Graph-Based Inferences in a Semantic Web Server for the Cartography of Competencies in a Telecom Valley

Fabien Gandon, Olivier Corby, Alain Giboin, Nicolas Gronnier, and Cecile Guigard

INRIA, ACACIA, 2004 rt des Luciole, BP93, 06902 Sophia Antipolis, France
Fabien.Gandon@sophia.inria.fr
http://www-sop.inria.fr/acacia/

Abstract. We introduce an experience in building a public semantic web server maintaining annotations about the actors of a Telecom Valley. We then focus on an example of inference used in building one type of cartography of the competences of the economic actors of the Telecom Valley. We detailed how this inference exploits the graph model of the semantic web using ontology-based metrics and conceptual clustering. We prove the characteristics of theses metrics and inferences and we give the associated interpretations.

1 Semantic Annotation of Competencies

In knowledge-based solutions, user interfaces have the tricky role of bridging the gap between complex knowledge representations underlying collective applications and focused views tuned to day-to-day uses. For this reason, we believe that interface design and knowledge representation must be tackled in parallel. In this paper, we describe and analyze an experience in simulating the inferences done by economists and management researchers in building a cartography of the competences of the economic actors of a region. The implementation is now part of a public semantic web server maintaining annotations about the actors of a telecom valley. This paper will explain how we designed such an inference using ontology-based metrics defined above the graph structure of the semantic web annotations statements, but before we go in such details we need to introduce the overall project: the Knowledge management Platform (KmP[1]) of the Telecom Valley of Sophia Antipolis[2].

The goal of KmP was the elaboration of a public repository supporting three application scenarios: (1) promoting the Scientific Park of Sophia Antipolis and its international development by providing the local institutions with a pertinent and up-to-date snapshot of the park. (2) facilitating partnerships between different industrial firms of the park. (3) facilitating collaboration on projects between industrial partners and the different research institutes. This platform is available online[3] and relies on a semantic web server publicly available for all the actors of the value chain of the Telecom Valley of Sophia Antipolis. The steering committee of KmP is composed of eleven pilot companies involved in the specifications of the application and the

[1] http://www-sop.inria.fr/acacia/soft/kmp.html
[2] http://www.sophia-antipolis.org/index1.htm
[3] http://beghin.inria.fr/

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 247 – 261, 2005.

population of the ontologies: Amadeus, Philips Semiconductors, France Telecom, Hewlett Packard, IBM, Atos Origin, Transiciel, Elan IT, Qwam System and Cross Systems.

KmP is a real world experiment on the design and usages of a customizable semantic web server to generate up-to-date views of the telecom valley and assist the management of competencies at the level of the organizations (companies, research institute and labs, clubs, associations, government agencies, schools and universities, etc.). This platform aims at increasing the portfolio of competences of the technological pole of Sophia Antipolis by helping companies, research labs and institutions express their interests and needs in a common space in order to foster synergies and partnerships. The platform implements a public knowledge management solution at the scale of the telecom valley based on a shared repository and a common language to describe and compare the needs and the resources of all the organizations.

Ontologies were built from models provided by domain experts [Lazaric & Thomas, 2005] and end-users: models of competencies, models of the telecom domains (networks, computer science, etc.), task models, value chain of the telecom valley, etc. The implementation merges the frameworks of the semantic web (RDF, RDFS), the classic web (HTML, CSS, SVG) and the structured web (XML, XSLT) to integrate data coming from very different sources, allow queries from different viewpoints, adapt content to users, analyze, group, infer and render indicators of the Telecom Valley situation. KMP relies on the integration of multiple components: databases for back-end persistence, web servers with JSP and servlets to provide front ends, and the CORESE semantic web server [Corby et al, 2004] to provide semantic web processing capabilities. Databases are used to store the different ontologies (e.g. ontology of technologies, of actions, of deliverables, of markets, of cooperation, etc.), the models (e.g. value chain of a telecom valley), and the users' data (e.g. descriptions of firms, research centers, competences, projects, etc.). Direct accesses and modifications of ontologies and other data are managed directly at the database level. Wrappers extract the relevant and authorized data from the databases and export them in RDF/S to feed CORESE as needed.

The platform integrates contributions coming from whole the Telecom Valley:

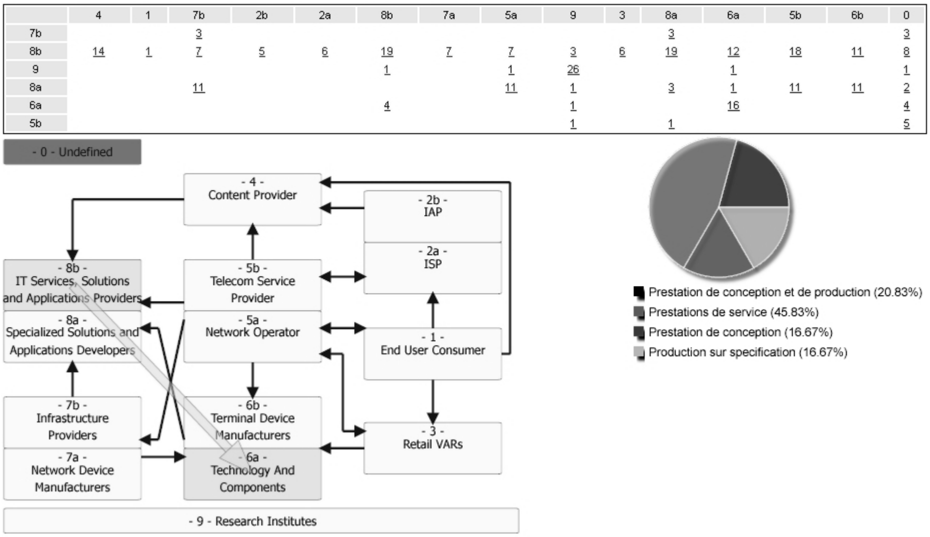
- several ontologies are populated and validated by multiple actors using interviews and brainstorming sessions animated by the local government administration[4].
- several sources of data are integrated: models provided by practitioners and researchers in management, descriptions of firms using industrial and economic markets vocabulary, description of research institutes using academic terms, etc.

