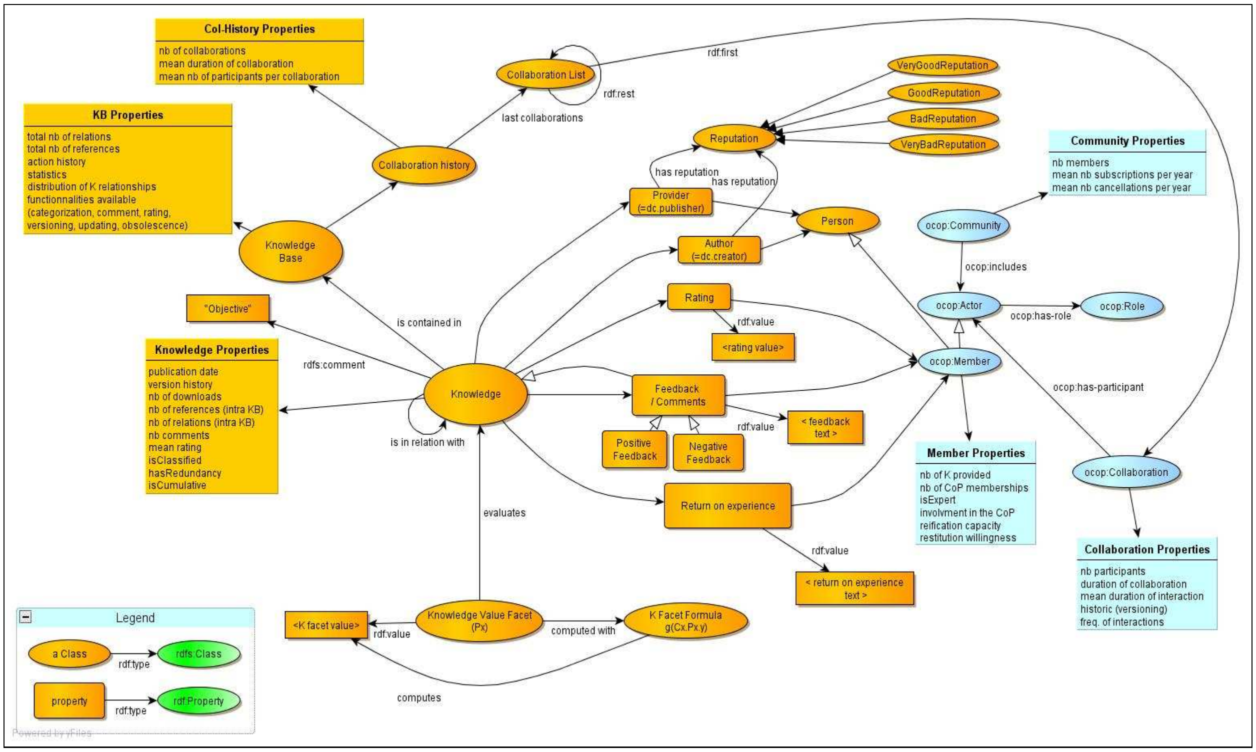


*Figure 23. KECES ontology*

### 3.3.3. KECES Calculation Motor

The KECES calculation motor will interrogate this ontology by the way of CAKB REST (REpresentational State Transfert) Web services, and will be the one doing the calculation of

knowledge value on the basis of indicators values and formulas stored in the ontology.
KECES calculation motor will provide functions to present different views on knowledge. For instance, it will provide a function determining the ten best-valued pieces of knowledge to be shown in a widget.
The result of this function, which will be provided as a REST Web service, will be saved in an RSS formatted file. The REST Web service will be called by a CRON every x times (for example every 20 minutes), so that the RSS file is updated with current values every x times.

