Related Paper Recommendation to Support Online - Browsing of Research Papers

Manabu Ohta[1], Toshihiro Hachiki[1], Atsuhiro Takasu[2]
[1]Graduate School of Natural Science and Technology
Okayama University
Okayama, 700–8530
Japan
[2]National Institute of Informatics
Tokyo, 101-8430
Japan
{ohta, hachiki}@de.cs.okayama-u.ac.jp, takasu@nii.ac.jp

ABSTRACT: *An online-browsing support system for research papers has been developed that extracts technical terms from a paper and presents links to useful pages such as those explaining the terms. A method to further use the extracted technical terms is proposed to recommend papers to a user that are related to the paper he or she is browsing. Specifically, the proposed method generates a bipartite graph consisting of papers retrieved by the extracted technical terms, which are called related papers, and technical terms appearing in these related papers. It then ranks the related papers using the HITS algorithm for analyzing the bipartite graph and recommends top-ranked papers to the user. The proposed method was compared with other recommendation methods in terms of effectiveness in an experiment.*

Keywords: Browsing Support, Research Paper Recommendation, Technical Term Extraction, HITS

Received: 3 September 2011, Revised 17 December 2011, Accepted 23 December 2011

1. Introduction

In the early days, digital libraries (DL) were constructed by scanning published books and papers and extracting machinereadable text using document image analysis technology such as optical character recognition (OCR) and document image analysis [1]. Such technology made it possible for old books and papers to be accessed via the Internet in the same way as the latest papers. Current DLs also constitute a kind of global library on the Internet. Although they are accessible on the Web, they still remain a digital version of traditional libraries. The Web contains a variety of information that can be used for various purposes. By linking them, books and papers can provide information in more effective ways. For example, research papers contain many unfamiliar technical terms to novice researchers, undergraduate students, and people whose expertise differs from the domain of interest. It is, however, not efficient for them to use a dictionary or search the Web every time they encounter unfamiliar terms. Therefore, we proposed enhancing research papers of DLs with the other resources, i.e., the Web [2], and implemented a prototype online-browsing support system for research papers. Specifically, the proposed system searches the Web for explanatory Web pages of the unfamiliar terms and provides links to the explanatory pages.

This paper proposes further using the extracted technical terms to recommend research papers to a user that are related to the paper he or she is browsing. We use XML files of research papers with OCR markups to find technical terms because OCRed text can be cheaply obtained from the scanned research papers in our DL. We first collect research papers related to the extracted technical terms by searching the DL and extract technical terms again from the collected research papers. Then, we generate a bipartite graph by assuming links from the collected papers to the technical terms appearing in them. We apply the HITS algorithm [3] for analyzing the bipartite graph to rank and recommend related papers.

In the related research literature collected using a browsed paper in this way, technical terms frequently appearing in them are considered important. Hence, papers written with a lot of such important technical terms are considered relevant to the browsed paper. In analysis of the proposed bipartite graph by HITS, we can consider relevant papers and important technical terms to correspond to hubs and authorities, respectively. It is, therefore, possible to recommend the most relevant papers by applying HITS to the bipartite graph and by selecting papers with the highest hub scores. As far as we know, the effectiveness of applying HITS to research paper recommendation in such a mode has not yet been examined.

The remainder of this paper is structured as follows. Section II briefly reviews the related work on research paper recommendation and the HITS algorithm. Section III introduces our paper recommendation method, and section IV gives experimental results to evaluate its performance. Section V summarizes the paper and mentions future work.

2. Related Work

2.1 Recommendation of Research Papers

Sugiyama et al. proposed scholarly paper recommendation using a user’s latent research interests that exist in their publication list [4]. They used not only a researcher’s past publications but also their neighboring papers such as citation and reference papers as context to build their research profiles. By experimentation, they verified the effectiveness of their approach for two classes of researchers: junior researchers who had only one recently published paper and senior researchers who had multiple past publications.

Ekstrand et al. presented and empirically tested a large collection of recommender algorithms for the task of generating an introductory reading list for a new researcher [5].

For user-based evaluation, they gave the recommender system a query set of five-to-ten research papers collected using a search tool and received a reading list consisting of five papers that were relevant to the query and important within the research literature. They augmented existing collaborative and content-based filtering algorithms with measures of the importance of a paper within the literature. They measured a node’s importance in the citation graph using common algorithms, such as HITS [3] and PageRank [6]. They reported that collaborative filtering that used citation information generated such reading lists well.

Our proposed method, however, uses neither user profiles nor citation information, which differentiates it from the above work.

In addition, Song et al. proposed a learning framework for tag recommendation for scientific and Web documents [7]. They defined tagged training documents as triplets (words, docs, tags), and represented them in two bipartite graphs, which were partitioned into clusters. Tags in each topical cluster were ranked by their ranking algorithm. Their experiments on large-scale tagging datasets of research papers indicated that their framework effectively recommended tags in one second on average.

2.2 HITS Algorithm

HITS proposed by Kleinberg [3] is a major ranking algorithm for Web search results, which is often compared to another major one, PageRank, proposed by Page et al. [6]. HITS is also applied to finding communities on the Web. HITS discovers authority and hub nodes by analyzing links among them on the basis of the notion that the relationship a node has with important nodes affects the importance of the node more than that it has with less important nodes. In the context of Web analysis, authorities are pages having sufficient information on a specific topic, whereas hubs are ones that have sufficient links to such authoritative pages.

According to the HITS algorithm, the hub and authority scores for a node are iteratively calculated by the following equations:

$$a_p = \sum_{q,\, q \to p} h_q, \qquad (1)$$

$$h_p = \sum_{q,\ p \to q} a_q. \qquad (2)$$

Note that $p \to q$ means node q is linked by node p . As iteratively calculating hub and authority scores starting with each node having a hub and authority score of 1 leads to diverging values, these scores must be normalized after every iteration. The final hub and authority scores are determined after repetitions of this process. Nodes are ranked in accordance with the final hub or authority scores.

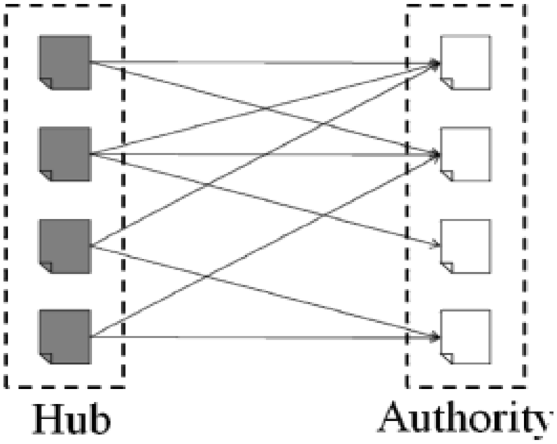


Figure 1. Reletionship between hub and authority

Kleinberg stated authoritative pages do not have links connecting each other, but are connected by hub pages that have links to multiple authoritative pages. Therefore, a good hub points to many good authorities while a good authority is pointed to by many good hubs, as shown in Figure 1.

As for application of the HITS algorithm, Nanba et al. proposed automatic detection of survey papers from a multilingual database using HITS [8]. They paid attention to the fact that important papers and survey papers respectively correspond to "*authority*" and "*hub*" in their citation relationship. They also modified HITS to improve accuracy of detecting survey papers by taking into account the contents of each paper.

3. Research Paper Recommendation

3.1 Outline

This section briefly describes how to recommend research papers relevant to a browsed one by using the technical terms extracted from it. We first collect related papers by searching our DL for every extracted technical term and then further extract technical terms from the related papers. We apply the HITS algorithm to ranking the related papers. HITS was originally used for ranking Web search results or finding Web communities. This paper, however, proposes applying HITS to link analysis of a bipartite graph consisting of related papers and technical terms appearing in them. By assuming each paper has links to every technical term appearing in it, paper nodes with only outlinks and technical term nodes with only inlinks constitute a bipartite graph.

1. The more frequently technical terms appear in the set of related papers, the more important the terms are to a browsed paper.
2. The more such important terms appear in a paper, the more relevant the paper is to the browsed one.

Figure 2. Assumptions on relationship between browsed paper and its related papers

For ranking related papers, we make two assumptions about the relationship between a browsed paper and its related papers as shown in Figure 2. On the basis of these assumptions, we can consider that highly relevant papers and important technical terms correspond to good hubs and authorities, respectively. Therefore, we rank related papers in accordance with their hub scores and recommend top-ranked papers to a user.

In concrete terms, we define the following procedure:

(i) Extract technical term candidates from a browsed paper p_{target} , rank them using TF-IDF, and select *K* top-ranked terms as a set of technical terms. We describe this term extraction in detail in subsection *B*.

(ii) Retrieve related papers $p_{ij} \in P_i$ (j =1,..., N) using each extracted technical term $t_i \in T$ (i =1,..., K)as a query for our DL. Note here that our DL ranks papers in descending order of their citation counts. In addition, we collect at most *N* papers for each term where papers previously retrieved by other terms are not counted among the *N* papers. Note also that $P = \bigcup_{i=1}^{K} P_i$.

iii) Extract a set of technical terms T^a_{ij} from each retrieved paper p_{ij} in the same way as (i). Note here that $T^a_i = \bigcup_{i=1}^{N} T^a_{ij}$ and $T^a = \bigcup_{i=1}^{K} T^a_i$.

(iv) A bipartite graph is generated by linking the set of papers collected in (ii), P , to the set of technical terms extracted in (iii), T^a . Applying HITS to the bipartite graph makes it possible to rank the papers in *P* on the basis of their hub scores. We explain the application procedure in more detail in subsection *C*.