The whole system relies on RDF, RDFS, and production rules [Corby et al, 2004] to describe the models and actors of the Telecom Valley. Exploiting this semantics the platform is able to:

- apply rules to enrich the different contributions and bridge the different viewpoints allowing a broad variety of queries and analysis to be run e.g. a set of rules generalize and group identical competences detailed in the profiles of the actors to provide statistics to researchers in management;

[4] http://www.telecom-valley.fr/index.php4?lang=ang

- exploit underlying models to propose graphic views of the Telecom Valley using XSLT to produce SVG rendering and combining on-the-fly models defined by the economists with data entered by the different actors; e.g. figure 1 shows an SVG interface to browse the value chain of the Telecom Valley and obtain statistics on the exchanges by clicking on the arrows. To each arrow is attached a query that CORESE solves against the RDF/S annotations of the Telecom Valley. For instance the screenshots shows statistics on the exchanges between two segments of the value chain (8b and 6a) and the distribution of these exchanges over the disjoint sub-classes of exchanges.
- apply complex query constructors to find partners, build consortiums, extract indicators, build statistics, sort and group results, find approximate answers, etc.
- apply clustering algorithms and produce graphic representations in SVG to allow institutional and industrial actors to get abstract views of the cartography of competences in the Telecom Valley;



	4	1	7b	2b	2a	8b	7a	5a	9	3	8a	6a	5b	6b	0
7b			3								3				3
8b	14	1	7	5	6	19	7	7	3	6	19	12	18	11	8
9						1		1	26			1			1
8a			11					11	1		3	1	11	11	2
6a						4			1			16			4
5b									1		1				5

- 0 - Undefined
- 4 - Content Provider
- 2b - IAP
- 2a - ISP
- 8b - IT Services, Solutions and Applications Providers
- 5b - Telecom Service Provider
- 8a - Specialized Solutions and Applications Developers
- 5a - Network Operator
- 1 - End User Consumer
- 7b - Infrastructure Providers
- 6b - Terminal Device Manufacturers
- 3 - Retail VARs
- 7a - Network Device Manufacturers
- 6a - Technology And Components
- 9 - Research Institutes
Prestation de conception et de production (20.83%)
Prestations de service (45.83%)
Prestation de conception (16.67%)
Production sur specification (16.67%)

Fig. 1. SVG view of exchanges on the value chain of the Telecom Valley of Sophia Antipolis

In this article we focus on one inference supported by the graph models underlying this semantic web repository: an ontology-based conceptual clustering providing a customizable and up-to-date cartography of competences available in the telecom valley. Section 2 briefly introduces an extract of the domain models and the users' requirements. Section 3 details the inferences underlying this representation, in particular the ontology-based metrics exploiting the semantic web graph structures. Section 4 concludes with the evaluation of this representation.

2 Model-Based Automated Cartography of Competencies

The first requirement and scenario of KmP is "*to acquire and give a broader visibility of the community of the technological pole*". As part of its answers, the platform provides a dynamic up-to-date cartography of the competencies available in the technological pole and grouped in clusters.

In KmP, the overall design methodology was oriented toward use and users. We relied on participatory design involving end-users, domain experts, information management experts and knowledge modeling experts. A large part of the specifications relied on mock-ups of interfaces built from the visual representations the users are used to. In particular, figure 2 shows a draft made by users when making explicit their notion of competences; it shows what they called a "*readable representation of the clusters of competencies in the technological pole*".

The first consequence of such a readability requirement is a set of expressivity requirements on the ontology. The current model used in the project relies on an ontology that consists of more than a thousand concept types and a hundred relations. Central to the modeling is the concept of "competence" used by the organizations when describing their profiles or the profile of the partners they are looking for. The model proposed by researchers in management and economics [Lazaric & Thomas, 2005] uses four facets to describe a competence and each facet is formalized in a part of the ontology. For instance, the competence "*designing microchips for the 3G mobile market using GSM, GPRS and UMTS*" is decomposed into four elements: an *action* (design); a *deliverable* (microchip); a *market* (3G mobile technology); a set of *resources* (GSM, GPRS, UMTS).

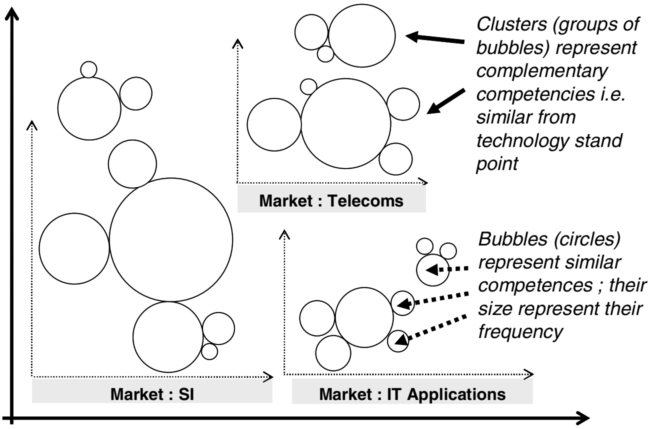

Clusters (groups of bubbles) represent complementary competencies i.e. similar from technology stand point
Market : Telecoms
Bubbles (circles) represent similar competences ; their size represent their frequency
Market : SI
Market : IT Applications

Fig. 2. Draft of a representation of clusters of competences

The second consequence of the readability requirement is the ability to simulate the inferences mobilized by the users when building this representation. The branch or the level of the ontology used to describe the situation is not always the same as the one used to display inference results. For instance, different users (e.g. industrialists vs. economists) may enter and use knowledge at different levels. In simple cases, we use rules close to Horn clauses to bridge these gaps. For the inferences behind the representation in Figure 2, the algorithm is much more complex and is a matter of conceptual clustering usually performed by economy and management analysts:

1. Analysts chose a market to which the analysis will be limited; all sub-types of this market will be considered, all ancestors or siblings will be discarded.
2. In this market, analysts group competences according to the *similarity* of their resources; a competence may have several resources (e.g. java, c, c++, project management) and one is chosen as the most representative (e.g. programming). This first grouping represents a *cluster*.
3. In each cluster, analysts group competences according to the *similarity* of their action (e.g. design) to form *bubbles*.

On the one hand we use ontology-based modeling to provide meaningful and dynamic representations (clusters as core competences of the technological pole) and on the other hand we need ontology-based inferences to automate this clustering (clusters as emergent structures in knowledge analysis). Questions associated to this problem include: what are the inferences underlying this representation? How can they be linked to semantic web models of the valley? How can we ensure that the clustering will be meaningful to the users?

In literature, the work on the formal side of the semantic web is largely influenced by the fact that logic-based languages are the most frequently used implementation formalisms. However, entailment is not the only product one should expect from a knowledge-based system, and the conceptual structures of the semantic Web can support a broad variety of inferences that goes far beyond logical deduction evening its simplest forms (RDF/S). Let us take the example of the class hierarchy which is considered to be the backbone of the RDFS schemata. The interpretation of the subsumption link is that the extension of a concept type (e.g. laptop) is a subset of the extension of another concept type (e.g. computer). What this logical implication hides is a graph structure that links the concept types through their *genus* and *differentia*. The graph structure of the semantic web formalisms supports inferences that go far beyond the set inclusion. The rest of this article shows how we designed such inferences to recreate the representation drafted in Figure 2 and how this specific example illustrates the richness of the underlying graph model of the semantic web.

3 Semantic Metrics to Visualize Knowledge

3.1 Semantic Metrics on the Ontological Space

The idea of evaluating conceptual relatedness from semantic networks representation dates back to the early works on simulating the humans' semantic memory [Quillian, 1968] [Collins & Loftus, 1975]. Relatedness of two concepts can take many forms for instance, functional complementarity (e.g. nail and hammer) or functional similarity (e.g. hammer and screwdriver). The latter example belongs to the family of semantic similarities where the relatedness of concepts is based on the definitional features they share (e.g. both the hammer and the screwdriver are hand tools). The natural structure supporting semantic similarities reasoning is the concept type hierarchy where subsumption links group types according to the characteristic they share. When applied to a semantic network using only subsumption links, the relatedness calculated by a spreading algorithm gives a form of semantic distance e.g. the early system of [Rada et al., 1989] defined a distance counting the minimum number of edges between two types.

We can identify two main trends in defining a semantic distance over a type hierarchy: (1) the approaches that include additional external information in the distance, e.g. statistics on the use of a concept; see for instance [Resnik, 1995] [Jiang & Conrath, 1997] (2) the approaches trying to rely solely on the structure of the hierarchy to tune the behavior of the distances [Rada et al., 1989][Wu & Palmer, 1994]. Including external information implies additional costs to acquire relevant and up-to-date information and furthermore, this information has to be available. Thus in a first approach we followed the second trend.

In the domain of Conceptual Graphs [Sowa, 1984], where the graph structure of knowledge representation is a core feature, a use for such a distance is to propose a non binary projection, i.e. a similarity $S:C^2 \rightarrow [0,1]$ where 1 is the perfect match and 0 the absolute mismatch. We used the CORESE platform provided by [Corby et al, 2004] to build our system. It is provided with an implementation of a depth-attenuated distance allowing approximate search. The distance between a concept and its father is given in (1):

$$dist\ (t, father\ (t)) = \left[\frac{1}{2} \right]^{depth\ (father\ (t))} \tag{1}$$

where *depth(t)* is the depth of *t* in the ontology i.e. the number of edges on the shortest path from *t* to the root. In the rest of the article we will only consider tree structures (not latices in general) and therefore there will be one and only one directed path between a concept and one of its ancestors; thus the distance is:

$$dist(t_1, t_2) = \frac{1}{2^{depth(LCST(t_1,t_2))-2}} - \frac{1}{2^{depth(t_1)-1}} - \frac{1}{2^{depth(t_2)-1}} \tag{2}$$

where $LCST(t_1,t_2)$ is the least common supertype of the two concept types t_1 and t_2.

3.2 Ontological Granularity and Detail Level

The representation in Figure 2 shows that the way market analysts usually group the competences correspond to what is called a monothetic clustering algorithm i.e. the different features of the competence are not combined in one distance but considered sequentially: first they chose the market sector they will limit their analysis to; second they chose the level of details at which the competences are to be grouped based on the resources they mobilize and form clusters; finally they chose a level of details for the actions and in each of the clusters previously obtained they group competences by types of actions to form bubbles in the clusters.

Limiting the competences to a given market sector is directly done by using the graph projection algorithm provided by CORESE: when one projects a query graph with a given market sector, by subsumption, only those competences with this market sector or a subtype of it will be retrieved. However the two other features (resources and action) require the ability to cluster competencies and to control the level of details of this clustering. The field of Data Clustering [Jain et al., 1999] studied this problem in great details and the typical structure built to control clustering details is a dendrogram: cutting a dendrogram at a given height provides a clustering at a corresponding granularity.