#### CRON [wikipedia]

*CRON is a time-based scheduling service in Unix-like computer operating systems. The name is derived from Greek "chronos" (χρόνος), meaning time.*
*CRON is driven by a "crontab", a configuration file that specifies shell commands to run periodically on a given schedule.*

#### RSS format

The RSS format contains for each item its title, its description and the link to access the knowledge details in the CAKB. Further information concerning KECES results are saved in a particular type named <keces:keces>. In this tag can be found a general value (the knowledge value), stored in <keces:global>, but also all propositions values, in <keces:propositions>.

```
<?xml version="1.0"?>
<rss version="2.0" xmlns:keces="http://www.tudor.lu/Ontologie/KECES.rdfs">
  <channel>
    <title>KECES results</title>
    <description>KECES results</description>
    <link>{REST WS}</link>  <!-- link to the REST WS -->
    <item>
      <title>   <!-- knowledge title -->
        {title}
      </title>
      <description>  <!-- knowledge description -->
        {description}
      </description>
      <link>  <!-- link to access the knowledge -->
        {link to knowledge details}
      </link>
      <keces:keces>  <!-- additional values calculated by keces -->
        <keces:global>[0;1]</keces:global>  <!-- global value of the knowledge -->
        <keces:propositions>
          <keces:p1>[0;1]</keces:p1>
          <keces:p2>[0;1]</keces:p2>
          <keces:p3>[0;1]</keces:p3>
          <keces:p5>[0;1]</keces:p5>
          <keces:p6>[0;1]</keces:p6>
          <keces:p7>[0;1]</keces:p7>
        </keces:propositions>
      </keces:keces>
    </item>
  </channel>
</rss>
```

### 3.3.4. KECES Widget

The values stored in the RSS file will be shown on a widget, in the PALETTE portal.

The first view of the widget will show a list of the ten best-valued pieces of knowledge, with for each knowledge its general value. Clicking on a knowledge value will allow to access detailed information on the value which can be a graph representing values for P1 to P7.

## 3.4. Limits and Discussion

It is important to notice that the knowledge value model developed here appears to be relevant in the case of a particularly mature virtual CoP, able to exchange best practices efficiently and to produce new knowledge. But as (Gongla and Rizzuto 2001) have pointed out, CoPs (and therefore virtual CoPs) evolve through different stages of development that highlight different needs in collaboration technologies. Several authors such as Gongla and Rizzuto (2001) and Wenger et al. (2002) have shown that like any other living things, CoPs evolve, go through a natural cycle of birth, growth, and death (see Chapter 2 for more details on CoP evolution). Hence, the virtual CoPs life length depends on a smart balance between autonomy and formalisation, in which the importance of collaboration technologies emerges, according to its stage of evolution. Of course, this must be considered in the use of the previous knowledge value model depicted in Figure 20 in order to match collaborative technologies and knowledge management issues with the virtual CoP stage of development.

For instance, Gongla and Rizzuto (2001) have noticed that during the first steps of their development, virtual CoPs focused on issues of connections and communication, while during their active and maturing phases virtual CoPs focused on issues of collaboration and learning. It is therefore necessary to study the considered virtual CoP life cycle in order to identify expected needs by distinguishing essential, useful, and non-essential functionalities (and related indicators as well).

## 3.5. Conclusions

The proposed formula calculating the value of knowledge from the perspective of the CoP member still need to be tested, tuned and adapted with simulated or real data. One important thing we have not considered yet is the temporal aspect of the used indicators and calculated values. For several criteria, it might be important to consider the sampling of indicators at different period of time. For others, evolution in time of some indicators might bring some additional information that can be considered to balance the value.

Needless to say that the usefulness of such value is directly linked with, on one side the CoP functioning and the IT tools it uses, and on the other side with the manner it will be presented to CoP members. Hintssuch as completeness of information might be provided to give a mean to assess the validity of processed values.
As we have seen, there are various indicators implied to calculate the knowledge value.
Indeed the more indicators informed we have, the higher the value obtained will be close to reality.
According to the numbers of indicators obtained for the calculation, we can determine a completeness index that informs us on the relevance of the knowledge value and on the volume of information obtained.

$$\boxed{completeness(k) = \frac{nb\ indicators\ Informed}{nb\ total\ indicators}}$$

The realised prototype might also highly influence the assessment of the approach. The proposed architecture might be implemented as a proof of concept and is not a final proposition. Future enhancements concern particularly the display of information by the way of different widgets. A possibility would be to show all propositions values using a radar graph, which would help showing a knowledge profile. Another way to represent knowledge value would be to show its evolution during time. Different views on knowledge could be shown on different widgets in addition to the ten best valued knowledge such as, for instance: the ten knowledge authors having posted the best valued pieces of knowledge, the ten best reified values, the ten pieces of knowledge having the less values, the ten pieces of knowledge that have had in time the best value. There exist lots of possibilities.

# Chapter 4: Conclusion and Further Work

We succinctly recap the research work done about the two services considered in this deliverable and present the research work which remains to be done during the last months of the Palette project.

## 4.1. Knowledge Evolution Service

*What has been done*

In order to design the Knowledge Evolution Service envisioned for CoPs in the Palette project, a service is developed aiming at managing the evolution of two kinds of knowledge (ontologies and semantic annotations) materialised in what we have called a “CoP Memory”. (1) We have identified,using CoP evolution cycle models as analysis frameworks, (a) the events that can originate the evolution of the community knowledge and (b) the types of knowledge evolution that can result from (a). Because community memory evolves according to the evolution of the community itself, this identification was intended to help understand knowledge evolution better and to support it better. (2) We studied two evolution scenarios of ontology/annotation evolution: with trace and without trace of ontology changes carried out during its evolution. These two scenarios often lead to inconsistencies of the annotations semantics using this modified ontology. (3) For this reason, we have explored two approaches: a procedural approach and a rule-based approach, corresponding to the scenarios, in order to manage semantic annotations evolution. These approaches allow us to detect inconsistent annotations and to guide the process of solving these inconsistencies. (4) These propositions are implemented in the CoSWEM prototype which facilitates the evolution management of the changes. It enables to carry out automatically or semi-automatically inconsistency detection and correction of the semantic annotations. Moreover, CoSWEM can highlight visually the different entities between two ontology versions according to the concept or property hierarchy.

*Further work*

We intend to improve the verification/validation step of ontology/annotation evolution management in CoSWEM. Precisely, we have planned to design a generic type checker based on a model of ontology and annotation type errors. This type checker will be an improvement of the Corese search engine.

## 4.2. Knowledge Evaluation Service

*What has been done*

In order to design the Knowledge Evaluation Service also envisioned for CoPs in the Palette project, we (1) designed a model and indicators of knowledge value from the perspective of a CoP member. To elaborate this model, we studied the different activities occurring in a CoP basing on existing theories and tried to determine pertinent factors influencing positively or negatively the knowledge value. The indicators were then grouped in seven propositions within the model.(2) From these indicators and propositions, we elaborated an algorithm for computing the value of knowledge. Values have been proposed for the main propositions. A distinction was made between values for propositions 1 (incomes – pieces of knowledge), 2 (exchanges of knowledge and interactions within a CoP), 3 (assimilation and anticipation of the created value) and 5 (reification by doing) that qualifypieces of knowledge, and for propositions 6 (knowledge base of the CoP) and 7 (interaction with the knowledge base) that qualify the whole knowledge base; and (3) we described the architecture of the future service that will provide an interface to knowledge value computing. The service will be composed of a calculation motor associated to the model ontology, and will send results into a RSS feed by the way of a CRON system.

*Further work*

We intend to develop a Knowledge evaluation widget, and to test it locally. In order to test it, a test board will be implemented that will permit to simulate indicators values. Those trial values will give us test cases to evaluate the Knowledge evaluation process and the associated Widget. For a set of knowledge pieces in a CoP, the widget will display by knowledge the potential and realised value, and might additionally allow displaying a graph showing the values corresponding to the six propositions of the knowledge value model.

# References

Amin A., Roberts J. (2006). *Communities of Practice: Varieties of Situated Learning*. Dynamics of Institutions and Markets in Europe research paper.

Barlatier P-J., Vidou G., Jacquemart S., Latour T. (2007). *Don't value the valueless: toward a model of evaluation of knowledge within e-CoPs*. 2nd International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, Crete, Greece.

Boland R.J. and Tenkasi R.V. (1995). *Perspective making and perspective taking in communities of knowing. Organization Science,* Vol.6, N°4, 350-372.

Brown J.S., Duguid P., (1991). *Organizational learning and communities-of-practice: toward a unified view of working, learning and innovation. Organization Science,* Vol.2 N°1, 40-57.