3.2 Technical Term Extraction

We morphologically analyze the text of research papers using a Japanese morphological analyzer Sen [9] to extract technical term candidates as feature terms in accordance with the following rules:

1. Extract all the nouns and unknown terms solely consisting of alphanumerics, *katakana*, or *kanji* as feature terms.

2. Concatenate the continuous feature terms (if any) into one feature term.

3. Remove from the above feature terms those terms i) solely consisting of numerics or *hiragana*, ii) consisting of one character, and iii) in our stopword list.

After extraction, we apply OCR error correction to the extracted feature terms. We utilize the query correction function of the Yahoo!JAPAN search engine [10], i.e., “Did you mean *guessed-corrected-term*”. This query correction, however, is not always appropriate, especially for acronyms with various meanings. Therefore, the original feature term is corrected to the suggested query term only if the number of search results for the original term is less than 1,000. Then, we use TF-IDF to score all the extracted feature terms. The $tfidf_i$ of the term t_i is defined as follows:

$$fidf_i = tf_i \times \log \frac{num}{df_i}, \qquad (3)$$

where tf_i is the frequency count of the term t_i in the document from which *i t* is extracted, df_i is the document frequency of the term t_i , and *num* is the total number of documents. We define this df_i to be the number of papers retrieved by t_i , and regard *num* as the total number of papers stored in our DL. When we conducted the experiments described in section IV, the *num* was 13,206,916.

All the feature terms extracted from a research paper are ranked in accordance with this TF-IDF score, and the *K* topranked terms are selected as the technical terms used for retrieving related papers.

3.3 Applying HITS to Paper Recommendation

We first generate a bipartite graph as follows to apply HITS to related paper recommendation:

- Generate links from each paper to technical terms appearing in the paper.
- Regard papers and technical terms as hubs and authorities, respectively.

In analyzing the Web by HITS, each node can have both inlinks and outlinks and, hence, both hub and authority scores. In the bipartite graph mentioned above, however, paper nodes have only outlinks and term nodes only inlinks by their definitions. Therefore, we assign only hub scores to paper nodes and only authority scores to technical term nodes.

Figure 3 illustrates a simple image of the bipartite graph generated by the proposed method. In this bipartite graph, we iteratively calculate authority scores of term nodes and hub scores of paper nodes by using equations (1) and (2), respectively. After each iteration, these scores are normalized by the following equations:

$$\sum_p a_p^2 = 1, \tag{4}$$

$$\sum_p h_p^2 = 1. \tag{5}$$

The iteration continues until the absolute difference in authority or hub scores between iterations becomes sufficiently small. Finally, we obtain lists of related papers and technical terms ranked by their hub scores and authority scores, respectively. Thus, we can recommend highly ranked related papers. If two or more papers have the same hub score, we rank them by evaluating the following metrics in this order:

1. The rank of a related paper in the search result of our DL.
2. The TF-IDF score of the technical term by which a related paper is retrieved.

3.4 Prototype Browser

Figure 4 shows the GUI of the implemented prototype browser for reading research papers. The left window shows major bibliographies such as title, authors, abstract, and keywords. The title and authors are in Japanese and English, but the abstract and keywords are only in Japanese because our system targets only Japanese papers at present.

The right window shows a list of recommended papers ranked by the proposed method. A paper on "Personalized Web Search" is displayed in the left window, and four recommended papers are visible in the right window in Figure 4.

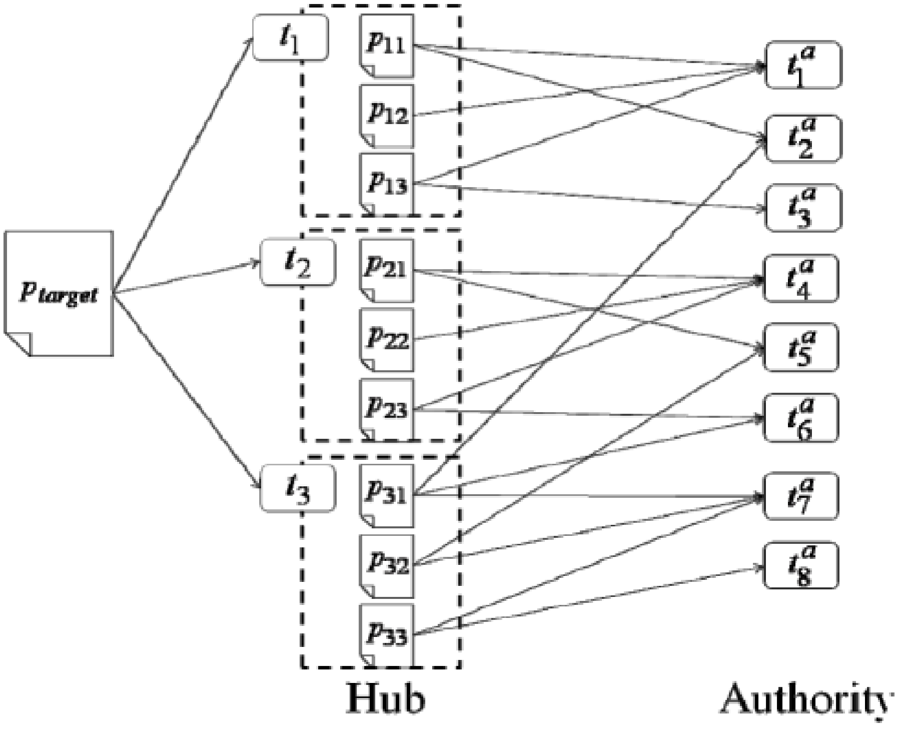
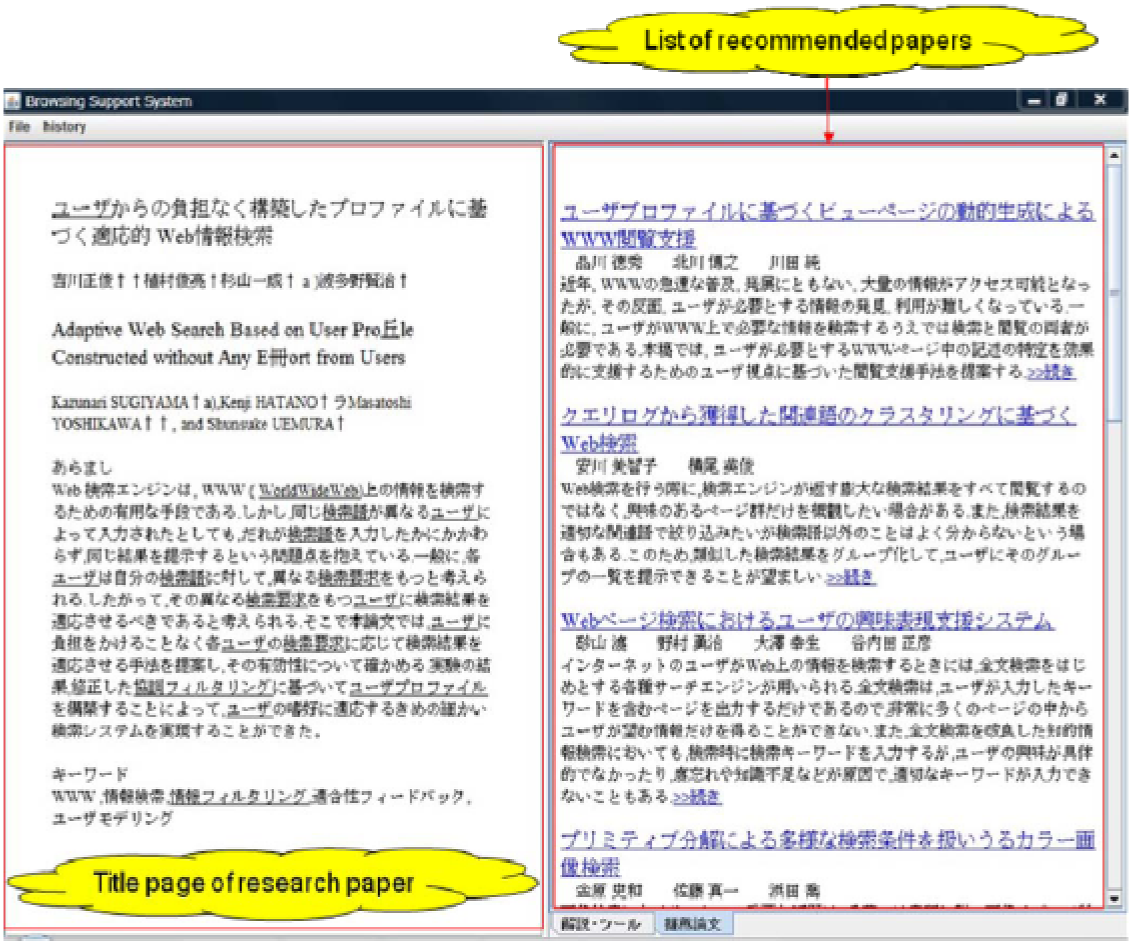


Figure 3. Bipartite graph to be analyzed by HITS



ユーザからの負担なく構築したプロファイルに基づく適応的 Web情報検索

吉川正俊†† 植村俊亮† 杉山一成† a) 波多野賢治†

Adaptive Web Search Based on User Pro丘le Constructed without Any E冊ort from Users

Kazunari SUGIYAMA† a),Kenji HATANO† ラMasatoshi YOSHIKAWA††, and Shunsuke UEMURA†

あらまし
Web 検索エンジンは，WWW (WorldWideWeb)上の情報を検索するための有用な手段である.しかし,同じ検索語が異なるユーザによって入力されたとしても,だれが検索語を入力したかにかかわらず,同じ結果を提示するという問題点を抱えている.一般に,各ユーザは自分の検索語に対して,異なる検索要求をもっと考えられる.したがって,その異なる検索要求をもつユーザに検索結果を適応させるべきであると考えられる.そこで本論文では,ユーザに負担をかけることなく各ユーザの検索要求に応じて検索結果を適応させる手法を提案し,その有効性について確かめる.実験の結果,修正した協調フィルタリングに基づいてユーザプロファイルを構築することによって,ユーザの嗜好に適応するきめの細かい検索システムを実現することができた。