We already have a tree structure (the hierarchy of concepts) and a similarity measure (semantic similarities). However, the construction of a dendrogram relies on an ultrametric and the similarity measure defined between the classes does not comply with the definition of an ultrametric. Indeed, an ultrametric is a metric which satisfies a strengthened version of the triangle inequality:

$$dist(t1,t2) \leq max(dist(t1,t'), dist(t2,t')) \quad for\ any\ t'$$

Figure 3 gives a counter example where the distance defined in (2) violates this inequality.

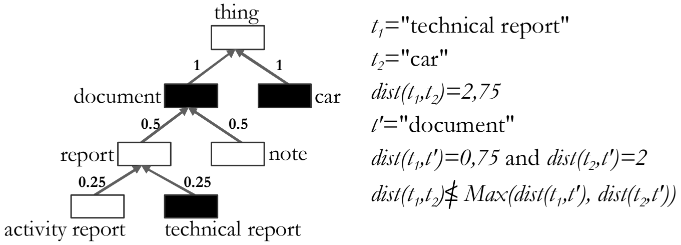


Fig. 3. Counter example the metric defined in (2) to be an ultrametric

The problem we then considered was a transformation of the ontological distance that would provide an ultrametric and transform the ontological tree into a dendrogram used to propose different levels of details in clustering the semantic annotations on competencies. A simple transformation would be to use a maximal distance that would only depend on the least common supertype of the two types compared:

$$dist_{MH}(t_1,t_2) = \max_{\forall t < LCST(t_1,t_2)} (dist(t,ST(t)) = \left[\frac{1}{2}\right]^{depth(LCST(t_1,t_2))} \tag{3}$$

where *ST(t)* is the supertype of *t*.

As shown in Figure 4, this transformation provides a dendrogram with levels of clustering that directly correspond to the levels of the ontology and therefore brings no added value compared to the direct use of the ontology depth to control the level of detail. In order to provide the users with a better precision in choosing the level of

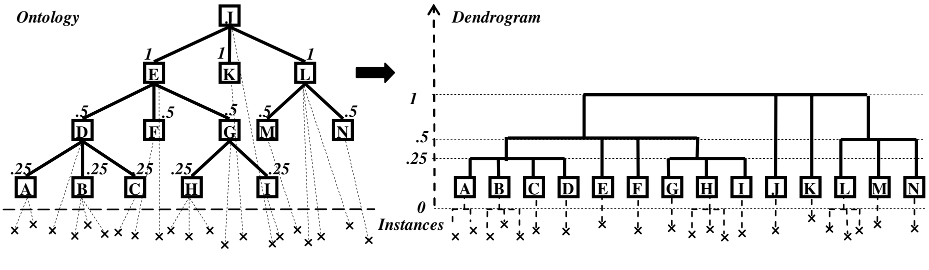


Fig. 4. Simple transformation using depth of LCST

details we needed a criterion to differentiate the classes and order their clustering. The distance given in (4) takes into account the depth of the hierarchy below the least common supertype.

$$\begin{aligned} dist_{CH}(t_1,t_2) &= \max_{\forall st \leq LCST(t_1,t_2)} \big(dist\big(st, LCST(t_1,t_2)\big)\big) \text{ when } t_1 \neq t_2 \\ dist_{CH}(t_1,t_2) &= 0 \text{ when } t_1 = t_2 \end{aligned} \tag{4}$$

where $st \leq LCST(t_1,t_2)$ means that st is a subtype of the least common supertype of t_1 and t_2.

Doing so, it allows us to differentiate between classes that already gather a number of levels of details and classes with a shallow set of descendants. Figure 5 shows the result of this transformation using the same initial ontology as Figure 4; we see a new level appeared that differentiates the classes L and E based on the level of details they already gather.

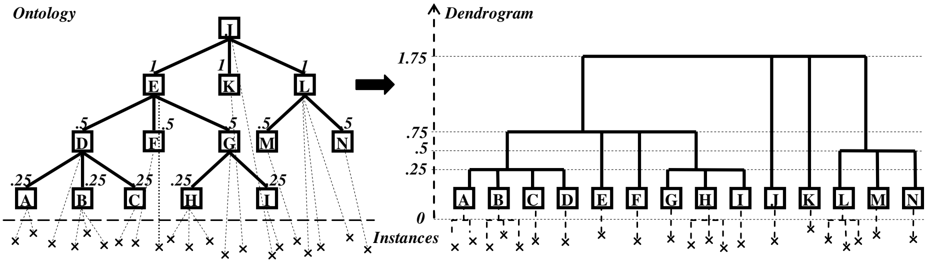


Fig. 5. Improved transformation using depth of descendants

To be precise, $dist_{MH}$ takes its values in (5):

$$E_{MH} = \left\{ \frac{1}{2^n}; 0 \leq n < D \right\} \cup \{0\} \tag{5}$$

where D is the maximal depth of the ontology.

