Summarizing Ontology-based Schemas in PDMS

Carlos Eduardo Pires[#], Paulo Sousa[*], Zoubida Kedad[+], Ana Carolina Salgado[*]

[#]*Computer Science Department, Federal University of Campina Grande (UFCG)*
Av. Aprígio Veloso, 882, Bodocongó - 58109-970 - Campina Grande - PB - Brazil
cesp@dsc.ufcg.edu.br

[*]*Center for Informatics, Federal University of Pernambuco (UFPE)*
Av. Professor Luís Freire, s/n, Cidade Universitária - 50740-540 - Recife - PE - Brazil
povqs@cin.ufpe.br
acs@cin.ufpe.br

[+]*Université de Versailles Saint-Quentin-en-Yvelines (UVSQ)*
45 Avenue des Etats-Unis, 78035 Versailles, France
zoubida.kedad@prism.uvsq.fr

Abstract - **Quickly understanding the content of a data source is very useful in several contexts. In a Peer Data Management System (PDMS), peers can be semantically clustered, each cluster being represented by a schema obtained by merging the local schemas of the peers in this cluster. In this paper, we present a process for summarizing schemas of peers participating in a PDMS. We assume that all the schemas are represented by ontologies and we propose a summarization algorithm which produces a summary containing the maximum number of relevant concepts and the minimum number of non-relevant concepts of the initial ontology. The relevance of a concept is determined using the notions of centrality and frequency. Since several possible candidate summaries can be identified during the summarization process, classical Information Retrieval metrics are employed to determine the best summary.**

I. INTRODUCTION

Quickly understanding the content of a data source is very useful in several contexts, ranging from querying a single database to automatically managing large-scale data integration systems. A complex database might be difficult to be understood and a user querying this database might be interested in a summary providing the main information it contains. In data integration systems, a summary of a data source schema could be useful to comprehend its content and to improve some automated integration tasks such as schema comparison: comparing source summaries instead of the whole source schemas could contribute to reduce the cost of the overall process. Summarizing databases has recently been the focus of some research works such as [1] [2]. In this paper, we present an approach for summarizing the schemas of peers participating in a Peer Data Management System (PDMS) [3].

In PDMSs, each peer is an autonomous source that makes available a local schema. In some PDMSs [4][5], peers are semantically clustered in the overlay network according to their local schema. Each cluster of peers (cluster, for short) has a schema which is used as a semantic representation to provide a common understanding of the terms that are being shared inside the cluster. In practice, the cluster schema is obtained by merging the local schema of several peers [4]. When a peer joins the system, is has to be assigned to the most appropriate cluster. This is done by comparing the schema of the incoming peer against the schemas of all current clusters, which is performed by a matching service [6]. To support this task, we propose the use of schema summarization techniques [1][2][7] to produce a succinct version of each cluster's schema. For each comparison of an incoming peer's schema and a cluster's schema, instead of comparing the whole schemas, only the summarized representation will be used.

Creating a good summary is a non-trivial task. Ideally, the summary should be concise enough for incoming peers to allow the quick understanding of the cluster, yet it needs to convey enough information in such a way that the incoming peer can obtain a decent understanding of the cluster. Manual summarization in a PDMS setting is labor-intensive and impractical especially in situations where a high number of clusters need to be summarized. In addition, manual summary generation might cause that the summary will not be updated when the cluster schema is modified, resulting in an outdated and misleading summary [1]. The use of representative summaries can improve scalability and consistency as well as minimize computation efforts.

The goal of this work is to propose an automatic process to summarize an integrated schema representing multiple local schemas. We assume that all the schemas are represented by ontologies [8], which have become one of the most common ways of expressing knowledge in different distributed and opened applications such as PDMSs. An ontology summary corresponds to a subontology of the initial ontology under a specified size. The summary should contain the maximum number of relevant concepts and the minimum number of non-relevant concepts of the initial ontology. Unlike the existing approaches [1][2], the identification of relevant concepts is based not only on the notion of centrality [9] but also on the notion of frequency (i.e., the number of occurences of a concept in local schemas). These two combined criteria are used to measure the relevance of concepts. Since several possible candidate summaries can be identified during the summarization process, classical Information Retrieval metrics [10] are employed to determine the best summary.

The main contributions of this work are: (i) the definition a summarization service for PDMS; (ii) an original definition of the relevance of a concept combining both centrality and

frequency; (iii) an algorithm for building a cluster's summary; and (iv) an evaluation of the proposed summarization algorithm. The paper is organized as follows: Section 2 provides an overview of the proposed summarization process. Section 3 describes the metrics used to determine the relevance of the elements in a schema. Section 4 presents the summarization process, the algorithm, and an illustrative example. Section 5 presents the experiments. Related work is discussed in Section 6 and finally, Section 7 presents our conclusions and suggestions for further research.

II. General Overview

In our context, each peer is described by a local schema, i.e., a local ontology O_n. Each cluster of peers is associated to an ontology O which is the integration of the ontologies $O_1,\ldots O_n$ (local schemas) describing the peers belonging to that cluster. The ontologies are expressed in formal languages with a well-defined semantics such as the Web Ontology Language (OWL) [11]. As illustrated in Figure 1, the proposed summarization process consists in, given an initial ontology O, generating an abridged version of O, named ontology summary (denoted OS). The relevant concepts of O (depicted in grey color) are initially identified and OS corresponds to the subontology of O concentrating the maximum number of relevant concepts. Since relevant concepts can be non-adjacent in O, non-relevant concepts (white color) may be also introduced in an OS. Such "undesired" concepts are needed to maintain the original relationships among relevant concepts. If the relevant concepts are simply identified and added to an ontology summary (ignoring their relationships), then a human intervention would be necessary to (re)link them. Therefore, OS also corresponds to the subontology of O containing the minimum number of non-relevant concepts.

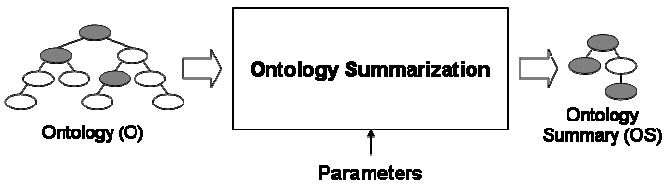

Fig. 1 An overview of the proposed ontology summarization process

We use a graph notation to represent an OWL ontology O, which is modeled as a connected directed labeled graph $O = (C, R)$, where $C = \{c_1,...,c_n\}$ is a finite set of vertices (concepts) and $R = \{r_1,...,r_n\}$ is a finite set of edges (relationships between concepts). A relationship $r_k \in R$ represents a directed relation between two adjacent concepts c_i and $c_j \in C$; i.e., $r_k = (c_i \times c_j)$. Two concepts $c_i, c_j \in C$ are adjacent in O if $\exists\ r_k \in R$ / $r_k = (c_i \times c_j)$ or $r_k = (c_j \times c_i)$. A directed labeled edge is defined from c_i to c_j if c_i is a direct subconcept of c_j. Similarly, if c_i is a domain concept and c_j its range concept then a directed labeled edge is added from c_i to c_j. The number of concepts in C indicates the size of O, denoted $|O|$. Similarly, we define an ontology summary OS as a proper subgraph of O since $OS \subset O$. Thus, the same notation is valid for OS. Formally, $OS = (CS, RS)$, where $CS \subset C$ and $RS \subset R$.

III. Relevance Measures

The relevance of an ontology concept c_n is measured considering the relationships of c_n with other concepts in an ontology O (*centrality*) and the occurrences of c_n in local ontologies $O_1,\ldots,O_n$ that compose O (*frequency*). In our approach, centrality is used to capture the importance of a given concept within an ontology, whilst frequency is used when an ontology results from an integration process and captures the occurrences of a concept in the set of underlying local ontologies.

A. Centrality Measure

Centrality [9] is one of the most important ways to identify relevant vertices within a graph. The most widely used centrality measures are: degree, closeness, and betweenness. The *degree centrality* [9] is based on the idea that a vertice v with a large number of links to other vertices has wider and more efficient access to the other vertices in the graph. The other two centrality measures are based on the notion of graph paths [12]. A path in a graph is a sequence of consecutive edges. A geodesic path is the shortest path, in terms of number of edges traversed, between two vertices. The *closeness centrality* [9] of v means the geodesic distance between v and all its reachable vertices. The *betweenness centrality* [9] of v is the number of geodesic paths between other vertices that v falls on.

In this work, we extend the original definition of the degree centrality measure not only to consider the number of relationships between ontology concepts but also the types of relationships between them. In this light, two types of relationships are identified: *standard* and *user-defined*. A standard relationship is one of the followings: *is-a*, *part-of*, and *same-as*. The semantics of a user-defined relationship is specified by the user and is domain-dependent, e.g. *hasItems* or *authorOf*. The normalized formula for the extended degree centrality of a concept c_n is:

$$centrality(c_n) = \frac{nr \times \left(\frac{n_s \times w_s}{max_s} + \frac{n_{ud} \times w_{ud}}{max_{ud}} \right)}{|C| - 1}$$

where n_s and n_{ud} are respectively the number of standard and user-defined relationships maintained by c_n. Note that, if c_n maintains more than one relationship with another concept, it is counted only once. w_s and w_{ud} are respectively the weights of the standard and user-defined relationships. max_s and max_{ud} represent respectively the maximum number of standard relationships and the maximum number of user-defined relationships held by a concept in the considered ontology. nr represents the number of distinct concepts with which a concept c_n maintains relationships. In addition, (i) centrality(c_n) $\in [0,1]$; (ii) $w_s + w_{ud} = 1$; and (iii) $n_s + n_{ud} = n_r$.

B. Frequency Measure

Frequency is a measure that can be used when the ontology to be summarized is an integrated ontology obtained as a result of merging several local ontologies $O_1,\ldots,O_n$. Ontology merging [13] is the process in which two (or more) local

ontologies are merged into one target ontology. In general, the local ontologies remain, along with correspondences between the elements of the merged ontology and the elements of each local ontology. In [14], different types of ontology correspondences are defined, e.g. equivalence and subsumption. Table I describes concept correspondences in the PDMS SPEED [4]. For instance, *Faculty* in the target ontology *O* is identified as: (i) equivalent to *Faculty* in the local ontology O_1; (ii) sub-concept of *Worker* in O_2; and (iii) super-concept of *Professor* in O_3 and *PostDoc* in O_4.

TABLE I
EXAMPLES OF CONCEPT CORRESPONDENCES [14]

Correspondences for the concept O:Faculty	
O:Faculty $\equiv$ O_1:Faculty	O:Faculty $\sqsupset$ O_3:Professor
O:Faculty $\sqsubset$ O_2:Worker	O:Faculty $\sqsupset$ O_4:PostDoc

In this work, since we assume that *O* can be a merged ontology then a concept $c_n \in C$ corresponds to one or more concepts contained in $O_1,\ldots,O_n$. In this sense, the frequency of c_n is defined as the ratio between the number of concept correspondences involving c_n (denoted $|correspondences(c_n)|$) and the number of distinct local ontologies (denoted $|O_1,\ldots,O_n|$). Both information can be extracted from the ontology correspondences. Formally,

$$frequency(c_n) = \frac{|correspondences(c_n)|}{|O_1,\ldots,O_n|}$$

where $frequency(c_n) \in [0,1]$. Given the correspondences illustrated in Table I, the concept *Faculty* is involved in four correspondences. Assuming that the number of local ontologies is also four then $frequency(Faculty) = 1.0$.

In a PDMS scenario, ontology summaries are used to resume the elements shared in a cluster of peers and, consequently, to improve the efficiency of peer clustering. The use of frequency as a measure to determine the relevance of concepts is motivated by two facts: (i) when an incoming peer joins a cluster it will automatically locate several other semantically related peers; and (ii) the majority of the peers within the cluster share the same elements of the incoming peer.

IV. BUILDING AN ONTOLOGY SUMMARY

The main steps of the ontology summarization process are: (i) calculate the relevance of ontology concepts; (ii) determine the relevant concepts; (iii) group adjacent relevant concepts; (iv) identify paths between groups of concepts; (v) analyze the identified paths; and (vi) determine the ontology summary.

1) Calculate concept relevance: our proposal to combine centrality and frequency consists in using a weighted formula in which the weights are defined according to the importance of each measure. The following formula is used to calculate the relevance of a particular concept c_n in an ontology *O*:

$$relevance(c_n) = \lambda \times centrality(c_n) + \beta \times frequency(c_n)$$

where $relevance(c_n) \in [0,1]$ and $\lambda + \beta = 1$.

2) Determine relevant concepts: this step consists in identifying the set of relevant concepts (denoted *RC*, where $RC \subseteq C$) of an ontology *O*. Ideally, the concepts in the identified set should be contained in the ontology summary *OS*. Several options can be used to determine *RC*. We can consider that *RC* has a fixed size *k* and select the top *k* concepts. Alternatively, we can include in *RC* the concepts for which the relevance is above a certain threshold value informed by the user:

$$\forall c_n \in C,\ if\ relevance(c_n) \geq relevance\ threshold \Rightarrow c_n \in RC$$

3) Group adjacent relevant concepts: this step consists of forming groups of concepts containing only relevant concepts which are adjacent in the initial ontology *O*. Such groups are created in order to facilitate the identification of paths between relevant concepts (explained in Step 4). When building groups of concepts the following situations can occur: (i) each group is formed by a single relevant concept (all relevant concepts are non-adjacent in *O*); (ii) at least one of the groups has more than one relevant concept (some relevant concepts are not adjacent in *O*); and (iii) only one group is formed, containing all the relevant concepts. In the first two situations, the summarization process proceeds with Steps 4, 5 and 6. In the last situation, the summarization process finishes and the ontology summary corresponds to the identified group of concepts.

4) Identify paths between groups of concepts: if there are at least two groups of concepts in the initial ontology *O* (situations i and ii of Step 3), all paths between groups of concepts in *O* are detected. Each group of concepts is treated as a single concept. Given two groups of concepts G_1 and G_2, each path is denoted $path_i = c_1 \rightarrow c_2 \rightarrow \ldots \rightarrow c_{n-1} \rightarrow c_n$, where each c_i is a non-relevant concept in the initial ontology *O*. This path is such that: (i) $c_1 = G_1$; (ii) $c_n = G_2$; (iii) $\forall\ i \neq 1$ and $i \neq n \rightarrow c_i \notin G_1 \cup G_2$; and (iv) $path_i$ has no cycles. Given this definition, this step consists in enumerating all the possible paths between each pair of groups. Since multiple paths between two groups of concepts can be detected, to reduce computational complexity the step is restricted to the enumeration of the *k*-first paths satisfying the requested summary size (i.e., $|RC|$).

5) Analyze identified paths: multiple paths between groups of concepts can be identified. The classical metrics *recall* and *precision*, commonly used in Information Retrieval [10], are employed to determine the level of coverage and conciseness of each path, respectively. Recall means that a path should be an extraction of *O* reflecting as many relevant concepts as possible. Precision expresses whether a path is succinct enough to facilitate an analysis of the entire ontology *O*.

$$Recall = \frac{|Path_i \cap RC|}{|RC|} \qquad Precision = \frac{|Path_i \cap RC|}{|Path_i|}$$

Paths cannot be compared based solely on precision and recall. The path which has high recall may have a low precision and vice-versa. For this purpose, *f-measure* [10] is used to aggregate precision and recall.

$$f-measure = \frac{Precision \times Recall}{(1-\alpha) \times Precision + \alpha \times Recall}$$

6) Determine the ontology summary: the selection of the best candidate path is done according to: (i) *f-measure*: the

path should be the one having the maximum number of relevant concepts and the minimum number of non-relevant concepts. In other words, the path with the highest value of f-measure should be selected; (ii) *average relevance*: if two distinct paths with the same value of f-measure are identified, the path with the highest average relevance should be preferred. The average relevance of a path is the ratio between the sum of the individual concept relevance in a path and the number of concepts in the same path.

Figure 2 depicts the proposed summarization algorithm. It accepts the ontology to be summarized *O* (mandatory), a set of ontology correspondences *Co* (optional), and a set of parameters *P* (mandatory). If the correspondences are not informed, only centrality is used to calculate the relevance of concepts. To meet diverse application types, the algorithm can accept various parameters. Depending on the parameter values that are provided, different ontology summaries *OS* can be generated for the same ontology *O*.

```
SummarizeOntology (input: Ontology; input: Co; input: P; output: OS)
{
CalculateConceptRelevance(Ontology, λ, β, centrality measure, Co);
RC ← DetermineRelevantConcepts(Ontology, relevance criteria);
Ontology ← GroupAdjacentRelevantConcepts(Ontology);
If Ontology.Groups = 1 and RC ⊆ Ontology.Groups[1].Concepts then
   OntologySummary ← Ontology.Groups[1];
Else
   Paths ← IdentifyPaths(Ontology, Δ);
   AnalyzePaths(Paths, α);
   OntologySummary ← GetBestPath(Paths);
End if
Return(OntologySummary);
}
```

Fig. 2 The ontology summarization algorithm

As an example, consider a public ontology (*networkA*) describing nodes in a local area network (Figure 3). Assume that an ontology summary containing 6 concepts with size variation of 1 must be generated. To simplify matters, only centrality is used to determine the relevance of concepts. Assume that recall and precision have equal importance (α=0.5).

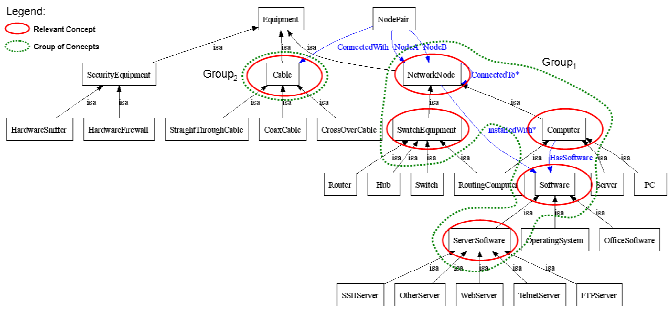

Fig. 3 The networkA ontology

Considering these parameter values, *RC* = {*ServerSoftware* (0.231), *NetworkNode* (0.192), *SwitchEquipment* (0.192), *Computer* (0.192), *Software* (0.192), *Cable* (0.192)}. The first five concepts are adjacent in the *networkA* ontology. Thus, they are combined into the group of concepts $Group_1$. The other group of concepts ($Group_2$) is composed solely by *Cable*. Since more than one group of concepts has been identified, the summarization process proceeds. All paths between $Group_1$ and $Group_2$ are identified. There are only two paths whose size is in the interval defined by Δ. The first path ($Path_1$) is: $Group_1$ → *Equipment* → $Group_2$. The second path ($Path_2$) is: $Group_1$ → *NodePair* → $Group_2$. The value of f-measure is identical for both paths (92.5). However, the average relevance of $Path_1$ (0.187) is higher than the average relevance of $Path_2$ (0.181). As a result, $Path_1$ is chosen as the ontology summary. The resulting summary is shown in Figure 4.

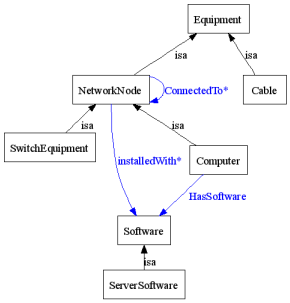

Fig. 4 The ontology summary for networkA

V. Experiments

In this section, we present an evaluation of the proposed ontology summarization process. First, we asked expert users, which are knowledgeable about specific ontologies, to produce manual summaries. This created a "gold standard" set of summaries against which our automatic summaries can be compared and analyzed. We also analyzed the impact of applying the summarization process in peer clustering.

A. Implementation

We have developed an ontology summarization tool to produce automatic summaries of OWL ontologies. The tool is implemented in Java and uses the OWL API to manipulate ontologies. A first version of the summarization tool is available for download at our project's website http://www.cin.ufpe.br/~speed/OWLSummarizer. The tool offers a graphical interface which enables the instantiation of the input parameters. A log file is created at each successful execution. The log file contains processing information produced by the tool during an execution, e.g. elapsed time and identified candidate paths. Such information can be useful to adjust the input parameters.

B. Comparison with Expert Summaries

We have selected three OWL ontologies belonging to distinct knowledge domains as test cases: the network ontology (Section IV), an office ontology, and an university ontology. Then, we invited three expert users to generate "gold standard" summaries for the three ontologies. Summaries of different sizes were requested: 4, 8, and 12 concepts. Correspondingly, we generated automatic summaries at the same sizes and measured the agreement between the automatic summaries and the expert summaries. Since frequency is not an intrinsic characteristic of ontologies, in order to be comparable with expert summaries only centrality was considered when generating the automatic summaries.

The agreement between two ontology summaries is defined as the percentage of the number of concepts selected by both the expert users and the summarization tool over the requested

summary size. An agreement of a particular summary size is generated by combining all expert summaries and retaining only the concepts selected by a majority of the experts (in this case, at least two experts). We have also compared the expert summaries against the summaries produced by OntoSum [15], a demo tool for summarizing small ontologies in real-time on the Web. Particularly, for OntoSum, we have used the Weighted PageRank measure since the authors affirm to have obtained the best evaluation for their ontology summaries. Table II illustrates the results of our experiments.

TABLE II
COMPARISON BETWEEN AUTOMATIC SUMMARIES AND EXPERT SUMMARIES

networkA.owl	**4-Concept**	**8-Concept**	**12-Concept**
Expert 1 against Automatic	50%	100%	-
Expert 2 against Automatic	50%	75%	-
Expert 3 against Automatic	50%	50%	-
User agreement against Automatic	50%	75%	-
User agreement against OntoSum	75%	75%	-
office-env2.owl	**4-Concept**	**8-Concept**	**12-Concept**
Expert 1 against Automatic	100%	75%	67%
Expert 2 against Automatic	75%	63%	58%
Expert 3 against Automatic	75%	63%	58%
User agreement against Automatic	100%	63%	58%
User agreement against OntoSum	50%	50%	75%
univ-cs.owl	**4-Concept**	**8-Concept**	**12-Concept**
Expert 1 against Automatic	-	75%	92%
Expert 2 against Automatic	-	88%	83%
Expert 3 against Automatic	-	50%	67%
User agreement against Automatic	-	75%	92%
User agreement against OntoSum	-	63%	75%

Except for the office ontology, our system was in reasonable consonance with human experts. The results for the office ontology are due to the fact that a high relevant concept was positioned far from the other relevant concepts. Consequently, candidate summaries containing this concept were very well evaluated (f-measure), even with some non-relevant concepts. Obviously, such non-relevant concepts were not chosen by the expert users. Experts do not always agree on what is the best summary. In general, the percentage of agreement between expert summaries and automatic summaries increases as the summary size augments. Briefly, Table II shows that our tool was able to produce summaries at different sizes that appear to be similar to what an expert may have produced.

During the experiments, we have observed some particular situations which are important to be stated: (i) as the summary size increases, the probability of forming only one group of concepts containing all relevant concepts is also increased. Consequently, the possibility of introducing non-relevant concepts in the summary decreases; (ii) at most one group of concepts with two relevant concepts was formed for the chosen ontologies; and (iii) in general, the use of a fixed summary size ($\Delta = 0$) does not allow the identification of the best summary. For a certain summary size, there were cases in which no summary was identified, e.g. a 4-Concept summary for the university ontology (Table II).

C. Using Ontology Summaries in Peer Clustering

In this experiment we evaluated how much the result of peer clustering is affected when each cluster of peers is represented by its corresponding ontology summary, instead of its entire cluster ontology. In our PDMS [4], peer clustering is mainly an incremental process, i.e., when an incoming peer arrives it must search for a cluster containing semantically similar peers in order to join. Each cluster is represented by a cluster ontology that is obtained by merging the schemas (ontologies) of the peers participating in the cluster. The search starts at an initial cluster and continues by visiting the semantic neighbors of the initial cluster disposed in an unstructured overlay network. At each visited cluster, the semantic similarity between the cluster and the incoming peer is computed. To this end, the ontology matching service proposed in [14] is used. It takes as arguments two ontologies (i.e., a cluster ontology and a local ontology) and returns a global measure which indicates the degree of similarity between both ontologies. A cluster is semantically similar to an incoming peer if the global measure between their ontologies is above a certain threshold. The most similar cluster is the one having the highest similarity with the incoming peer.

We included the ontology summarization tool in a PDMS simulator and analyzed cluster formation as incoming peers joined the system. Twenty peers (ontologies) were used in the experiment. We evaluated the clustering results using classical statistical indices [16]: *Rand Index*, *Jaccard Coefficient*, *Fowlkes-Mallows (FM) Index*, and *Hubert's statistic*. The indices were computed by comparing the clustering results obtained using entire cluster ontologies against the ones obtained using summarized cluster ontologies. The values of all statistical indices are between 0 and 1. The larger their value the higher the agreement between the two clustering results. We considered five variations for the centrality weight (C) and frequency weight (F). The result illustrated in Figure 5 indicates that with a well-balanced combination of the centrality and frequency measures it is possible to produce representative summaries that can replace cluster ontologies and still maintain satisfactory clustering results.

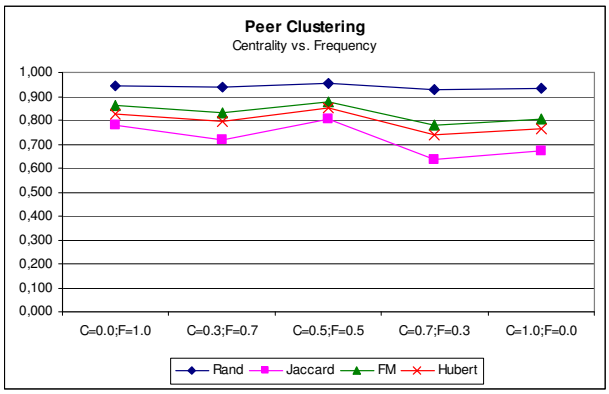

Fig. 5 Clustering results obtained by varying centrality and frequency weights.

VI. RELATED WORK

The first studies on schema summarization have focused on entity-relationship (ER) model abstraction. In such model, since data is not available, only the structural characteristics of ER diagrams are exploited. The authors of [7] use clustering techniques to produce a summarized version of an ER diagram. They present an algorithm for performing schema clustering, and then discuss criteria for representing clusters by means of

abstract elements and for abstracting links between elements. The amount of human effort required in this technique is significant, especially to define links between abstract elements. In [1], a summarization process for relational and XML schemas is proposed. The authors affirm that while schema structure is of vital importance in summarization, data distribution often provides important insights that significantly improve the summary quality. The authors of [2] argue that the previous summarization process cannot be applied to relational schemas since these schemas come with specific challenges that are not usually encountered in XML schemas. Thus, they propose a novel approach to summarize relational schemas that addresses the challenges associated with them.

We have also analyzed other summarization processes in which ontologies do not represent schemas. In [15], the authors propose a novel process to automatic ontology summarization based on RDF Sentence Graph. Summaries are customizable, i.e., users can specify the length of summaries and navigational preferences. A notion of RDF sentence is proposed as the basic unit of summarization. An RDF Sentence Graph is proposed to characterize the links between RDF sentences derived from a given ontology. The salience of each RDF sentence is assessed in terms of its centrality in the graph. An ontology is summarized by extracting a set of salient RDF sentences according to a re-ranking strategy.

In [17], an automatic method for structure-based ontology partitioning is proposed. The method consists in dividing a large ontology into smaller and disjoint modules based on the structural properties of the ontology. Each module contains information about a subtopic of the ontology. Concepts inside a module are stronger related among them than with concepts outside the module. The result is a connected graph where each node corresponds to a subtopic of the ontology. Although the set of modules can be considered as an ontology summary, considerations must be made: (i) a module is not a concept; (ii) since the modules are not too close to each other in the graph, no information is provided to explicit the kind of relationship between them; and (iii) during the partitioning process, the semantics of the relationships is not exploited to determine the level of dependency between concepts.

Some notion of centrality is used to calculate the relevance of concepts in all the discussed works. However, none of them exploits the type of relationships between concepts. Although the works of [1][15] affirm that their summarization process is fully automatic, the size of summaries is still manually provided. In [17], the number and the size of modules also need to be informed. Using frequency as a criterion for determining relevant concepts to be included in a summary is not considered by the presented works. This is due to the fact that existing solutions do not consider merged ontologies in their summarization processes. Another aspect that differentiates our process from the others is that summaries are not generated with the goal of easing the comprehension of large ontologies (schemas) by users. They should be used by ontology matching services in order to avoid matching operations involving large-scale ontologies in a PDMS.

VII. Conclusions and Future Work

This work proposes an automatic process to summarize ontologies representing multiple local schemas. The process has been instantiated in a PDMS to improve the efficiency of peer clustering which makes intensive use of an ontology matching service. To determine the relevance of concepts a combination of two measures was used. Centrality is calculated using an extended definition of the degree centrality measure. Frequency is used as a distinguishing criterion when the ontologies to be summarized are merged ontologies. A detailed description of the summarization process was presented as well as an algorithm for ontology summarization. Experiments have shown that the process is able to find good summaries compared to the ones manually generated by expert users.

There are a number of ongoing research issues concerned with the proposed summarization process which will be the goal of our future activity. An issue to be studied in deep detail regards the application of transitivity rules to identified paths in order to eliminate non-relevant concepts. In some situations, instead of adding non-relevant concepts in the summary, some relationships between relevant concepts could be inferred. The main idea is to automatically derive new relationships between relevant concepts which are separated by a non-relevant concept, and then remove the non-relevant concept and its relationships. Another research activity is devoted to executing experiments with the other types of centrality measures, i.e., closeness, betweenness, and eigenvector.

References

[1] C. Yu and H. V. Jagadish, "Schema Summarization", in Proc. VLDB'06, 2006, pp. 319-330.

[2] X. Yang, C. M. Procopiuc, and D. Srivastava, "Summarizing Relational Databases", in Proc. VLDB'09, 2009, pp. 634-645.

[3] A. Y. Halevy, Z. Ives, D. Suciu, and I. Tatarinov, "Schema Mediation in Peer Data Management Systems", in Proc. ICDE'03, 2003, pp. 505-516.

[4] C. E. Pires, "Ontology-based Clustering in a Peer Data Management System". Ph.D. thesis, UFPE, Recife, Brazil, Apr. 2009.

[5] V. Kantere, D. Tsoumakos, and T. Sellis, "A Framework for Semantic Grouping in P2P Databases", Information Systems Journal, vol. 33, pp. 611-636, March 2008.

[6] J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer-Verlag, Heidelberg, 2007.

[7] S. Castano, V. Antonellis, M. G. Fugini, and B. Pernici, "Conceptual Schema Analysis: Techniques and Applications", ACM Transactions on Database Systems, vol. 23, pp. 286-333, Sep. 1998.

[8] T. R. Gruber, "Toward Principles for the Design of Ontologies used for Knowledge Sharing", Journal Human-Computer Studies, vol. 43, pp. 907-928, Nov./Dec. 1995.

[9] P. Mika, *Social Networks and the Semantic Web*. Springer-Verlag New York, Inc., 2007.

[10] R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, Harlow, England, ACM Press, 1999.

[11] (2009) The OWL Web Ontology Language website. [Online]. Available: http://www.w3.org/TR/owl-features/

[12] R. Diestel, *Graph Theory*, 3rd ed., Springer-Verlag, Heidelberg, 2005.

[13] N. F. Noy and M. A. Musen, "Prompt: Algorithm and Tool for Automated Ontology Merging and Alignment", in Proc. AAAI'00, 2000, pp. 450-455.

[14] C. E. Pires, D. Souza, T. Pachêco, and A. C. Salgado, "A Semantic-Based Ontology Matching Process for PDMS", in Globe'09, 2009, pp. 124-135.

[15] X. Zhang, G. Cheng, and Y. Qu, "Ontology Summarization Based on RDF Sentence Graph", in WWW'07, 2007, pp. 707-716.

[16] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 2nd ed., Academic Press, 2003.

[17] H. Stuckenschmidt and M. Klein, "Structure-Based Partitioning of Large Concept Hierarchies", in ISWC'04, 2004, pp. 289-303.