キーワード
WWW ,情報検索,情報フィルタリング,適合性フィードバック，ユーザモデリング

ユーザプロファイルに基づくビューページの動的生成によるWWW閲覧支援
品川 徳秀　北川 博之　川田 純
近年, WWWの急速な普及, 発展にともない, 大量の情報がアクセス可能となったが, その反面, ユーザが必要とする情報の発見, 利用が難しくなっている.一般に, ユーザがWWW上で必要な情報を検索するうえでは検索と閲覧の両者が必要である.本稿では, ユーザが必要とするWWWページ中の記述の特定を効果的に支援するためのユーザ視点に基づいた閲覧支援手法を提案する.>>続き

クエリログから獲得した関連語のクラスタリングに基づくWeb検索
安川 美智子　横尾 英俊
Web検索を行う際に,検索エンジンが返す膨大な検索結果をすべて閲覧するのではなく,興味のあるページ群だけを概観したい場合がある.また,検索結果を適切な関連語で絞り込みたいが検索語以外のことはよく分からないという場合もある.このため,類似した検索結果をグループ化して,ユーザにそのグループの一覧を提示できることが望ましい.>>続き

Webページ検索におけるユーザの興味表現支援システム
砂山 渡　野村 眞治　大澤 幸生　谷内田 正彦
インターネットのユーザがWeb上の情報を検索するときには,全文検索をはじめとする各種サーチエンジンが用いられる.全文検索は,ユーザが入力したキーワードを含むページを出力するだけであるので,非常に多くのページの中からユーザが望む情報だけを得ることができない.また,全文検索を改良した知的情報検索においても,検索時に検索キーワードを入力するが,ユーザの興味が具体的でなかったり,度忘れや知識不足などが原因で,適切なキーワードが入力できないこともある.>>続き

プリミティブ分解による多様な検索条件を扱いうるカラー画像検索
金原 史和　佐藤 真一　浜田 喬

Figure 4. Prototype browsing support system with showing list of recommended papers

4. Experiment

4.1 Experimental Setup

First, we randomly selected 10 papers from six years worth of issues published by the Institute of Electronics, Information and Communication Engineers (from 2000 (Vol. J83-D-I) to 2005 (Vol. J88-D-I)). We evaluated performance on the basis of precision of top-ranked recommended papers such as precision at 10 recommended papers (p@10). Note that the experiments only used the Japanese title and abstract of papers. We, however, could extract technical terms from the whole paper in the same way. Moreover, we set the maximum number of extracted technical terms per paper because the more technical terms in a paper, the larger its hub score. We set this number to 10 on the basis of a preliminary experiment.

Second, our OCR system had good recognition accuracy. In recognizing a different Japanese academic journal, the accuracy was 99.00% for the abstract and 97.01% for the references. Mixtures of Japanese and English characters, as well as various fonts and punctuation symbols often appearing in the references, were difficult to recognize correctly.

In the experiment, one of the authors judged the relevance of the recommended papers to the browsed paper. We adopted rigid and relaxed relevance judgments in accordance with the degree of relevance. In the rigid judgment, only those papers that have the same purpose as the browsed one are relevant. In the relaxed one, papers that use the same technique are relevant in addition to those that have rigid relevance. For example, if a user browses a paper titled “Document classification using support vector machine”, papers on document classification are regarded as relevant in both rigid and relaxed judgments and papers reporting the use of support vector machine for another purpose are considered relevant only in the relaxed judgment.

4.2 Methods for Comparison

We implemented two recommendation methods for comparison with the proposed one.

- Vector space (VS) model-based method: This method is based on a VS model widely used in information retrieval. The VS model represents each paper as a vector of technical terms extracted from it. We define a similarity measure as the cosine of the angle between two paper vectors in order to recommend the papers most similar to a browsed one.

- Baseline method: This method directly reflects the assumptions shown in Figure 2. First, we score technical terms in related papers in accordance with their document frequencies. Then, we score each related paper a sum of the score of the technical terms appearing in the paper. Finally, we rank the related papers by their scores to recommend top-ranked papers to a user. The baseline method calculates scores of both papers and technical terms without iterative calculation, which differentiates it from the proposed HITS-based method.

4.3 Experimental Results

First, we show the number of related papers that can be collected from our DL with varying values of *N*, the maximum number of retrieved papers per technical term, in Table I. As explained in subsection *A*, we extract at most 10 technical terms from a paper. Hence, the maximum number of related papers is calculated as 10 × *N*. Table I shows the actual number of related papers is smaller than this because some papers have nine or fewer extracted terms and some technical terms have fewer than *N* related papers.

Next, we show the average indegrees of technical term nodes in the bipartite graph in Table II. Note here that the average number of technical terms refers to the average size of a set of technical terms, T^a, extracted from the related papers of the 10 papers selected for experiment. The terms with a large indegree affect hub and authority scores because such terms appear in many papers. Table II shows that the larger the *N*, the more technical terms are obtained, and the larger indegrees have the technical term nodes. The increase in indegrees is, however, subtle compared with that in the number of technical terms.

Moreover, we summarize some statistics of the technical terms with two or more indegrees in Table III because such term nodes play an important role in the proposed bipartite graph. The last row in Table III shows that the ratio of such terms to all the technical terms is from 6.1% to 9.2%, which means that more than 90% of extracted technical terms only appear in the one paper from which they are extracted.

Finally, Figure 5 plots the precision at 10 recommended papers of the VS model-based, baseline, and proposed methods with two relevance judgments w.r.t. the number of retrieved papers per term, *N*, of 5, 10, 30, and 50. As we can see from Figure 5, the proposed method achieved the best precision of 0.35 in the rigid judgment with *N* = 5 and of 0.90 in the relaxed one with *N* = 10. We can also see that the precision declines with the increase in *N* in the rigid judgment while the precision marks its highest value at 10 = *N* among 5, 10, 30, and 50 in the relaxed judgment. Compared with the other two methods, the proposed one performed better in the relaxed judgment irrespective of *N*. In the rigid judgment, the proposed method indeed achieved the highest precision of 0.35, but its precision worsens more than the other methods with *N* = 30 and *N* = 50.

# of retrieved papers/term (*N*)	5	10	30	50
Maximum # of related papers	50	100	300	500
Average # of related papers	37.9	69.3	178.7	269.0

Table 1. Average Number Of Related Papers

# of retrieved papers/term (*N*)	5	10	30	50
Average # of technical terms	346	615	1514	2228
Average indegree	1.09	1.12	1.17	1.19

Table 2. Average Indegreesr of Technical Terms t^a

# of retrieved papers/term (*N*)	5	10	30	50
Average # of technical terms	20.0	39.9	124.3	206.4
Average indegree	2.46	2 .74	3.00	3.09
Ratio of terms with two or more indegrees to all the terms (%)	6.05	6.53	8.32	9.21

Table 2. Average Indegreesr of Technical Terms t^a (≥ 2)

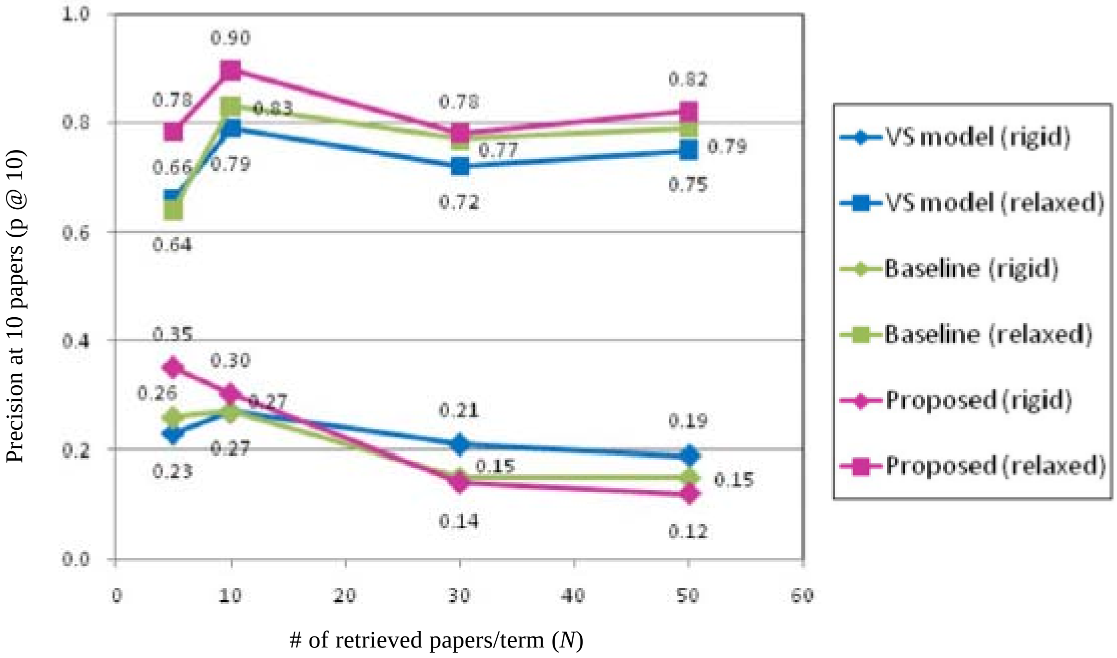


Figure 5. Precision of recommended papers

The VS model-based method achieved only low precision with a small value of *N* such as 5 irrespective of relevance judgment criteria. Although it achieved its best precision with *N* = 10, the values were the lowest among the three methods in both judgments.

On the other hand, the proposed method could recommend more relevant papers by using a small value of *N* such as 5, especially in the rigid relevance judgment. Using larger *N* of 30 or 50 did not improve precision. One of the major reasons for this is that inappropriate terms among the ones with high authority scores increased as *N* increased. The inappropriate terms are the extracted technical terms that are of little or no relevance to the main theme of a browsed paper and often badly affect paper recommendation. One of the remedies for this is considered to be weighting technical terms and related papers retrieved by the terms with the TF-IDF values defined by equation (3) because the terms and papers are respectively handled evenly once extracted or retrieved. We also need to sophisticate our technical term extraction procedure to select more appropriate terms.

When the proposed method was compared with the baseline one, the proposed method outperformed the baseline in most cases. The baseline method first calculates the scores of extracted technical terms on the basis of their document frequencies and then calculates the scores of papers. The proposed method takes into account linkage structure expressed in the bipartite graph in addition to the document frequency, which is considered effective for paper recommendation.

Figure 5 also shows that precisions in the relaxed judgment are more than double those in the rigid judgment irrespective of recommendation methods. In the rigid judgment, only papers with the same research purpose as the browsed one are judged as relevant recommendations. However, we were not always able to collect such papers sufficiently, which leads to this low precision. To improve precision in the rigid judgment, it may be effective to retrieve related papers by using plural technical

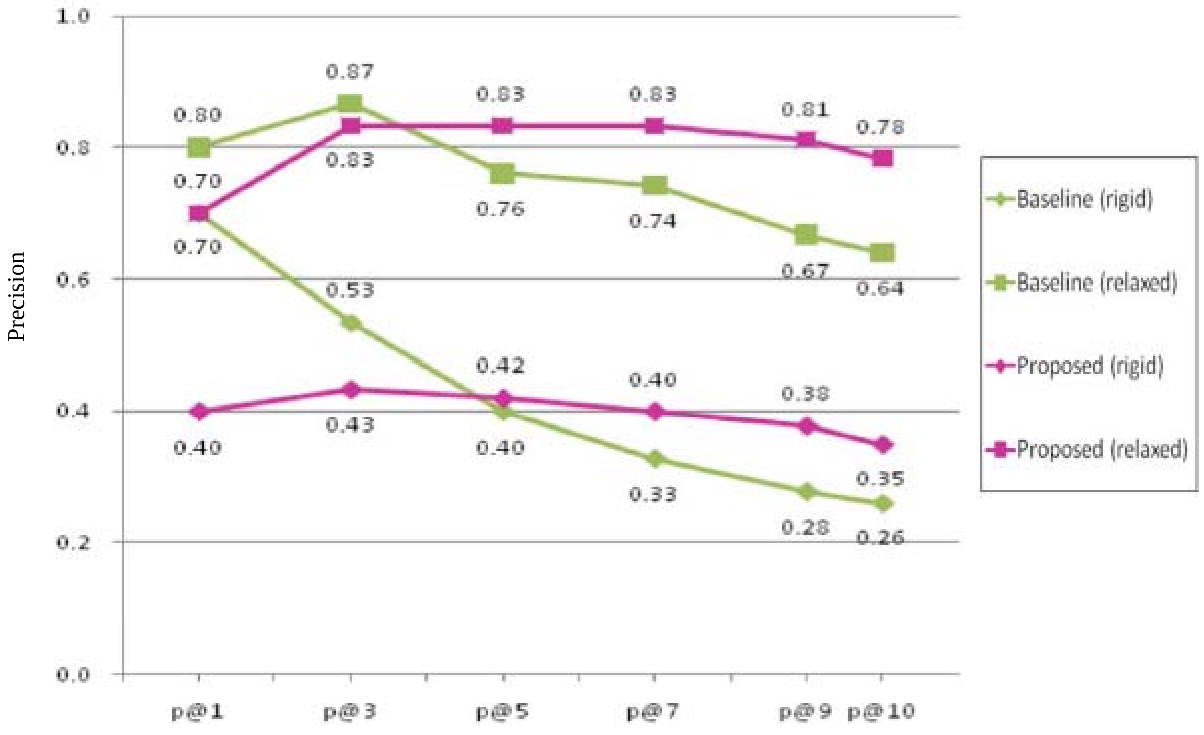


Figure 6. Comparison of proposed and baseline methods (*N* =5)

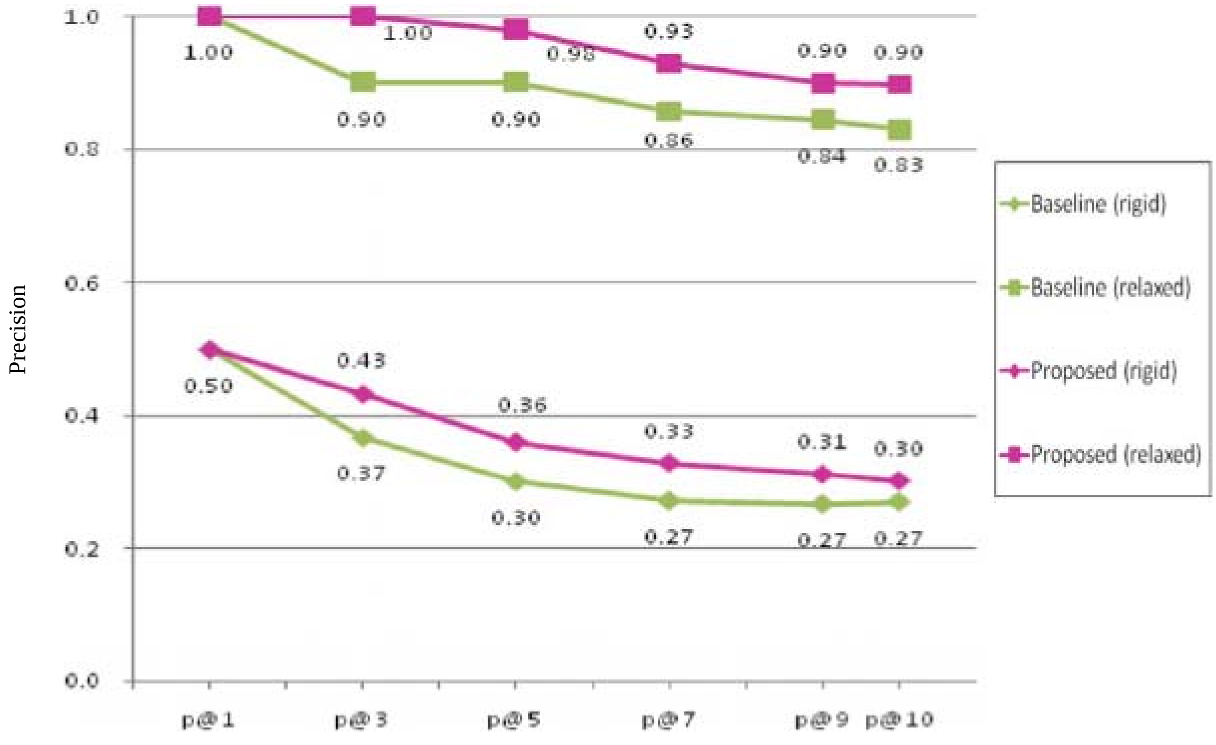


Figure 7. Comparison of proposed and baseline methods (*N* =10)

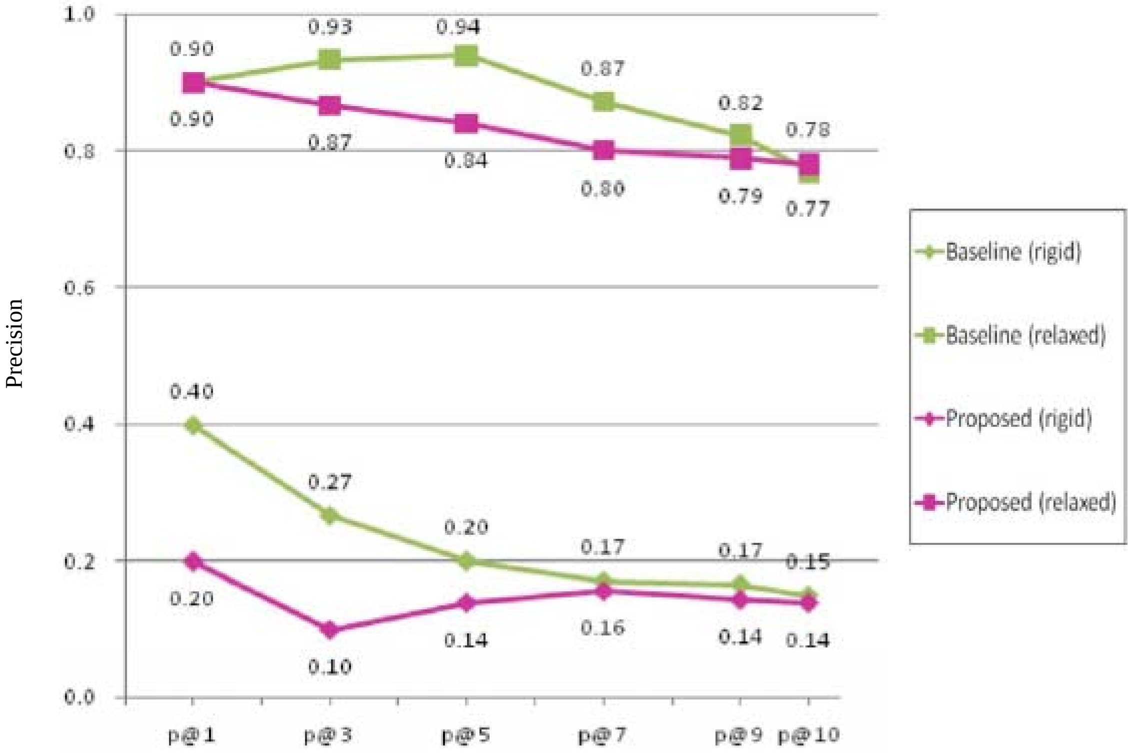


Figure 8. Comparison of proposed and baseline methods (*N* =30)

terms at the same time for Boolean search, because the proposed method as well as the other two methods collects related papers by using each technical term separately as explained in section III.

4.4 Comparison of the Proposed Method against the Baseline

We evaluated precision of highly ranked recommended papers by using p@1, p@3, p@5, p@7, and p@9 in addition to p@10 to further compare the proposed method against the baseline one. Figures. 6, 7, and 8 respectively show these precisions with *N* = 5, *N* = 10, and *N* = 30. The proposed method shows little differences among p @ *X* except for Figure 7 where the precision declines as *X* of p@*X* increases. The baseline method, on the other hand, tends to show a high precision with small *X* such as p@1 or p@3, which is especially prominent in Figure 6. That is, the baseline method can rank a few relevant papers highly and recommend them to a user while the proposed one outperforms the baseline when recommending more papers, say, 10 papers.

5. Summary

We proposed a method to recommend related papers to support online-browsing of research papers. The proposed method generates a bipartite graph by assuming linkage between related papers retrieved by the technical terms extracted from abrowsed paper, and technical terms appearing in the set of the related papers. It applies the HITS algorithm to analysis of the bipartite graph, then ranks the related papers in accordance with the hub scores assigned to the papers, and recommends top-ranked papers. We evaluated the precision of recommended papers in an experiment and showed that the proposed method could recommend relevant papers selected from a relatively small set of related papers more precisely than the other recommendation methods.

Although we only used titles and abstracts of papers in the experiment, we plan to use their whole contents to extract useful information such as bibliography. In such a case, citation information listed in the references is considered to be especially useful for paper recommendation [11]. As another venue for future work, we aim to embed the proposed functions into existent document browsers on the basis of the findings for the developed prototype browser for OCRed research papers.

6. Acknowledgements

We thank Wesco Scientific Promotion Foundation for its financial support. We also thank the Japan Society for the Promotion of Science (JSPS) for their Grant-in-Aid for Young Scientists (B), No. 23700119.

References

[1] Bunke, H., Wang ed, P. S. P. (1997). Handbook of character recognition and document image analysis, World Scientific.
[2] Ohta, M., Hachiki, T., Takasu, A. (2009). Using Web resources for support of online-browsing of research papers, *In:* Proc. of IRI, p. 348– 353.
[3] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM, 46(5), p. 604–632, 1999.
[4] Sugiyama, K., Kan, M, Y. (2010). Scholarly paper recommendation via user's recent research interests, *In:* Proc. of the 10th annual joint conference on Digital libraries, p. 29–38.
[5] Ekstrand, M. D., Kannan, P., Stemper, J. A., Butler, J. T., Konstan, J. A., Riedl, J.(2010). Automatically building research reading lists, *In:* Proc. of RecSys 2010, p. 159–166.
[6] Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank citation ranking: bringing order to the Web, Technical Report, Stanford InfoLab.
[7] Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W., Giles, C. L. (2008). Real-time automatic tag recommendation, *In:* Proc. of SIGIR, p. 515–522.[8] Nanba, H., Okumura, M. (2005). Automatic detection of survey articles, *In:* Proc. of ECDL, p. 391–401.
[9] Sen Project, http://ultimania.org/sen/
[10] Yahoo!JAPAN, http://search.yahoo.co.jp/
[11] Shi, X., Leskovec, J., McFarland, D. A. (2010). Citing for high impact, *In:* Proc. of the 10th annual joint conference on Digital libraries, p. 49–58, 2010.

A Step Towards Ambiguity Less Natural Language Software Requirements Specifications

Ashfa Umber[1], Imran Sarwar Bajwa[2]
[1] Department of Computer Science & IT
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
[2] School of Computer Science
University of Birmingham
Birmingham, UK
ashfaumber@yahoo.com, i.s.bajwa@cs.bham.ac.uk

ABSTRACT: *In modern software engineering practice, the ability to specify ambiguity less software requirements in a natural language (NL) in a seamless way is highly valuable and desirable. Though, the software requirements are typically captured in natural languages (NL) such as English, there is a very high probability that more than half NL requirements can be ambiguous. For example, Mich identified that approx. 72% of the NL requirements are potentially ambiguous. A primary reason of such ambiguous NL requirements is syntactic and semantic ambiguities in a natural language such as English. A problem with ambiguous NL requirements is that a software engineer can miss-interpret requirements and can generate an erroneous and absurd software model. In this paper, we aim to address this challenge by presenting a novel approach that is based a semantically controlled NL representation for software requirements. To generate a semantically controlled NL representation, we propose the use of Semantic of Business Vocabulary and Rules (SBVR) standard. We solve a case study to bear out that a SBVR based controlled representation can not only help in generating accurate and consistent software models but can also simplify the machine processing of requirements. The results show that our approach can be helpful in generating the accurate and consistent software models from NL software requirements. A Java implementation of the used approach is also presented a proof of concept that is also available as an Eclipse plugin.*

Keywords: Software Requirements Specifications, Ambiguity in NL Specifications, SBVR

Received: 1 September 2011, Revised 23 November 2011, Accepted 30 November 2011

1. Introduction

It is a typical practice that software requirements are specified in natural languages (NL). It is a common knowledge that 71.80% of the software requirements specifications are captured in NL [1]. However, the natural languages are intrinsically ambiguous. For automated software modeling, impervious and explicit software requirements are a primary necessity as computers cannot accurately process ambiguous requirements. A few scientists have proposed various approaches to identify and measure the typical ambiguities in NL based software requirements specifications (SRS) e.g. Kiyavitskaya et al. [3] presented a couple of tools to identify ambiguous sentence in a NL SRS document and find the reason of ambiguity. Similarly, Popescu et al. presented a tool *Dowser* [4] to identify ambiguous and inconsistent sentences in a NL SRS. However, a drawback of the used approach is that input should be in a constrained language, and this pitfall makes the approach impractical. According to our knowledge, there is no appropriate approach or tool that can provide an automatic procedure of minimizing or removing ambiguity in NL SRS.

In this paper, we aim to present an approach capable of automatically generating an unambiguous and semantically consistent representation of SRS specified in English language. To achieve a semantically controlled representation, we propose the use of Semantic of Business Vocabulary and Rules (SBVR) 1.0 [4]. SBVR is an OMG standard, initially presented to assist business requirements specifiers and analyzers. In [5] and [18] we presented that similar to business requirement, the software requirements can be captured and specified using SBVR syntax. In this paper, we propose the use of SBVR to overcome the typical ambiguities in a natural language. The SBVR incorporate not only ability of generating accurate and consistent software representation but also provides capability of machine processing as SBVR is based on mathematical or higher order logic [4]. The presented approach is also implemented in Java. The performance of the tool is evaluated by solving a case study, presented in section 4. The remaining paper is structured into the following sections: Section 2 states preliminaries of the presented research. Section 3 presents the framework for translation of English to SBVR representation. Section 4 presents a case study. The evaluation of our approach is presented in section 5. Finally, the paper is concluded to discuss the future work.

2. Semantic Business Vocabulary and Rules (SBVR)

In 2008, OMG presented a new standard *Semantic Business Vocabulary and Rules* (SBVR) [4]. SBVR supports capturing of requirement in a controlled natural language. There are various controlled natural languages such as Attempto but we have used SBVR due to following reasons:

- SBVR is a standard. Latest available version is 1.0.
- SBVR is easy to read and understand for human beings as SBVR uses syntax of natural languages e.g. English.
- SBVR is easy to machine translate as it is based on higher logic such as First Order Logic (FOL).

A typical SBVR representation is based on a set of SBVR business vocabulary and SBVR business rules in particular business domain.

2.1 SBVR Business Vocabulary

A business vocabulary [4] (section: 8.1) consists of all the specific terms and definitions of concepts used by an organization or community in course of business. In SBVR, A concept can be a noun concept or fact type. Noun concepts can be further categorized into object type, individual concept, and characteristic. Hence we have four key elements in SBVR:

- In SBVR, an object type is a general concept that exhibits a set of characteristics to distinguishes that object type from all other object types" [3] (section: 8.1) e.g. robot, user, etc.
- In SBVR, an individual noun is a qualified noun that corresponds to only one object [3] (section: 8.1) e.g. 'Robby', a famous robot.
- In SBVR, characteristic is an abstraction of a property of an object [4] (section: 8.1) e.g. name of robot is Robby, here name is characteristic.
- In SBVR, a fact type or a verb concept [4] (section: 8.1) specifies the relationships among noun concepts e.g. car has wheels. A fact type can be binary fact type e.g. "customer *places* orders".

2.2 SBVR Business Rules

A SBVR business rule is a formal representation under business jurisdiction 'Under business jurisdiction' [4]. Each SBVR business rule is based on at least one fact type.

- The SBVR rules can be a structural business rule [4] (section: 12.1) those are used to define an organization's setup.
- Another type of SBVR rules *is* a behavioural business rule [4] (section: 12.1) and they are used to express the conduct of a business entity.

2.3 SBVR based Controlled Representation

SBVR was originally presented to assist business people in creating clear and unambiguous business policies and rules in their native language [4]. The following characteristics of SBVR can help in generating a controlled representation of English:

2.3.1 Rule-based Conceptual Formalization

SBVR standard provides a rule-based conceptual formalization that can be employed to generate a syntactically formal representation of English. SBVR contains a vocabulary for conceptual modeling and captures expressions based on this vocabulary as formal logic structures. The SBVR vocabulary can be used to formally specify representations of concepts, definitions, instances, and rules of any knowledge domain in natural language. These features make SBVR well suited for describing business domains and software requirements to implement software models.

2.3.2 Natural Language Semantic Formulation

SBVR is typically proposed for business modeling in NL. However, we are using the formal logic based nature of SBVR to semantically formulate the English software requirements statements. A set of logic structures called semantic formulations are provided in SBVR to make English statements controlled such as atomic formulation, instantiate formulation, logical formulation, quantification, and modal formulation.

2.3. 3 SBVR Formal Notation

Structured English is one of the possible SBVR notations, given in SBVR 1.0 document, Annex C [4], is applied by prefixing rule keywords in a SBVR rules. The other possible SBVR notation is Rulespeak, given in SBVR 1.0 document, Annex F [4], uses mixfixing keywords in propositions. SBVR formal notations help in expressing propositions with equivalent semantics that can be captured and formally represented as logical formulations.

3. Translating NL to SBVR

This section briefly explains how English text is mapped to SBVR representation and object oriented information is extracted from SBVR representation. Figure 1 show the used approach that works in three phases:

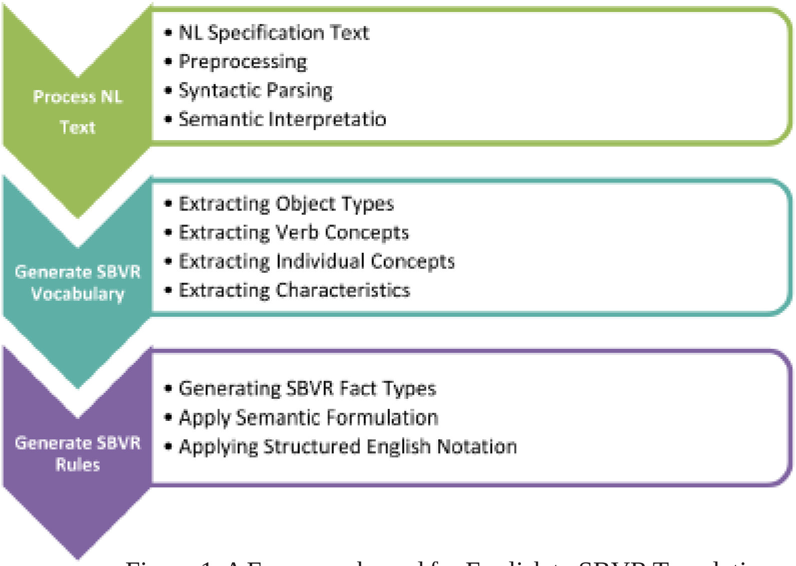

Process NL Text
• NL Specification Text
• Preprocessing
• Syntactic Parsing
• Semantic Interpretatio
Generate SBVR Vocabulary
• Extracting Object Types
• Extracting Verb Concepts
• Extracting Individual Concepts
• Extracting Characteristics
Generate SBVR Rules
• Generating SBVR Fact Types
• Apply Semantic Formulation
• Applying Structured English Notation

Figure 1. A Framework used for English to SBVR Translation

3.1 Parsing NL Software Requirement Text

The first phase of SR-Elicitor [5], [17] is NL parsing that involves a number of processing units (organized in a pipelined architecture) to process complex English statements. The NL parsing phase lexically, syntactically and semantically processes the English text as following:

3.1.1 Lexical Processing

The NL parsing starts with the lexical processing of a plain text file containing English software requirements specification. The lexical processing initiates with the tokenization of the input English text.The tokenized text is further passed to Stanford parts-of- speech (POS) [13] tagger v3.0 to identify the basic POS tags e.g.

3.1.2 Syntactic Processing

We have used an enhanced version of a rule-based bottom-up parser for the syntactic analyze of the input text used in [11]. English grammar rules are base of used parser. The text is syntactically analyzed and a parse tree is generated for further semantic processing as shown in Figure 2.

A task is a component of the schedule with a start and end date.

[A/DT] [task/NN] [is/VBZ] [a/DT] [component/NN] [of/IN] [the/DT] [schedule/NN] [with/IN] [a/DT] [start/NN] [and/CC] [end/NN] [date/NN].

Figure 2. A Framework used for English to SBVR Translation

A task is a component of the schedule with a start and end date.

```
(ROOT
 (S
  (NP (DT A) (NN task))
  (VP (VBZ is)
   (NP      (NP (DT a) (NN component))
   (PP (IN of)
    (NP
     (NP (DT the) (NN schedule))
     (PP (IN with)
      (NP (DT a) (NN start)
        (CC and)
       (NN end) (NN date))))))))
```

Figure 3. Parsing English text using Stanford Parser

3.1.3 Semantic Interpretation

In this semantic interpretation phase, role labeling [12] is performed. The desired role labels are actors (nouns used in subject part), co-actor (additional actors conjuncted with ‘and’), action (action verb), thematic object (nouns used in object part), and a beneficiary (nouns used in adverb part) if exists. These roles assist in identifying SBVR vocabulary.

3.2 Extracting SBVR Vocabulary

In this phase, the basic SBVR elements e.g. noun concept, individual concept, object type, verb concepts, etc are identified from the English input that is preprocess in the previous phase. The extraction of various SBVR elements is described below:

3.2.1 Extracting Object Types

All common nouns (actors, co-actors, thematic objects, or beneficiaries) are represented as the object types or general concept (see figure 3) e.g. belt, user, cup, etc. In conceptual modelling, the object types are mapped to classes.

3.2.2 Extracting Individual Concepts

All proper nouns (actors, co-actors, thematic objects, or beneficiaries) are represented as the individual concepts.

3.2.3 Extracting Fact Types

The auxiliary and action verbs are represented as verb concepts. To constructing a fact types, the combination of an object type/ individual concept + verb forms a unary fact type e.g. “vision system senses”. Similarly, the combination of an object type/ individual concept + verb + object type forms a binary fact type e.g. belt conveys part is a binary fact type.

3.2.4 Extracting Characteristics

In English, the characteristic or attributes are typically represented using is-property-of fact type e.g. “name is-property-of customer”. Moreover, the use of possessed nouns (i.e. pre-fixed by’s or post-fixed by of) e.g. student’s age or age of student is also characteristic.

3.2.5 Extracting Quantifications

All indefinite articles (a and an), plural nouns (prefixed with s) and cardinal numbers (2 or two) represent quantifications.

3.2.6 Extracting Associative Fact Types

The associative fact types [4] (section 11.1.5.1) are identified by associative or pragmatic relations in English text. In English, the binary fact types are typical examples of associative fact types e.g. “The belt conveys the parts”. In this example, there is a binary association in belt and parts concepts. This association is one-to-many as ‘parts’ concept is plural. In conceptual modeling of SBVR, associative fact types are mapped to associations.

3.2.7 Extracting Partitive Fact Type

The partitive fact types [4] (section 11.1.5.1) are identified by extracting structures such as “is-part-of”, “included-in” or “belong-to” e.g. “The user puts two-kinds-of parts, dish and cup”. Here ‘parts’ is generalized form of ‘dish’ and ‘cup’. In conceptual modeling of SBVR, categorization fact types are mapped to aggregations.

3.2.8 Extracting Categorization Fact Types

The categorization fact types [4] (section 11.1.5.2) are identified by extracting structures such as “is-category-of” or “is-type-of”, “is-kind-of” e.g. “The user puts two-kinds-of parts, dish and cup”. Here ‘parts’ is generalized form of ‘dish’ and ‘cup’. In conceptual modeling of SBVR, categorization fact types are mapped to generalizations. All the extracted information shown in figure 4 is stored in an arraylist for further analysis.

A task is a component of the schedule with a start and end date.

[A] [task/object_type] [is/verb_concept] [a] [component/characteristic] [of] [the] [schedule/object_type] [with] [a] [start/object_type] [and] [end_date/object_type].

Figure 4. Semantic interpretation of English text

3.3 Generating SBVR Rules

In this phase, a SBVR representation such as SBVR rule is generated from the SBVR vocabulary in previous phase. SBVR rule is generated in three phases as following:

3.3.1 Extracting SBVR Requirements

To generate a rule from an English statement, it is primarily analyzed that it is a structural requirement or a behavioural requirement. Following mapping rules are used to classify a constraint type.

3.3.2 Extracting Structural Requirements

The use of auxiliary verbs such as ‘can’, ‘may’, etc is identified to classify co requirement as a structural requirement. The sentences representing state e.g. “Robby is a robot” or possession e.g. “robot has two arms” can be categorized as structural requirements. Moreover, the general use of action verbs e.g. consists, composed, equipped, etc also represent a structural requirement.

3.3.3 Extracting Behavioural Requirements

The use of auxiliary verbs such as ‘should’, ‘must’ are identified to classify requirement as a behavioural rule. Moreover, the use of action verb can be categorized as a behavioural rule e.g. “robot picks up parts”.

3.3.4 Applying Semantic Formulation

A set of semantic formulations are applied to each fact type to construct a SBVR rule. There are five basic semantic formulations proposed in SBVR version 1.0 [12] but we are using following three with respect to the context of the scope of proposed research:

a. Logical Formulation:

A SBVR rule can be composed of multiple fact types using logical operators e.g. AND, OR, NOT, implies, etc. For logical formulation, the tokens ‘not’ or ‘no’ are mapped to negation (# a). Similarly, the tokens ‘that’ and ‘and’ are mapped to conjunction (a → b). The token ‘or’ is mapped to disjunction (a → b) and the tokens ‘imply’, ‘suggest’, ‘if, ‘infer’ are mapped to implication (a ⇒ b).

b. Quantification:

Quantification [13] is used to specify the scope of a concept. Quantifications are applied by mapping tokes like “more than” or “greater than” to at least n quantification; token “less than” is mapped to at most n quantification and token “equal to” or a positive statement is mapped to exactly n quantification.

c. Modal Formulation:

In SBVR, the modal formulation [13] specifies seriousness of a constraint. Modal verbs such as ‘can’ , ‘’ or ‘may’ are mapped to possibility formulation to represent a structural requirement and the modal verbs ‘should’, ‘must’ or verb concept “have to” are mapped to obligation formulation to represent a behavioural requirement.

4. A Case Study

To demonstrate the potential of our tool SR-Elicitor, a small case study is discussed from the domain of office time management system. This case study is online available. Following is a part of the problem statement for the case study, solved in the thesis to test SR-Elicitor.

> “*The two main functions of the time Monitor software system are to allow the developers to use a www browser to store timestamp records in a database, and to allow a manager to analyze these timestamp records. A timestamp record consists of the time duration consists of the duration of a specific activity with the unique identification. The unique identification is made of three components: the project, the user and the date when the activity is taken place. The description of an activity is divided into three components: a task name, an activity, and an artefact. For managerial purpose it is often useful to define the date in term of the current week. The current week is defined as the week starting on the Monday immediately preceding the current day of the week, and ending on the Sunday immediately following the current day of the week, inclusively. A task is unit of work defined by the manager and for which the developer is accountable. A task is a component of the schedule with a start and end date. Examples of task are Implement module A, Design library XYZ. Developers usually work with on assigned tasks. One developer may work on many tasks and a given task may involve many developers*.”

The problem statement of the second case study was given as input (NL specification) to the SR-Elicitor tool. The SR-Elicitort parses the text first, which includes lexical processing (Tokenization, Sentence splitting, POS tagging and Morphological

Category	Count	Details
Object Types	18	time_monitor_software, developer, user, task, www_browser, component, date, timestamp_record, activity, identification, project, name, artifact, week, manager, schedule, start_date, end_date
Verb Concepts	15	allow, use, store, analyse, consist, made, taken_place, define, decided, preceding, starting, ending, assign, work, involve
Individual Concepts	04	Monday ,Sunday, Implement Module A, Design library XYZ,
Characteristics	06	function, duration, description, term, day, unit, component,
Quantifications	04	Two, three, one, many
Unary Fact Types	03	activity *take place*, *define* date, current week *defined*
Associative Fact Types	07	time monitor software *allow* developer, developer *use* www browser, www browser store timestamp record, manager *analyze* timestamp record, week *start on* Monday, identification *made of* components, work *defined by* manager, developer *work with* task,
Partitive fact Types	02	timestamp record *consists of* time duration, time duration *consists of* activity,
Categorization Fact Types	03	activity *divided in* components, task *is unit of* work, *task is component of* schedule

Table 1. SBVR vocabulary generated from English text

analysis) syntactic analysis and semantic analysis. It extracts the SBVR vocabulary from the case study during syntactic analysis of text which includes noun concept, individual concept, object type, verb concept, characteristics, quantifications, associative fact types and partitive fact types etc. as shown in Table 1:

The Table I show the extracted SBVR elements such as 18 object types, 15 verb concepts, 4 characteristics, 3 unary fact types, 7 associative fact types, 2 partitive fact type and 3 categorization fact type. In the used case study's problem statement, there were 06 requirements as shown in table II:

#	SBVR Rules
1.	It is permissible that the two main *function* of the **time monitor software** *are* to *allow* the **developer** to *use* a **www browser** to *store* **timestamp record** in a **database**, and to *allow* a **manager** to *analyze* these **timestamp record**. /.
2.	It is necessity that a **timestamp record** *consists* of the **time duration** *consists* of the *duration* of a specific **activity** with the unique **identification**./.
3.	It is necessity that the unique **identification** *is made* of three **component**: the **project**, the **user** and the **date** when the **activity** *is taken place*./.
4.	It is necessity that the *description* of an **activity** *is divided* into three **component** a **task name**, an **activity**, and an **artefact**./.
5.	It is necessity that for managerial purpose, it *is* often useful to *define* the **date** in *term* of the current **week**./.
6.	It is necessity that the **current week** *is defined* as the **week** *starting* on the ‘Monday’ immediately *preceding* the current *day* of the **week**, and *ending* on the ‘Sunday’ immediately *following* the current *day* of the **week**, inclusively ./.
7.	It is necessity that a **task** *is unit* of **work** *defined* by the **manager** and for which the **developer** *is accountable*./.
8.	It is necessity that a **task** *is* a *component* of the **schedule** with a **start** and **end date**./.
9.	It is permissible that the example of **task** *are* ‘Implement module A’, ‘Design library XYZ’ ./.
10.	It is possibility that the **developer** usually *work* with one *assigned* **task** ./.
11.	It is possibility that One *may work* on many **task** and a *given* **task** *may involve* many **developer**./.

Table 2. SBVR Rule representation of software requirements

There are 11 requirements processed by the SR-Elicitor (see Table 2). According to SBVR structured English the object types are underlined e.g. **system**, **developer**, **manager**, **time** etc. the verb concepts are italicized e.g. *allow*, *analyze* etc. the SBVR keywords are bolded e.g. **at least**, **It is possibility** etc. the individual concepts are double underlined e.g. Monitor, Design library, etc. The characteristics are also italicized but with different colour: e.g. consist of, made of etc.

5. Evaluation

We have done performance evaluation to evaluate that how accurately the English specification of the software requirements has been translated into the SBVR based controlled representation by our tool ER-Elicitor. An evaluation methodology, for the performance evaluation of NLP tools, is used that was originally proposed by Hirschman and Thompson [14]. The used performance evaluation is typically based on three aspects:

There were seven sentences in the used case study problem. The largest sentence was composed of 39 words and the smallest sentence contained 10 words. The average length of all sentences is 24. The major reason to select this case study was to test our tool with the complex examples. The correct, incorrect, and missing SBVR elements are shown in Table 3.

Results of each SBVR element describe in above table separately. According to our evaluation methodology, table shows sample elements are 68 in which 59 are correct, 7 are incorrect and 4 are missing SBVR elements.

The following table describes the Recall and precision of SR-elicitor for NL- software requirements. In Table 4, the average recall for SBVR software requirement specification is calculated 86.76% while average precision is calculated 89.39%. Considering the lengthy input English sentences including complex linguistic structures, the results of this initial performance evaluation are very encouraging and support both the approach adopted in this paper and the potential of this technology in general.

#	**Type/Metrics**	**N_{sample}**	**$N_{correct}$**	**$N_{incorrect}$**	**$N_{missing}$**
1	Object Types	18	16	1	1
2	Verb Concepts	16	14	2	0
3	Individual Concepts	05	04	1	0
4	Characteristics	07	06	2	1
5	Quantifications	06	04	0	2
6	Unary Fact Types	03	03	0	0
7	Associative Fact Types	08	07	1	0
8	Partitive fact Types	02	02	0	0
9	Categorization Fact Types	03	03	0	0
	Total	68	59	7	4

Table 3. Results of NL to SBVR Translation by SR-Elicitor

Type/Metrics	***N_{sample}***	***$N_{correct}$***	***$N_{incorrect}$***	***$N_{missing}$***	***Rec%***	***Prec%***	***F-Value***
Requirements	68	59	7	4	86.76	89.39	88.05

Table 4. Recall and Precision of SR-Elicitor for NL software requirements for case study

Four other case studies were solved in addition to the case study presented in section 4. All the case studies were unseen. The solved case studies were of different lengths. The largest case study was composed of 143 words and 13 sentences. The smallest case study was composed of 97 words and 8 sentences. Calculated recall, precision and f-values of the solved case studies are shown in Table 5.

Input	**N_{sample}**	**$N_{correct}$**	**$N_{incorrect}$**	**$N_{missing}$**	**Rec%**	**Prec%**	**F-Value**
C1	48	37	8	3	77.08	82.22	79.65
C2	43	33	8	2	76.74	80.48	78.61
C3	39	31	5	3	79.48	86.11	82.79
C4	36	29	3	4	80.55	90.62	85.58
C5	68	59	7	4	86.76	89.39	88.05
Average					80.12	85.76	82.94

Table 5. Evaluatin results of SR-Elicitor

The average F-value is calculated 82.94% that is encouraging for initial experiments. We cannot compare our results to any other tool as no other tool is available that can generate SBVR-based SRS from NL specification. However, we can note that other language processing technologies, such as information extraction systems, and machine translation systems, have found commercial applications with precision and recall figure well below this level. Thus, the results of this initial performance evaluation are very encouraging and support both ER-Elicitor approach and the potential of this technology in general. Figure 5 shows the evaluation results of SR-Elicitor.

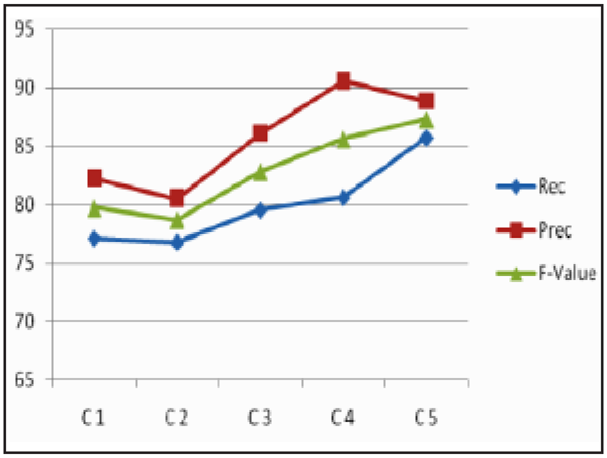


Figure 5. Semantic interpretation of English text

Figure 5 materializes the results of Recall, precision and F-Value that is obtained by five different case studies. According to our results C5 has high Recall, precision and F-Value. Moreover, C1 has lowest Recall, precision and F-Value.

6. Conclusion and Future Work

The primary objective of the paper was to address the ambiguous nature of natural languages (such as English) and generate a controlled representation of English so that the accuracy of machine processing can be improved. To address this challenge we have presented a NL based automated approach to parse English software requirement specifications and generated a controlled representation using SBVR. Automated object oriented analysis of SBVR based software requirements specifications using SR-Elicitor provides a higher accuracy as compared to other available NL-based tools.

The future work is to extract the object-oriented information from SBVR specification of software requirements such as classes, instances and their respective attributes, operations, associations, aggregations, and generalizations. Automated extraction of such information can be helpful in automated conceptual modelling of natural language software requirement specification.

References

[1] Mich Luisa , Franch Mariangela , Inverardi Pierluigi, (2004). Market research for requirements analysis using linguistic tools, Requirements Engineering, 9 (1) 40-56, February.

[2] Kiyavitskaya, N., Zeni, N., Mich, L., Berry, D. M. (2008). Requirements for tools for ambiguity identification and measurement in natural language requirements specifications, Requirements Engineering, 13 (3) 207-239, August.

[3] Popescu, D., Rugaber, S., Medvidovic, N.et al. (2007). Reducing Ambiguities in Requirements Specifications Via Automatically Created Object-Oriented Models, Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs: 14th Monterey Workshop 2007, Monterey, CA, USA, September, p.10-13.

[4] OMG, (2008). Semantics of Business vocabulary and Rules. (SBVR) Standard v.1.0. Object Management Group, Available: http://www.omg.org/spec/SBVR/1.0/

[5] Ashfa Umber, Imran Sarwar Bajwa, Asif Naeem, M. (2011). NL-Based Automated Software Requirements Elicitation and Specification. 1st International Conference on Advances in Computing and Communications (ACC-2011), Kerala, India, p.30-39

[6] Spreeuwenberg, S., Healy, K. A. (2010). SBVR's Approach to Controlled Natural Language. CNL 2009 Workshop, LNCS Volume 5972, p.155-169

[7] Ilieva, M. G., Ormandjieva, O. (2005). Automatic Transition of Natural Language Software Requirements Specification into Formal Presentation. *In:* NLDB, p.392-397

[8] Toutanova. K., Manning, C.D. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. *In:* the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. 63-70. Hong Kong.

[9] Fuchs, N. E., Kaljurand, K., Kuhn, T. (2008). Attempto Controlled English for Knowledge Representation. *In:* Reasoning Web, LNCS, V. 5224, 104–124.

[10] White, Colin, Rolf Schwitter. (2009). An Update on PENG Light. *In:* Proceedings of ALTA, p. 80–88.

[11] Clark, P., Harrison, P., Murray, W. R, Thompson, J. (2009). Naturalness vs. Predictability: A Key Debate in Controlled Languages. *In:* Proceedings 2009 Workshop on Controlled Natural Languages (CNL'09).

[12] Martin, P. (2002). Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. *In:* Proceedings of ICCS LNAI, V. 2393, p. 77–91.

[13] Schwitter, R. (2010). Controlled Natural Languages for Knowledge Representation, Coling, Poster Volume, Beijing, 1113–1121

[14] Huijsen, W. O. (1998). Controlled Language –An Introduction. *In:* Proceedings of CLAW 98:1–15.

[15] Hirschman, L., Thompson, H. S. (1995). Chapter 13 evaluation: Overview of evaluation in speech and natural language processing. *In:* Survey of the State of the Art in Human Language Technology.

[16] Umber, Ashfa., Bajwa, Imran Sarwar. (2011). Minimizing Ambiguity in Natural Language Software Requirements Specification, *IEEE Sixth International Conference on Digital Information Management* (ICDIM 2011) Melbourne, Australia

[17] Bajwa, Imran Sarwar., Mark G. Lee, Behzad Bordbar, (2011). SBVR Business Rules Generation from Natural Language Specification *In:* AAAI Spring Symposium – Artificial Intelligence 4 Business Agility, San Francisco, USA, p.541-545

Two Algorithms for Web Applications Assessment

Stavros Valsamidis, Sotirios Kontogiannis, Alexandros Karakos
Department of Electrical and Computer Engineering
Democritus University of Thrace
Xanthi, Greece

ABSTRACT: *The usage of web applications can be measured with the use of metrics. In a LMS, a typical web application, there are no appropriate metrics which would facilitate their qualitative and quantitative measurement. The purpose of this paper is to propose the use of existing techniques with a different way, in order to analyze the log file of a typical LMS and deduce useful conclusions. Three metrics for course usage measurement are used. It also describes two algorithms for course classification and suggestion actions. Although the case study concerns a LMS it can also be applied to other web applications such as e-government, e-commerce, e-banking, blogs e.t.c.*

Keywords: Learning Management Systems, Learning Web Applications, LMS Algorithms, Web Usage Management

Received: 12 August 2011, Revised 7 November 2011, Accepted 15 November 2011

1. Introduction

Learning Management Systems (LMSs) are extensively used nowadays and they provide a variety of information and communication channels for the users [19]. Among the features they provide are the development, management, distribution, diffusion and presentation of the educational material as well as tools for the management of users and courses. One of the main problems of the LMSs is the lack of exploitation of the acquired information due to its volume. Most of the times, these systems produce certain reports with statistical data, which however, do not help instructors to draw out useful conclusions either for the course or for the students and are useful only for the administrative purposes of each LMS.

Some of the most well known commercial LMS platforms used for educational purposes worldwide are Blackboard, WebCT and TopClass, while Claroline, Moodle, Ilias and aTutor are freely distributed under appropriate licenses [14]. In Greece, the Greek University Network (GUNet) uses the platform Open eClass [6], which is an evolution of Claroline [4]. This system is an asynchronous distant education platform, which uses Apache as a web server, MySQL as database server and has been implemented in PHP.

Server log files store information containing the page requests of each individual user [20]. Data mining techniques have been used to discover the sequence patterns of students' web usage after the analysis of log files data [16]. The extraction of sequential patterns has been proven to be particularly useful and has been applied to many different educational tasks [15].

The objectives of this paper are the analysis log file of a typical LMS and deduce useful conclusions. Metrics and algorithms for classification and suggestion, which were firstly introduced by the authors, are also used [21-22]. Finally two algorithms for classification of the courses and suggestions to the users are presented.

The chapter is organized as follows. Section 2 describes the background theory. Section 3 describes the logging data and pre-processing procedure. Section 4 describes the data processing procedure with the introduced metrics. Section 5 describes two algorithms. Section 6 presents the conclusions along with future directions.

2. Background

There are several studies that show the impact of data mining on eLearning. While data mining methods have been systematically used in a lot of e-commercial applications, their utilization is still lower in the LMSs [24]. It is important to notice that traditional educational data sets are normally small [7], if we compare them to files used in other data mining fields such as e-commerce applications that involve thousands of clients [17]. This is due to the typical, relatively small size of the classroom, although it varies depending on the type of the course (elementary, primary, adult, higher, tertiary, academic or/and special education); corresponding transactions are therefore also fewer. The user model is also different in both systems [16].

Very interesting is the iterative methodology to develop and carry out maintenance of web-based courses, to/in which a specific data mining step was added [5]. The proposed system finds, shares and suggests the most appropriate modifications to improve the effectiveness of the course. The discovered useful information is used directly by the educator of the course in order to improve instructional/learning performance. This system recommends the necessary improvements to increase the interest and the motivation of the students. It is well known that motivation is essential for learning: lack of motivation is correlated to learning rate decrease [3]. There are several specialized web usage mining tools that are used in the e-learning platforms. CourseVis [10] is a visualization tool that tracks web log data from an LMS. By transforming this data, it generates graphical representations that keep instructors well-informed about what precisely is happening in distance learning classes. GISMO [11] is a tool similar to CourseVis, but provides different information to instructors, such as student's details in using the course material. Sinergo/ColAT [2] is a tool that acts as an interpreter of the students' activity in a LMS. [13] describes a tool which uses log files in order to represent the instructor-student interaction in hierarchical structure. MATEP [25] is another tool acting at two levels. Firstly, it provides a mixture of data from different sources suitably processed and integrated. These data originate from e-learning platform log files, virtual courses, academic and demographic data. Secondly, it feeds them to a data webhouse which provides static and dynamic reports. Analog is another system [23] which consists of two main components. The first is performing online and the second offline data processing according to web server activity. Past users activity is recorded in server log files which are processed to form clusters of user sessions.

[18] propose a new approach for automatic user satisfaction measurement by identifying and indexing the target groups of various e-learning material like e-courses, educational games etc through the EDUSA test. Some other researchers like [1] [12] propose metrics for e-learning evaluation. In e-commerce web usage analysis some metrics are proposed by [9].

A methodology for the maintenance of web-based courses was also proposed by [8] which incorporates a specific data mining step. Publications of the authors relevant to this paper are the automated suggestions and course ranking through a web mining system [21] and the proposal of two new metrics, homogeneity and enrichment, for web applications assessment, which are also used in this chapter [22].

3. Logging the data and pre-processing

3.1 Logging the data

This stage involves the logging of specific data from LMS. Apache web server uses the following configurations for the production of its log files: *Common Log Format* (*CLF*), *Extended Log Format* (*ELF*), *Cookie Log Format* (*CKLF*) and *Forensic Log Format* (*FLF*).

A module that uses FLF format and records attributes before and after web server request processing, was implemented. FLF format is used, instead of CLF, because of the advantage that it stores server requests, before and after request processing at the web server. