Towards Evaluation of Peer-to-Peer-based Distributed Information Management Systems

Marc Ehrig and **Christoph Schmitz** and **Steffen Staab** and **Julien Tane** and **Christoph Tempich**
{ehrig,schmitz,staab,tane,tempich}@aifb.uni-karlsruhe.de
Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS/

Abstract

Distributed knowledge management systems (DKMS) have been suggested to meet the requirements of today's knowledge management. Peer-to-peer systems offer technical foundations for such distributed systems. To estimate the value of P2P-based knowledge management evaluation criteria that measure the performance of such DKMS are required. We suggest a concise framework for evaluation of such systems within different usage scenarios. Our approach is based on standard measures from the information retrieval and the databases community. These measures serve as input to a general evaluation function which is used to measure the efficiency of P2P-based KM systems. We describe test scenarios as well as the simulation software and data sets that can be used for that purpose.

Introduction

Many enterprizes have spent large amounts of money to implement centralized knowledge management systems to keep in business in today's knowledge-based economy, often with little success. Among others (Bonifacio, Bouquet, & Traverso 2002) suggest a distributed approach to knowledge management which better fits organizations and their employees.

Participants can maintain individual views of the world, while easily sharing knowledge in ways such that administration efforts are low. The distributed environment is implemented by a peer-to-peer network (which is basically equivalent to a system of distributed agents) without any centralized servers. P2P systems have been used for collaborative working or file sharing, but knowledge sharing applications herein mostly relied on keyword search and very basic structures. Modern (centralized) knowledge management systems are based on ontologies which have shown to be the right answer for problems in knowledge modelling and representation (OLeary 1998). An ontology (Gruber 1993) is a shared specification of a conceptualization. Through their structure ontologies allow answering a wider range of queries than standard representations do. Semantic Web technologies can augment this (Maedche 2002). Current research projects[1] attempt to exploit the best of the two worlds. Specifically, we want to do semantic information retrieval in a peer-to-peer environment - resulting in a Distributed Knowledge Management System (DKMS).

In this work, we suggest a framework for evaluation of such distributed knowledge-based systems. Only through a thorough evaluation we can gain the insights to further develop and enhance ideas and systems. Evaluation is either possible through user-based evaluation or system evaluation. User-based evaluation measures the users satisfaction with a system, system evaluation compares different systems with respect to a given measure.

While system evaluation permits a more objective confrontation of different approaches, the correlation of the results with the final user satisfaction is not always clear. However, user-based evaluation is expensive, time-consuming and it is difficult to eliminate the noise which is due to user experience, user interface and other human specific factors.

Tools developed within the cited projects focus on the technical aspects of knowledge management. Thus, we use the system evaluation approach. Techniques from traditional Information Retrieval (Voorhees 2002) and networking research (Yang & Garcia-Molina 2002c) will have to be combined with ontology specific measures to gain meaningful results.

This paper is structured as follows: in the first section we will introduce a set of use cases to illustrate the different dimensions of the problem at hand. A definition of evaluation measures will be given in the second section. In the following section we want to give a notion of tools which can be used. A part on generation of test data for these simulations follows in the succeeding section. Further we give a practical view describing the test parameters. Related and future work conclude this paper.

Scenarios

The field of possible applications for peer-to-peer computing is huge. Currently running systems include file sharing (e.g. Gnutella[2] (Kan 1999)), collaboration (e.g. Groove[3]),

[1] SWAP (swap.semanticweb.org) and Edutella (edutella.jxta.org)

[2] http://www.gnutella.com

[3] http://www.groove.net

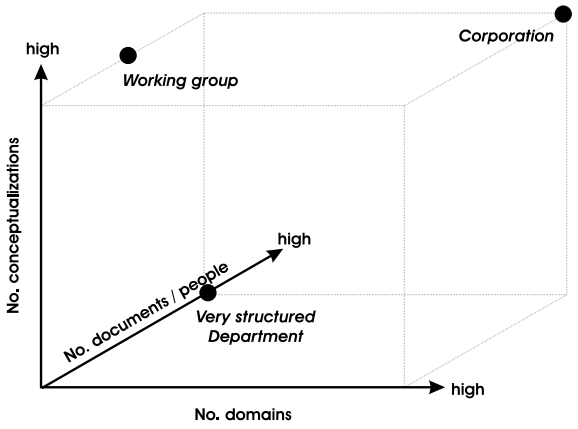

Figure 1: Scenarios Overview

computing (e.g. Seti@home[4]) to name but a few. For this reason we provide some scenarios for DKMS we examine. Various conclusions for our ontology based KMS will be drawn from these scenarios.