Bruijn J., Martin-Recuerda F., Manov D. et Ehrig M. (2004). *State-of-the-art survey on Ontology Merging and Aligning V1*. Deliverable D4.2.1 (WP4). EU-IST Integrated Project (IP) IST-2003-506826 SEKT.

Cohen D., (1998). *Toward a Knowledge Context: report on the first annual U.C Berkeley forum on knowledge and the firm. California Management Review,* Vol.40, N°3, 22-39.

Cook D.N., Brown, J.S., (1999). *Bridging epistemologies: the generative dance between organizational knowledge and organizational knowing. Organization Science,* Vol.10, N°4, 381-400.

Corby O., Dieng-Kuntz R., and Faron-Zucker C. (2004). *Querying the Semantic Web with the CORESE search engine*. In Proceedings of the 16th ECAI (ECAI'2004), edited by R. Lopez de Mantaras and L. Saitta, Valencia, pp. 705-709, IOS Press Amsterdam.

Daele A., (2006). A model for representing professional development through the participation in a virtual CoP: uses for developing enhanced services. 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, Crete, Greece.

Dieng-Kuntz R., Corby O., Gandon F., Giboin A., Golebiowska J., Matta N. , Ribière M. (2005) Knowledge Management. *Méthodes et outils pour la gestion des connaissances* (3ème édition), Paris, Dunod.

Dillenbourg P., Baker M., Blaye A., O'Malley C. (1996). The evolution of research on collaborative learning. In E. Spada & P. Reiman (Eds) *Learning in Humans and Machine: Towards an interdisciplinary learning science*. Oxford: Elsevier, 198-211.

Flouris, G. (2006). *On Belief Change and Ontology Evolution*. PhD thesis, University of Crete, Department of Computer Science.

Gongla P., Rizzuto C. R., (2001). *Evolving Communities of Practice: IBM Global Services Experience*. *IBM Systems Journal*, Vol 40, N°4, 842-862.

Gruber, R. (1993). *A translation approach to portable ontology specifications, Knowledge Acquisition*, An International Journal of Knowledge Acquisition for Knowledge-Based Systems, Volume 5, Number 2, pp.199-220.

Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H. et Sure, Y. (2005). *A framework for handling inconsistency in changing ontologies*. In Y. Gil, E. Motta, V. Benjamins, and M. Musen, editors, Proceedings of the 4th International Semantic Web Conference (ISWC ’05), volume 3729 of Lecture Notes in Computer Science, pages 353–367. Springer.

Jawahar I. M., McLaughlin G. L., “Toward a Descriptive Stakeholder Theory: An Organizational Life Cycle Approach”, *Academy of Management Review*, 26(3), 2001, p.397-414.

Klein, M. 2004. *Change management for distributed ontologies*. Ph.D. thesis, Vrije Universiteit Amsterdam.

Klein, M. et Fensel, D. (2001). *Ontology versioning for the Semantic Web*. In Proceedings of the First International Semantic Web Working Symposium (SWWS), pages 75-91, Stanford University, California, USA.

Klein, M., A. Kiryakov, D. Ognyanov, and D. Fensel. 2002. *Ontology versioning and change detection on the Web*. In Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW-2002), pp. 197–212, Siguenza, Spain, October 2002.

Kogut B., Zander U., (1996). *What Firms Do? Coordination, Identity and Learning*. *Organization Science*, Vol.7, N°5, 502-518.

Kühn, O., Abecker, A. (1997). Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges, *Journal of Universal Computer Science*, vol. 3, no. 8 (1997), 929-954.

Lave J., Wenger E., (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press.

Luong P-H. 2007. *Gestion de l'Evolution d'un Web Sémantique d'Entreprise*. Ph.D. thesis, Ecole des Mines de Paris.

Luong P-H. and Dieng-Kuntz R. (2007) *A Rule-based Approach for Semantic Annotation Evolution*. The Computational Intelligence Journal, 23(3), 320-338. Blackwell Publishing, Malden, MA 02148, USA, and Oxford, OX4 2DQ, UK, 2007.

McDermott R., (1999). Learning Across Teams: The role of communities of practice in team organizations, *Knowledge Management Review*, May/June.

McDermott, R. (2000). Community Development as a Natural Step: Five Stages of Community Development, *KM Review 3*, No. 5 (November/December 2000).

Meyer M.W., Gupta V., (1994). The Performance Paradox. In Staw B.W., Cummings L., (Eds.), *Research in Organizational Behavior*, JAI Press, Greenwich, CT, 309-369.

Miller, E. et Manola, F. (2004). *RDF Primer*. W3C Resource Description Framework, available at http://www.w3.org/RDF.

Moingeon, Bertrand & Quélin, Bertrand & Dalsace, Frédéric & Lumineau, Fabrice, 2006. Inter-organizational communities of practice: specificities and stakes, *Les Cahiers de Recherche 857*, Groupe HEC.

Nahapiet J., Ghoshal S., (1998). *Social Capital, Intellectual Capital, and the Organizational Advantage*. *Academy of Management Review*, Vol. 23, N°2, 242-266.

Nonaka I., Takeuchi H., (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. New York: Oxford University Press.

Orr J., (1990). *Talking about machines: An ethnography of a Modern Job*, Cornell University.

Quinn R. E., Cameron K., "Organizational Life Cycles and Shifting Criteria of Effectiveness: Some Preliminary Evidence", *Management Science*, 29(1), 1983, p.33-51.

Roddick, J.F. (1996). *A survey of schema versioning issues for database systems*. Information and Software Technology, 37(7):383–393.

Scarbrough H., Swan J., Preston J., (1999). *Knowledge Management: A Literature Review*, Institute of Personnel and Development: London.

Siesfeld G.A., (1998). *The Measurement of Knowledge: Introduction*. In Neef D., Siesfeld G.A., Cefola J. (Eds). *The Economic Impact of Knowledge*. Butterworth-Heinemann, 191-202.

Stojanovic, L. (2004). *Methods and tools for ontology evolution*. Ph.D. Thesis, University of Karlsruhe.

Stojanovic, L., A. Maedche, N. Stojanovic, and B. Motik. (2002a). *User-driven ontology evolution management*. In Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management (EKAW-2002), pp. 285–300, Springer-Verlag, Spain, October.

Stojanovic, L., N. Stojanovic, and S. Handschuh. (2002b). *Evolution of the metadata in the ontology-based knowledge management systems*. The 1st German Workshop on Experience Management: Sharing Experiences about the Sharing of Experience, 2002, pp. 65–77.

Tifous, A., Dieng-Kuntz, R., Durville, P., El Ghali, A., Evangelou, C., Giboin, A., and Vidou, G. (2007) *CoP-dependent ontologies*. Deliverable D.KNO.02, Palette FP6-028038, March

Varela F.J., (1989). *Autonomie et Connaissance : Essai sur le Vivant*, Editions du Seuil, Paris.

Visser, P. R. S., Jones, D.M., Bench-Capon, T. J. M., et Shave, M. J. R. (1997). *An analysis of ontological mismatches: Heterogeneity versus interoperability*. In AAAI 1997 Spring Symposium on Ontological Engineering, Stanford, USA.

Von Krogh G., (2003). *Knowledge sharing and the communal resource*. In Easterby-Smith M., Lyles M., (Eds.) *Handbook of Organizational Learning and Knowledge Management*, Malden, MA: Blackwell, pp. 372-392.

Vygotsky L.S., (1978). *Mind in society: The development of higher psychological processes*. Cambridge: Harvard University Press

Weick K.E., (1979), *The Social Psychology of Organizing, 2d ed*. Reading, MA: Addison-Wesley

Wenger, E. C. (1998). Communities of Practice: Learning as a Social System, *Systems Thinker* 9, No. 5, 2–3 (June/July 1998).

Wenger E., (1998). *Communities of Practice: Learning, Meaning and Identity*. Cambridge: Cambridge University Press.

Wenger E., McDermott R., Snyder W, (2002). *Cultivating Communities of Practice: A Guide to Managing Knowledge*, Harvard Business School Press.

Zack M.H., Serino M. (2000). *Knowledge Management and Collaboration Technologies*. In Smith D.E. (Ed.) *Knowledge, Groupware, and the Internet*. Butterworth-Heinemann.