Therefore the maximum number of levels in the dendrogram of $dist_{MH}$ is $Card(E_{MH})=d+1$. In comparison, $dist_{CH}$ takes its values in (6):

$$E_{CH} = \left\{ \sum_{i=m}^{n} \frac{1}{2^i}; 0 \leq m \leq n < D \right\} \cup \{0\} \tag{6}$$

Thus, at a given depth d the maximum number of levels is recursively defined by $NL(d) = NL(d-1) + d$ because an additional depth possibly adds one more arc to any path from the root. Since $NL(0)=1$, we can deduce that:

$$Card(E_{CH}) = NL(D) = \left(\sum_{n=1}^{D} n \right) + 1 = \frac{D(D+1)}{2} + 1 = \frac{D^2}{2} + \frac{D}{2} + 1 \tag{7}$$

Therefore, for a given maximal depth D we have

$$Card(E_{CH}) - Card(E_{MH}) = \frac{D^2}{2} - \frac{D}{2} = \frac{D}{2}(D-1) > 0 \text{ since D} > 1 \tag{8}$$

Thus $dist_{CH}$ generates more levels than $dist_{MH}$ and the difference is upper-bounded by the square of D. Now we need to prove that $dist_{CH}$ is an ultrametric:

(a) By definition $dist_{CH}(t,t) = 0$ see (4)

(b) Let us show that $dist_{CH}(t_1,t_2) = dist_{CH}(t_2,t_1)$

$$dist_{CH}(t_1,t_2) = \max_{\forall st \le LCST(t_1,t_2)} \left(dist\left(st, LCST(t_1,t_2)\right)\right) = \max_{\forall st \le LCST(t_2,t_1)} \left(dist\left(st, LCST(t_2,t_1)\right)\right) = dist_{CH}(t_2,t_1)$$

This is because $LCST(t_1,t_2)=LCST(t_2,t_1)$ i.e. the least common supertype of t_1 and t_2 is also the least common supertype of t_2 and t_1.

(c) Let us show that $dist_{CH}(t_1,t_2) = 0 \Rightarrow t_1=t_2$: if $t_1 \neq t_2$ we have

$$dist_{CH}(t_1,t_2) = \max_{\forall st \le LCST(t_1,t_2)} \left(dist\left(st, LCST(t_1,t_2)\right)\right) \ge dist\left(t_1, LCST(t_1,t_2)\right) + dist\left(t_2, LCST(t_1,t_2)\right) > 0$$

So the only way to have $dist_{CH}(t_1,t_2) = 0$ is when $t_1=t_2$

(d) Let us show that $\forall t'\ dist_{CH}(t1,t2) \le max(dist_{CH}(t1,t'), dist_{CH}(t2,t'))$ (strengthened triangle inequality)

If $t_1=t_2$ then $dist_{CH}(t_1,t_2) = 0$ and the inequality is verified.

If $t' \le t_1$ and $t_1 \not\le t_2$ and $t' \not\le t_2$ then

$$dist_{CH}(t',t_2) = \max_{\forall st \le LCST(t',t_2)} \left(dist\left(st, LCST(t',t_2)\right)\right) = \max_{\forall st \le LCST(t_1,t_2)} \left(dist\left(st, LCST(t_1,t_2)\right)\right)$$

since if $t' \le t_1$ then $LCST(t',t_2)=LCST(t_1,t_2)$ or more generally the least common supertype for t_1 and t_2 is the same as the one of the subtypes of t_1 and t_2 since we are in a tree.

If $t' \le t_2$ and $t_2 \not\le t_1$ and $t' \not\le t_1$ the same reasoning applies *mutatis mutandis.*

If $t_1 \le t_2$ and $t' \le t_2$ then $LCST(t',t_2)=t_2$ and $LCST(t_1,t_2) = t_2$ so $dist_{CH}(t_2,t') = dist_{CH}(t_1,t_2)$. If $t_2 \le t_1$ and $t' \le t_1$ the reasoning is, mutatis mutandis, the same.

If $t' \not\le t_2$ and $t' \le t_1$ then $LCST(t_1,t_2) \le LCST(t',t_1)$ or $LCST(t_1,t_2) \le LCST(t',t_2)$ otherwise we would have $LCST(t_1,t_2) > LCST(t',t_1)$ and $LCST(t_1,t_2) > LCST(t',t_2)$ and since $t_1 \neq t_2$, $t' \le t_2$ and $t' \le t_1$, it would require t' or one of its ancestors to have two fathers which is impossible in a tree. So if $LCST(t_1,t_2) \le LCST(t',t_1)$ then $dist_{CH}(t_1,t_2) \le dist_{CH}(t_1,t')$ since $\{st\ ;\ st \le LCST(t_1,t_2)\} \subset \{st\ ;\ st \le LCST(t',t_1)\}$. Likewise if $LCST(t_1,t_2) \le LCST(t',t_2)$ then $dist_{CH}(t_1,t_2) \le dist_{CH}(t_2,t')$ since $\{st\ ;\ st \le LCST(t_1,t_2)\} \subset \{st\ ;\ st \le LCST(t',t_2)\}$. Thus, in both cases the inequality is verified. Therefore, we covered all the cases and $dist_{CH}$ is an ultrametric that can be used to produce a range of levels of details exploitable in widgets for interfaces.

The maximal distance $dist_{Max}$ between two sister classes of depth d in an ontology of maximal depth D is

$$dist_{Max}(d,D) = \sum_{i=d-1}^{D} \left[\frac{1}{2}\right]^i = \frac{1}{2^{d-1}} \sum_{i=0}^{D-d+1} \frac{1}{2^i} = \frac{1}{2^{d+1}} \times \frac{1-\frac{1}{2^{D-d+2}}}{\frac{1}{2}} = \frac{1}{2^{d-2}}\left(1-\frac{1}{2^{D-d+2}}\right) = \frac{1}{2^{d-2}} - \frac{1}{2^D}$$

The minimal distance between two sister classes of depth d is $dist_{Min}(d) = 1/2^{\ d-1}$. Therefore $dist_{Max}(d+1) < dist_{Min}(d)$ i.e. the clustering of classes respects the ontology hierarchy and a class cannot be clustered before its descendants. However between two sister classes, the children of a shallow class will be grouped before the children of a class with a deep descendant hierarchy. Finally since the clustering follows the class hierarchy, a name can be given to every cluster *Cl* and it is very important to produce a meaningful clustering: *Name(Cl) = Name (LCST({t ; type t* $\in$ *Cl}))*

3.3 Ontology-Based Queries to Form Clusters

Using our transformation of an ontological distance into an ultrametric, we create two dendrograms respectively for the ontology of resources and the ontology of actions. Each dendrogram supports a widget (e.g. scrollbar) allowing the user to chose a clustering levels detail respectively for resources and actions.

To choose a level to cut the dendrograms amounts to select a number of classes that can be used to differentiate competences during the clustering: every class visible at this level may be used to describe a competence. Therefore the two levels of detail chosen for resources and actions result in two sets of classes of resources and actions that have to be considered. Based on these sets and the market sector the user chose, we generate all the combinations of queries that cover all the combinations of resources and actions; thus each one of these queries corresponds to a potential bubble, and the bubble will be shown if there is at least one answer to this query. To consider a competence once and only once, the queries exclude the subclasses that have not been collapsed *i.e.* the subclasses that are above the detail level and for which there will be a dedicated query. Each query is submitted to the CORESE search engine to retrieve and count instances of competences falling in the corresponding bubble.

As we mentioned, there might be several resources for a competence and the analyst chooses the most representative one. For each competence, this inference is simulated by considering the classes of resources available at the chosen level of details and by sorting them according to the number of instances of resources they cover for this competence. For instance if a competence uses *java*, *c*, *c++*, and *project management* as resources and the level of details include classes like *management theory*, *programming language* and *mathematic models*, these classes will be sorted as follows: *programming language* (3 instances), *management theory* (1 instance), *mathematic models* (0 instance). Therefore the most representative resource type picked will be *programming language* and this competence will be counted in a cluster on *programming language*. This process is illustrated in figure 6.

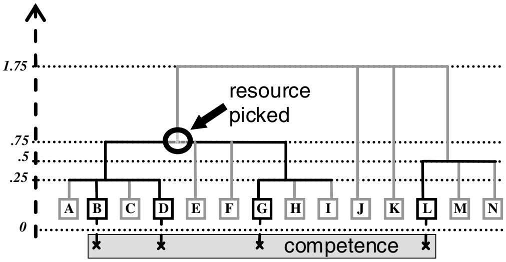


Fig. 6. Choosing the most representative resource

The final result is a list of bubbles grouped in clusters. The last problem is the display of these bubbles and clusters in an intelligent and intelligible fashion.

3.4 Conceptual Classification and Spatial Grouping

Users are interested in two aspects when comparing clusters: their size and the type of resource they use. They combined these two dimensions to obtain what they called a "radar view" of the technological pole that they draft in Figure 2. The radar view uses angular coordinates: the angle is derived from the place of the resource classes in the ontology and the radius from the size of the cluster. Figure 7 shows two opposite approaches in using the ontology for angular positions: a top-down division where the children equally share the angle allocated to their parent (left part of the figure); a bottom-up merging where leaves equally share the available angle and parents 'inherit' the sum of the angles of their children (right part of the figure).

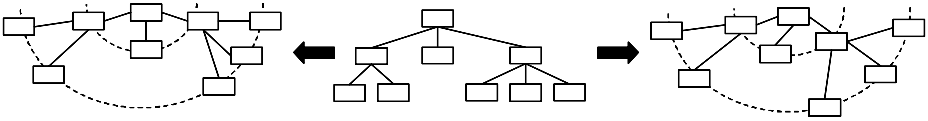


Fig. 7. Using the ontology for angular positions

Here the initial angle was 180°. On the left (top-down division) it was divided into three for the first level of children (60° each) and then the first and last thirds were respectively divided into two (30°) and three (20°) for the second descendants. On the right (bottom-up merging), the 180° were divided into 6 for the leaves of the tree (30° each) then the nodes of the first level are respectively allocated angles of 60°, 30° and 90°. The top-down division maintains the equality between brothers and favors the structure of the upper modules while the bottom-up merging maintains the equality between leaves and favors the detailed branches.

In our case the top-down division was more interesting since it divides the space equally between the main domains of resources and, as shown in Figure 8, the ontology is much more detailed in some parts (e.g. the computer resources) than in some others (e.g. management resources). On such an ontology, bottom-up merging would bias the view while the top-down division applied on 360° in Figure 8 maintains an equal angle for brothers; this is used to have an egalitarian positioning of the clusters based on their representative resource. Figure 9 shows the result of this approach applied to the list of clusters obtained with the inference previously described: the angular position is given by the place of the representative resource in the ontology, and the radius corresponds to the size of the cluster. As a result we can see that the activity of the technological pole is primarily focused in a given sector on which Figure 10 provides a partial zoom.

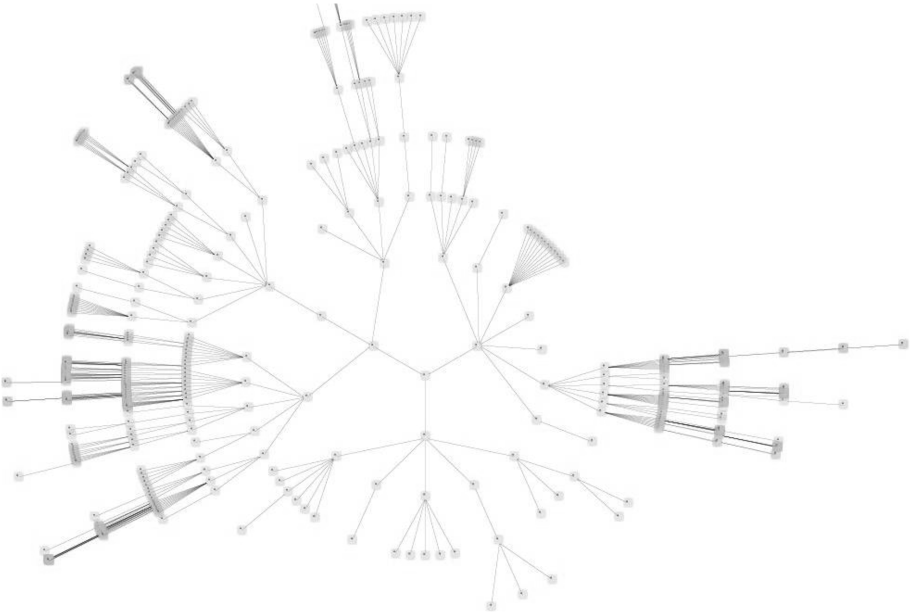


Fig. 8. Top-down division of the ontology applied on 360°

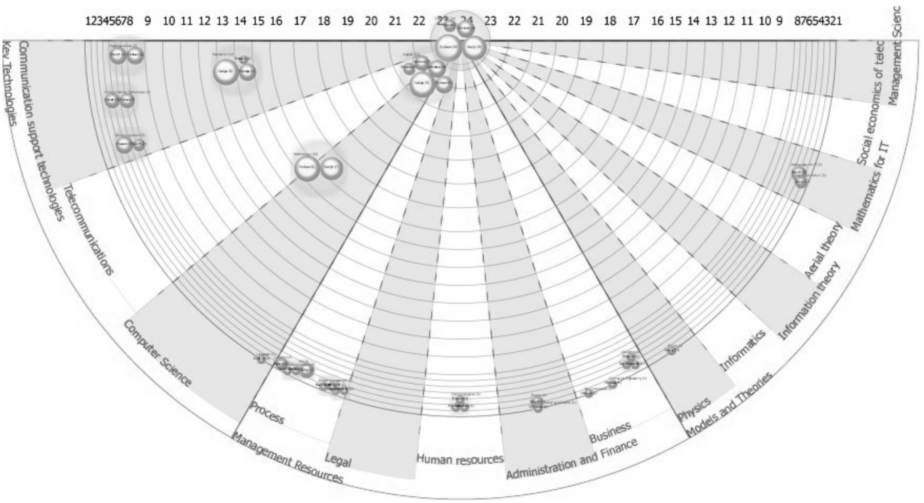


Fig. 9. Radar view of the clusters on 180°

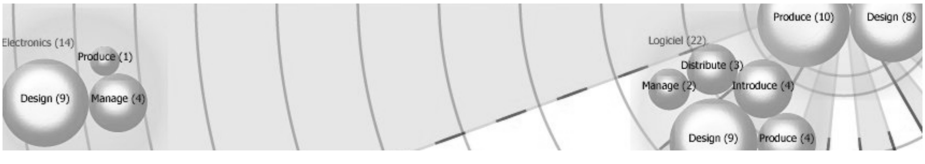


Fig. 10. Zoom on the Radar view in figure 8

4 Evaluation and Conclusions

The two types of evaluations were carried out: (1) usability and ergonomics; (2) complexity and real time. The usability and ergonomic evaluation triggered several evolutions of the interfaces (including two major reengineering) and some tests are still being carried out using different techniques (thinking aloud, video analysis, questionnaires, etc.). We are currently carrying out the second iteration of the usability and ergonomics studies. Needs for redesign have already been recorded, for instance: (a) a need to provide simple widgets to select the levels of details (b) a need to use statistics on the instances instead of the ontology, to calculate the angular position, thus accounting for the effective use of concepts rather than a theoretical importance (c) a need to have an idea of the time it will take to compute a clustering view. One of the major points is that the acceptance of KmP was so effective that we are now moving from a group of 20 pilot user companies to a park of 70 user companies and more and more public research centers are describing themselves, while the project is entering an industrialization phase. We also abstracted a methodology to involve users and keep them involved in the design of an ontology-based application [Giboin et al, 2005].

Concerning complexity and real time, there are two phases in the inference we detailed here: the initialization of the dendrograms and tree of angles (which is done once) and the calculation of the clusters and their position (which is done each time a user submits a query). The complexity of the algorithm for the initialization of the dendrograms and tree of angles breaks down as follows, where n and m are the number of classes respectively in the ontologies of resources and actions:

- Parsing the schema to build the tree: $O(n+m)$
- Initializing the depth and angular distribution: $O(n+m)$
- Sorting the dendrogram: $O(n\times log(n)+ m\times log(m))$

The size of the set of queries produced to build the clusters depends on the level of details chosen by the users. For two levels of details n' and m' respectively chosen in the dendrograms of resources and actions we have to solve $n'\times m'$ queries and so the worse case is $n*m$ queries. We carried out a number of real time tests using a small configuration (Pentium 4 M / 1.7GHz / 512 Mo running MS Windows):

- The average minimum time (for $n'=1$ and $m'=1$ and thus 1 query/potential cluster) is 86 milliseconds.
- The average maximum time ($n'=596$ and $m'=118$ thus 70 328 queries/potential clusters) is 11 minutes.
- The average typical time ($n'=109$ and $m'= 9$ thus 981 queries/potential clusters) is 9 298 milliseconds i.e. roughly 9 seconds.

The notion of typical query is due to the fact that the level of details used by users of the radar view is much lower than the level of details provided by the ontology and used by the end-user companies to describe themselves. For instance when market analysts are interested in activities of the telecom valley that involve programming in general, they do not want to differentiate between java, c++, c#, *etc.*

Figure 11 shows the behavior of the response time against the level of details provided by the possible combinations of n' and m' and Figure 12 shows the behavior of the response time against the actual value of $n'\times m'$ that is to say the number of generated queries. The linear regression of Figure 12 approximately corresponds to $y= x\times 8.42+89.24$ and it has two very useful applications: (a) it provides a very simple and rapid way to foresee and warn the users about the time a clustering request will take to be solved before they actually submit it, and (b) it predicts the level of details above which it is better to rely on a reporting functionality that prepares a set of views in batch mode at night rather than relying on a real-time calculation; typically above 15 seconds of response time *i.e.* above $n'\times m'$=2000 in our case.

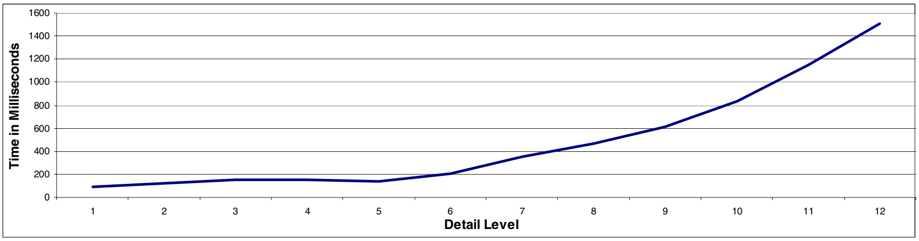


Fig. 11. Response time against detail level

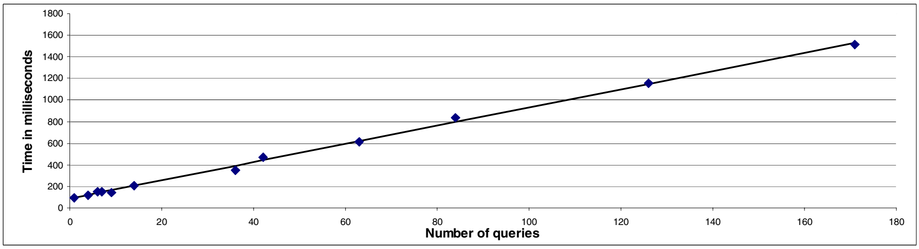


Fig. 12. Response time against number of queries

KmP now includes a set of views and analysis tools as the radar view we detailed in this article. This semantic Web portal provides indicators for institutional organizations to understand the landscape of the technological pole, and for actors to find opportunities, niches, partners, *etc.* By detailing the work done on one RDFS-based graph inference, we showed how the graph structure of semantic web formalisms can be exploited in new inferences to support intelligent interfaces in bridging the gap between the complexity of these underlying conceptual models and the ergonomic constraints of end-users' interfaces and daily concerns. The inferences at play here can be reused to support other functionalities and of course other inferences exist on these graph structures; in fact the algorithm has already been reused to produce other clustering view such as the identification of competency poles. Here, we proved the characteristics of the metrics and inferences we proposed and we illustrated the interpretation that can be associated to their results. In parallel we are conducting an experiment to evaluate and compare these simulated metrics with the ones humans naturally use in handling information [Gandon et al., 2005].

References

[Collins & Loftus, 1975] Collins, A., Loftus, E., A Spreading Activation Theory of Semantic Processing. *Psychological Review*, vol. 82, pp. 407-428, 1975

[Corby et al, 2004] Corby, O., Dieng-Kuntz, R., Faron-Zucker, C., Querying the Semantic Web with the Corese Search Engine, In Proc. of *European Conference on Artificial Intelligence*, IOS Press, pp.705-709, 2004

[Gandon et al., 2005] Gandon F., Corby O., Dieng-Kuntz R., Giboin A., Proximité Conceptuelle et Distances de Graphes, To be published in Proc. Raisonner le Web Sémantique avec des Graphes, Nice, Journée thématique de la plate-forme AFIA, Nice, 2005

[Giboin et al, 2005] Giboin A., Gandon F., Gronnier N., Guigard C., Corby, O., Comment ne pas perdre de vue les usage(r)s dans la construction d'une application à base d'ontologies ? Retour d'expérience sur le projet KmP, To be published in Proc. Ingénierie des Connaissances, plate-forme AFIA, p133-144, 2005

[Jain et al., 1999] Jain, A.K., Murty, M.N., and Flynn, P.J. (1999): Data Clustering: A Review, *ACM Computing Surveys*, Vol 31, No. 3, 264-323.

[Jiang & Conrath, 1997] Jiang, J., Conrath, D., Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In Proc. of *International Conference on Research in Computational Linguistics*, Taiwan, 1997

[Lazaric & Thomas, 2005] Lazaric N., Thomas C., "The coordination and codification of knowledge inside a network, or the building of an 'epistemic community': The 'Telecom Valley' case study" in "Reading the Dynamics of a Knowledge Economy », to be published in 2005 by Edward Elgar Publishing.

[Quillian, 1968] Quillian, M.R., Semantic Memory, in: M. Minsky (Ed.), *Semantic Information Processing*, M.I.T. Press, Cambridge, 1968.

[Rada et al., 1989] Rada, R., Mili, H., Bicknell, E., Blettner, M., Development and Application of a Metric on Semantic Nets, *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 19(1), pp. 17-30, 1989.

[Resnik, 1995] Resnik, P., Semantic Similarity in a Taxonomy: An Information-Based Measure and its Applications to Problems of Ambiguity in Natural Language. In *Journal of Artificial Intelligence Research*, vol 11, pp. 95-130, 1995

[Sowa, 1984] Sowa., J.F., *Conceptual structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, Massachusetts, 1984