Application scenarios

This part will describe some real life situations in order to find characteristics which influence the distribution of information within the examined scenarios (Figure 1). The purpose of the scenarios is not to give a detailed impression of the entire IT-structure within the scenario. But rather to emphasize the challenging points for an ontology based KM-system realized by a peer-to-peer network.

Corporation With their organization in many different units, entire corporations impose the most complex situation, with respect to number of domains, conceptualizations and documents, we want to consider here. Typically these units are distributed according to organizational tasks, like accounting or human resources, or more product related such as development and marketing. The product related units, for example, work on one product (topic) with diverse perspectives or on varying products with similar views, viz. use the same vocabulary.

We assume each peer[5] has its own ontology, but with the addition that employees working in similar business units use ontologies which have some concepts in common while ontologies in unrelated units describing e.g. the same product are not easily comparable, viz. use a different hierarchy and vocabulary.

Our evaluation has to show which techniques make best use of existing ontologies in order answer queries according to the user needs. These demands will be examined precisely in the future. Therefore queries must reach quickly the peers which can answer them, without flooding the network. The answers should be relevant with respect to the query. Further the demand for computer resources like storage and processor time has to be monitored.

Working group A special case within a big company is the single department. In this case the domain is predefined and terms with the same meaning occur more often in each ontology. However, the demand in terms of retrieval accuracy increases. A major research question here is, how to capitalize on ontologies from other peers. *viz.* Selforganization is often cited as one of the advantages of peer-to-peer systems. If every peer partly conceptualizes information the combination will result in a more detailed description for everybody, because each peer can add concepts from other peers to its own structure.

Very structured department A department of the kind using a very structured process. In this case it is possible to define and implement a single ontology which any employee has to follow.

Summary

To summarize the single cases from an ontological point of view we distinguish two dimensions. The number of domains which are conceptualized and conceptualizations used for one domain. From the combination four possibilities evolve. This observation is in line with the suggestions in (Wache *et al.* 2001).

nm ontologies Each peer uses its own ontology. These ontologies conceptualize different domains.

n1 ontologies Each peer uses its own ontology, but all peers conceptualize the same domain.

1m ontologies There is one general ontology, but it conceptualizes many domains. The peers use only parts of the entire ontology. But they can be merged from a top level perspective. In this case two different possibilities evolve:

- **disjoint** The peers commit to a particular part of the ontology. Hence two peers use either the same or a different ontology.
- **overlapping** Each peer has parts of the ontology without respect to the ontologies others are using.

11 ontology Each peer uses the same ontology in one domain.

From a technical point of view we consider networks with a small numbers of peers to huge corporate networks. This means, that different routing strategies have to be analyzed.

The evaluation criteria are straightforward. In all cases the relevance of the answer should be high and response time low using little resources of the peers. Further aspects are the network behavior if single peers fail or return wrong answers.

Ontologies provide means to define contexts. The effects on these criteria through incorporation of meaning will be evaluated.

The case studies have demonstrated the kind of peer-to-peer system we focus on. To evaluate our techniques we use well established measures from the Information Retrieval

[4] http://setiathome.ssl.berkeley.edu/

[5] A peer can be the computer system of one user or a general database.

and Peer-to-Peer community, but we also have to introduce new ones which take the use of ontologies into account. These measures are described in the following section.

Evaluation Functions and their Parameters

This section presents a theoretical model of evaluation. In a general overview we define the evaluation function followed by its premises. Additionally we present ideas of which input and output parameters can be of interest in a DKMS.

Evaluation as a Function

One can imagine our DKMS as a black box doing information retrieval in a Semantic Web environment. This black box is supposed to have a certain behavior from which evaluation figures result giving us insight into the DKMS. To test and measure this behavior we can adjust different input parameters and collect the output figures.

This can be modelled as a function. The function (F) describes the setting and the basic algorithms used, that is, the interior of our system. Different parameters are used as input (i_n) e.g. the number of peers. Specific output figures (o_m) result from it, e. g. relevance or performance measures. Input and output in this context are *not* queries and answers of the peer network, they rather are parameters of the DKMS and its methodologies.

$$(i_1, i_2, \ldots, i_n) \xrightarrow{F} (o_1, o_2, \ldots, o_m)$$

Having discussed the correlation between input and output one can adjust the parameters until an optimal solution is found.

This approach is designed along an implementation line with the function representing the hard-coded program and the parameters being variables of it.

Function Modelling

The function depends on the algorithms and other properties which will be described further.

Topology. The topology is crucial for the network load imposed by each query. Do we want to evaluate random graphs, the star topology, or the HyperCuP environment (Schlosser *et al.* 2002)? Further the content of each peer (and its semantic context) could be used for building a network structure.

Document distribution. The distribution of the documents in a real peer-to-peer system is hardly random. The influence of different document distributions on the output figures will be evaluated.

Query language. The query language defines the expressiveness of queries. It can be interesting to compare performance results of the peer-to-peer system between query languages which only allow conjunctions or disjunctions and query languages which allow complex recursive queries.

Selection function. Having a peer structure and a formulated query the next step is to find good ways of matching them. How to select and route to the best peers is a core component (Carzaniga & Wolf 2002).

For the reader it might be confusing why the mentioned points belong to function rather than to input parameters. In a way the function premises are also input parameters. The difference lies in fact that they are explicitly modelled in the algorithm and can not be changed easily. Input parameters on the other hand are more flexible and can be adjusted by changing the value of a variable of the algorithm, the algorithm itself will stay the same. The next paragraph will show this.

Input Parameters

A list of possible input parameters that can be entered into the system will follow:

Number of peers. The size of the peer-to-peer network affects the results of the system. The scalability of the system is represented by this number.

Number of documents or statements. Another type of scalability is checked with this parameter. Whereas peers are physical locations, this parameter describes content objects. They represent the smallest entities in the system.

Structure. Most of the decisions around topology directly influence the function. But depending on the chosen topology different parameters can be used for further adjustment. When using indexes an important figure is the *index size*. How much content will eventually be stored in the network and how detailed is the knowledge about other peers. Slightly different is the *level of connectivity* or *the size of the routing table*. These are figures representing the characteristics of the network.

Output Figures

The output figures of evaluation functions ensure comparability to other systems. As the area of semantic peer-to-peer systems is rather new, there are no established standard evaluation functions which makes it difficult to fulfill the first mentioned requirement. The following list will provide well-known evaluation functions from related research fields.

Relevance. Relevance is the subjective notion of a user deciding whether the information is of importance with respect to a query. Approximations can be done using, e. g., keywords. For comparison purposes one could imagine to have a rating between 0 and 1 for each answer.

Recall R. Recall is a standard measure in information retrieval. It describes the proportion of all relevant documents included in the retrieved set.

$$R = \frac{|relevant \cap retrieved|}{|relevant|}$$

Precision P. Precision is also a standard measure in information retrieval. It describes the proportion of a retrieved set that is relevant.

$$P = \frac{|relevant \cap retrieved|}{|retrieved|}$$

F-measure F. Several combinations of the two first mentioned measures have been developed. The most common one is the F-measure (Van Rijsbergen 1979) describing the normalized symmetric difference between retrieved and relevant documents.

$$F = \frac{(\beta^2+1)PR}{\beta^2 P+R} with \quad \beta = P/R$$

Information loss. A measure to evaluate the loss of information which occurs when a query must be generalized on the answering peer. This might happen if the queried ontology does not contain a specific concept, but one which is

more general and included in the ontology of the requesting peer (Mena *et al.* 2000).

Reliability. This can be split into two sub-parameters. *Fault tolerance* describes which degree of failures and problems are still tolerated until the system finally breaks down. Failures in a DKMS can be a peer leaving the network or unacknowledged messages. The *failure rate* specifies the percentage of actual breakdowns of the whole system.

Real time. This measures the time from sending off the query to getting a result. As this figure is critical for end users, we take it into consideration as well. It was used in (Nodine, Bohrer, & Ngu 1998).

Network load. This technical figure can be measured with different sub-parameters. This is especially important for internal technical measurements (Yang & Garcia-Molina 2002a). *Messages per query* traces to what extent the network is being flooded by one query. The number of *average hops* can indicate how goal-oriented a query is routed and how fast a answer may be returned.

Time to satisfaction. It is a combination between relevance and real time, with relevance having to exceed a certain value(Yang & Garcia-Molina 2002b). Again this is a very subjective figure.

Output combination

We have set up a theoretical model for evaluation. The benefit of semantic peer-to-peer lies not on its single areas but its strength actually is the combination of them. Just like the input parameters come from the different areas of peer-to-peer, Semantic Web, and information retrieval, it is also necessary to unite the output figures to achieve meaningful results. A possibility would be to arrange linear combinations. Normalized vectors represent another. The combination of different output figures will finally allow us to decide upon the quality of the new system.

The output figures will be provided using a simulation package.

P2P Network Simulation

P2P systems are not set up and maintained by a central authority; thus, creating and observing a non-trivial network and measuring the evaluation functions as described in the previous section is a hard task. Simulation can help to gain insight into the behavior of the system. Many research contributions such as Freenet (Clarke *et al.* 2000) and Anthill (Montresor, Meling, & Babaoglu 2002) have used simulations in order to demonstrate the performance of their systems. Simulation is a core component for evaluation.

Discrete Event Simulation (DES)

Discrete Event Simulation observes the behaviour of a model over time (Ball 1996). The model has a *state* described by variables of the model that completely define the future of the system. The state of the model is usually encapsulated into a set of *entities* (cp. objects in OOP). *Discrete Events* changing the state of the system occur at discrete points in time (as opposed to continuous state changes). Events may trigger new events. *Statistical Variables* define the performance measure the user is interested in. This could be something like "average load on the server" or "maximum queue length".

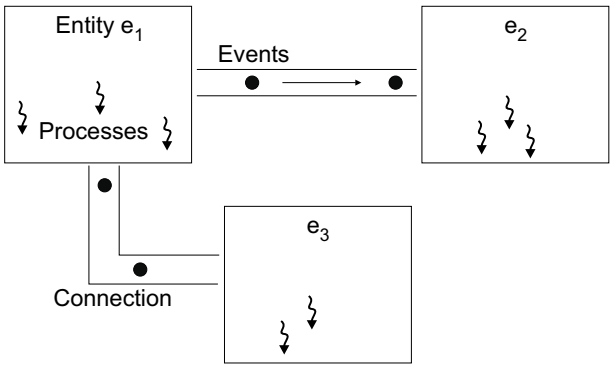

Figure 2: Abstractions in DES packages

Event oriented DES describes the dynamic behavior of the system solely by a sequence of events; the actions triggering the events are not considered. *Process oriented* DES combines the entities containing the state of the system and the actions that cause events (cf. OOP).

Typical Components of Simulation Software Packages DES software typically includes abstractions for entities, connections between entities, and events transmitted on those connections (see fig. 2), which corresponds well to the P2P scenario. Process oriented packages also include an abstraction for processes running on entities. Some simulators provide a *glue language* which can be used to compose models easily.

Simulation Packages

Several simulators, all of which contain the abstractions mentioned above, were examined in more detail: SSF in its incarnations DaSSF[6] (C++-based) and Raceway SSF[7] (Java); OMNet++[8]; JavaSim[9]; Simjava[10]; JADE[11].

Simulators Feature Matrix We only present a short overview of the details, as the features of the above mentioned packages are similar.

Selection Criteria for Simulation Infrastructures

Performance considerations Qualitative experiments concerning model sizes and performance were conducted on a commodity PC in order to find out how large models could become.

[6] http://www.cs.dartmouth.edu/ jasonliu/projects/ssf/

[7] https://gradus.renesys.com/exe/Raceway

[8] http://www.hit.bme.hu/phd/vargaa/omnetpp.htm

[9] http://javasim.cs.uiuc.edu/

[10] http://www.dcs.ed.ac.uk/home/hase/simjava/

[11] http://sharon.cselt.it/projects/jade/; JADE is a special case here because it is an agent platform rather than a simulation package, but nevertheless it contains the same abstractions as the others.

Name	Language	Distributable	Glue language
Raceway SSF	Java		DML
DaSSF	C++	MPI	DML
Omnet++	C++	PVM/MPI	NED
JavaSim	Java		Tcl/Python
SimJava	Java		
JADE	Java	CORBA	

Table 1: Overview of different simulations systems

JavaSim could handle about 6000 entities, while all other Java-based systems are restricted to less than 1024 active entities. The C++ systems can handle hundreds of thousands of entities and process tens of thousands of events per second.

Ease of implementation Undoubtedly, a model based on a Java-based API and simulator is much easier to program and debug than C++. The exception handling and debugging capabilities of the Java language facilitate a rapid generation of models.

Glue languages, graphical environments While glue languages and graphical editors are useful in order to get started, they may not be able to cope with complicated and/or large models. In that case, a clean programming interface on the C++/Java level is crucial.

Conclusion

A Java-based simulator would be much easier to get started with. On the other hand, for large models the C++ systems are able to handle numbers of entities that are two orders of magnitude larger than those of the Java systems.

We are currently implementing a simulation environment with JAVA using the SSF framework.

Data Generation - Evaluation Datasets

No evaluation can be done without using a dataset which we can query on the semantic level. The choice of this dataset will be influenced by different criteria. First, we need to consider what type of semantic data we want to query. Then we explore the problem of how the data should be distributed on the network.

Data Understanding

One can see the peer-to-peer network as a network of repositories called peers. Each provides a set of resources, which we will call documents. Every single document is then described by some sort of schema shared across the network. In usual peer-to-peer systems, the metadata is provided with a very simple schema of fields containing plain text (e.g: title, author or format of the document considered). The query must then have the following form: *return all instances (approximately) having the following values* $v_1, ..., v_k$ *in the fields* $f_1, \ldots, f_k$.

A semantic query in the peer-to-peer network is a query using the semantic metadata available on the given objects.

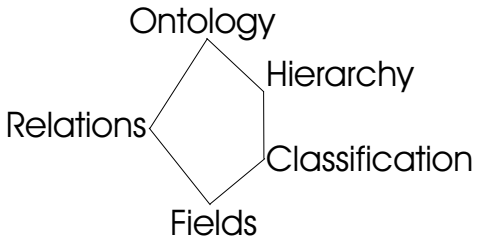

Figure 3: Different types of metadata schema

The structure of the metadata is enriched in order to address certain issues. We list them here with a few words of explanation as well as the kind of query that might require it.

1. identity problem: simple values do not determine identity
   e.g *return all the documents written by the CEO aka Mr Johnson*
2. relational problem: values in fields are not links to other objects
   e.g *return all the papers of scientist who had an accepted paper at the SIGMOD conference*
3. the classification problem: a given classification needs to be shared to be useful on the network [12]?
   e.g *return all documents written by university professors*
4. the inference problem: inclusion of classes also need to be shared
   e.g *return all documents on semantics theory in Computer science (not equal to the intersection of "semantics theory" and "Computer science")*

Moreover, these different types of queries can be combined, for example:
e.g *return the different names of all oil companies quoted at the Stock Exchange*. An ontology is a metadata schema whose semantic description addresses these issues.

Generating the Data - Existing Data

A possible approach for the evaluation of a semantic peer-to-peer network is to define a generation mechanism, which generates data with a semantic-like structure. However, for evaluation purposes the difficulty of deciding whether the data generated corresponds to typical real data might turn out to be a drawback.

A second approach is to use existing datasets and distribute them over the network. However, for certain types of datasets of very specialised domains, the drawback here will be that one might have some difficulty to interpret the results (for example MEDLINE dataset). Table summarizes the different datasets considered.

Each row corresponds to a given corpus we considered, whereas the columns are criterias of interests for our purpose. DBLP[13] is a computer science article bibliography database. Medline[14] offers the same purpose for medicine.

[12] the difference to keyword-based is that the values belong to predifined values, shared in the network

[13] http://www.informatik.uni-trier.de/ ley/db/

[14] http://www.nlm.nih.gov/bsd/sample_records_avail.html

Corpus name	nb of Docs	Text?	type
DBLP	310000	no	relational
Reuters 21578	21,578	yes	classification
Reuters 2002	806,791	yes	small hierarchy
DMOZ	190,000,000	yes	hierarchical
Medline	1141893	no	ontology

Table 2: Datasets of different types

Both of the Reuters datasets[15] are newswire collections from Reuters. DMOZ[16] is a collection of internet links organized in a hierarchy. The following criteria have been considered for our evaluation: the number of documents, whether the texts of the documents are available, and then the kind of metadata schema used.

Distributing the Dataset

Once the dataset has been chosen, it must be distributed on the peers of the network. For this, different possibilities might be chosen depending on the structure of the network and of the datasets. Of course, it is always possible to distribute the content among the peers randomly. However, this is probably not going to be the case in a peer-to-peer network. For instance, on a given peer it is more likely to find the similar content than on any other peer. Thus, other data distribution schemes have to be chosen according to the test scenarios we want to cover.

Test scenarios

In this section we suggest several test scenarios to evaluate different retrieval strategies. As discussed before the function modelling as well the input parameters influence the performance of the peer-to-peer system. It is essential that we examine each parameter separately to obtain meaningful results. Therefore we now outline the dimensions of our special interest.

Ontology

The first dimensions is the number of ontologies as discussed in the scenarios. Since already available ontologies are rare we will use different approaches to generate different kinds of ontologies. To generate different ontologies out of one general ontology it is possible to take the existing one and to exchang concept names with synonyms and deleting other concepts completely. However, we keep in mind that the generating strategy will certainly influence our results.

Matching algorithms

In the test cases with more than one general ontology mappings must be applied to identify the concepts with the same meaning on different peers. Some strategies are already available like (Magnini, Serafini, & Speranza 2002; Modica, Gal, & Jamil 2001; Melnik, Garcia-Molina, & Rahm 2001; Steels 1998) which are based on lexical, textual and structural matchings. Other are focusing on statistical information.

Network topology and routing

The network topology directly influences the available routing algorithms. In the firework routing model (Ng & Sia 2002), for example, a query is forwarded until a peer knows something about the query. The query is then distributed to all peers in the neighborhood. Within a semantic context the network topology must support that peers with similar domains know each other to use this model efficiently.

Query language

Query languages will be tested which support simple keyword based search but also complex recursive queries. Besides, it is also important to consider the construction of a query. Possibilities are to take just the actual chosen keyword and concepts or to expand them with various techniques.

Number and distribution of documents or statements

Our scenarios do not impose restrictions to the number of documents within the peer-to-peer network. However, the distribution of the documents will influence our results. In distributed database research documents are generally distributed uniformly (Callan & Connell 2001). When semantics come into play this does not seem appropriate. A short analysis of the Yahoo! categories suggests, that not only single words in documents are following a Zipf distribution (Zipf 1949), but also the allocation of documents in categories. Besides statistical distribution functions we also consider to distribute clusters. Different methods can be used to cluster our documents (Sebastiani 2002).

Number of peers

The number of peers surely influences the behavior of any peer-to-peer system. Therefore we will use ranges from small (about 10) to big (about 10^6) scenarios in our test cases. As with documents, the distribution of peers in a network at large follows power-laws (Faloutsos, Faloutsos, & Faloutsos 1999).

Structure

The available resources on each peer influences the information a peer can hold about the others. In this early phase of our projects we will not pay too much attention to this parameter but rather put as much (within reasonable limits) information on each peer as we need. The same holds for the computational effort on the peers. Therefore we distinguish three cases

1. All peers offer only a limited resources to the network.
2. Some peers offer a lot some peers offer limited resources.
3. All peers offer a lot of resources to the network.

Figure 4 gives an overview of the different scenarios.

[15] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html http://about.reuters.com/researchandstandards/corpus/index.asp

[16] http://www.dmoz.org

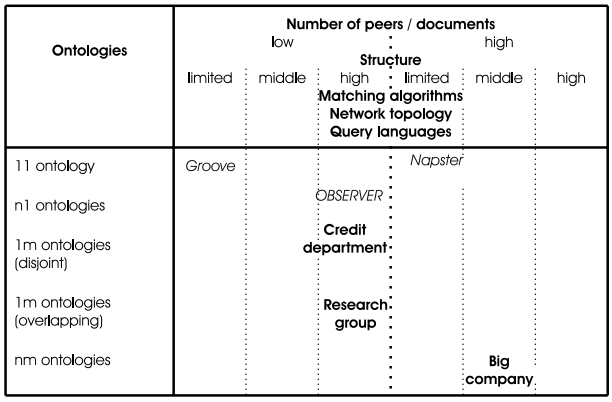

Figure 4: Test scenarios Overview

Related work

We found that there are different communities coping with the task of retrieving information from knowledge sources. They use either system evaluation or user-based evaluation.

Classical information retrieval from text documents is mostly affected by technical changes to the system. Therefore they predominantly use system evaluation to compare different methodologies (Voorhees 2002). Closest to the peer-to-peer approach in information retrieval are the results from e. g. (Callan & Connell 2001; Gravano & García-Molina 1995) to search distributed databases. The focus has been to choose from a known set of databases where the structures are known. The selection was made on keyword based criteria. As a testing environment the TREC dataset[17] was chosen and the different documents where distributed uniformly according to their creation date.

Our approach adds new dimensions to these results since the total number of peers is not known; neither are the information structures on the peers. Further we introduce new methods to distribute data on different peers.

Research in Ontology based search in distributed environments has been conducted with systems like OBSERVER (Mena *et al.* 2000). The focus was rather to find strategies for better information retrieval in one particular case in our scenario than on comparing different strategies for many scenarios as is proposed here.

The first user-based evaluation of an ontology based KM system was realized by (Iosif & Sure 2002). It delivers encouraging results about the use of ontologies to retrieve knowledge. In contrast to our scenarios the tests were accomplished on a centralized system using one ontology.

Efficient file allocation with hashing algorithms in peer-to-peer networks has been the focus in research such as (Stoica *et al.* 2001). However, this approach is feasible only for rather simple knowledge representation as necessary for music-file search, where keyword matching on a file name may be sufficient. (Yang & Garcia-Molina 2002c) introduced a function to calculate search costs in peer-to-peer networks and algorithms to optimize the function with respect to varying parameters. The peer selection is based on rather simple meta information such as response time. We want to advance this approach including more content based meta information.

(Dieng *et al.* 1999) have summarized evaluation attempts regarding the economic-financial and the socio-organizational viewpoint of KM. The research in this area is complementary to our approach.

Furthermore there is ample research on evaluation methods to classify response times of databases, e.g. TPC[18] and other technical aspects of information retrieval. This is also complementary to our suggestions.

Our impression is that there is a lot of research dealing with certain aspects of peer-to-peer systems and knowledge management but no general framework to compare the different systems.

Conclusion

We have examined the problem of evaluating a distributed knowledge management system. While evaluation of a centralized KM system is a challenging task in itself, the distributed case adds more parameters to the evaluation function.

The well-known notions of precision and recall are not sufficient to evaluate the performance of a DKMS. A performance measure for DKMS must include semantic retrieval quality as well as measures from the P2P field like the number of hops needed to answer a query.

Simulation packages for testing have been investigated.

For traditional database and information retrieval systems the generation of test data has been examined, and standardized data collections are supplied. In our case of P2P knowledge management, neither standardized data generation methods nor test data sets are available. We have made suggestions on how that problem may be tackled; it will have to be verified that the test data we generated are valid in the sense that they resemble real-world data from use cases like ours according to certain similarity measures.

Different application scenarios have shown a variety of possible uses for a DKMS which have different impacts on the performance of the system, and thus on the evaluation process.

Our suggested framework for evaluation can be used as a basis for future research and development of distributed knowledge management systems.

Acknowledgements. Research reported in this paper has been partially financed by EU in the IST project SWAP (IST-2001-34103) as well as by the German Federal Ministry of Education and Research (BMBF) in the PADLR project.

References

Ball, P. 1996. Introduction to discrete event simulation. In *Proceedings of the 2nd DYCOMANS workshop on "Management and Control : Tools in Action"*.

[17] http://trec.nist.gov

[18] http://www.tpc.org/

Bonifacio, M.; Bouquet, P.; and Traverso, P. 2002. Enabling distributed knowledge management: Managerial and technological implications. *Novatica and Informatik/Informatique* III(1).

Callan, J. P., and Connell, M. E. 2001. Query-based sampling of text databases. *Information Systems* 19(2):97–130.

Carzaniga, A., and Wolf, A. L. 2002. Content-based networking: A new communication infrastructure. In *Proceedings of the NSF Workshop on an Infrastructure for Mobile and Wireless Systems*. Springer-Verlag.

Clarke, I.; Sandberg, O.; Wiley, B.; and Hong, T. W. 2000. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*.

Dieng, R.; Corby, O.; Giboin, A.; and Ribiere, M. 1999. Methods and tools for corporate knowledge management. *Int. Journal of Human-Computer Studies* 51(3):567–598.

Faloutsos, M.; Faloutsos, P.; and Faloutsos, C. 1999. On power-law relationships of the internet topology. In *SIGCOMM*, 251–262.

Gravano, L., and García-Molina, H. 1995. Generalizing GlOSS to vector-space databases and broker hierarchies. In *International Conference on Very Large Databases, VLDB*, 78–89.

Gruber, T. R. 1993. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In Guarino, N., and Poli, R., eds., *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Deventer, The Netherlands: Kluwer Academic Publishers.

Iosif, V., and Sure, Y. 2002. Exploring potential benefits of the semantic web for virtual organizations. In *PAKM to appear*.

Kan, G. 1999. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly. chapter Gnutella, 94–122.

Maedche, A. 2002. Emergent semantics for ontologies. *IEEE Intelligent Systems*.

Magnini, B.; Serafini, L.; and Speranza, M. 2002. Linguistic based matching of local ontologies. In *Workshop on Meaning Negotiation (MeaN-02)*.

Melnik, S.; Garcia-Molina, H.; and Rahm, E. 2001. Similarity flooding: A versatile graph matching algorithm. In *Proc. 18th ICDE Conf.*

Mena, E.; Kashyap, V.; Illarramendi, A.; and Sheth, A. P. 2000. Imprecise answers in distributed environments: Estimation of information loss for multi-ontology based query processing. *International Journal of Cooperative Information Systems* 9(4):403–425.

Modica, G.; Gal, A.; and Jamil, H. M. 2001. The use of machine-generated ontologies in dynamic information seeking. In *Proceedings of the Ninth International Conference on Cooperative Information Systems (CoopIS 2001)*.

Montresor, A.; Meling, H.; and Babaoglu, O. 2002. Towards adaptive, resilient and self-organizing peer-to-peer systems. In *Proceedings of the International Workshop on Peer-to-Peer Computing*.

Ng, C. H., and Sia, K. C. 2002. Peer clustering and firework query model. Technical report, The Chinese University of Hong Kong.

Nodine, M.; Bohrer, W.; and Ngu, A. H. H. 1998. Semantic brokering over dynamic heterogeneous data sources in infosleuth. Technical report, MCC.

OLeary, D. 1998. Using ai in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems* 13(3):34–39.

Schlosser, M.; Sintek, M.; Decker, S.; and Nejdl, W. 2002. HyperCuP - hypercubes, ontologies and efficient search on P2P networks. In *Proceedings to the Eleventh International World Wide Web Conference*.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* 34(1):1–47.

Steels, L. 1998. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 1(2):169–194.

Stoica, I.; Morris, R.; Karger, D.; Kaashoek, F.; and Balakrishnan, H. 2001. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, 149–160.

Van Rijsbergen, C. J. 1979. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow.

Voorhees, E. M. 2002. The philosophy of information retrieval evaluation. In Peters, C.; Braschler, M.; Gonzalo, J.; and Kluck, M., eds., *Evaluation of Cross-Language Information Retrieval Systems, Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001*, volume 2406 of *Lecture Notes in Computer Science*, 9–26. Darmstadt, Germany: Springer.

Wache, H.; Voegele, T.; Visser, T.; Stuckenschmidt, H.; Schuster, H.; Neumann, G.; and Huebner, S. 2001. Ontology-based integration of information - a survey of existing approaches. In Stuckenschmidt, H., ed., *IJCAI-01 Workshop: Ontologies and Information Sharing*, 108–117.

Yang, B., and Garcia-Molina, H. 2002a. Designing a super-peer network. Technical report, Stanford.

Yang, B., and Garcia-Molina, H. 2002b. Improvig search in peer-to-peer networks. Technical report, Stanford University.

Yang, B., and Garcia-Molina, H. 2002c. Efficient search in peer-to-peer networks. In *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*.

Zipf, G. K. 1949. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. MA: Addison-Wesley. Reading.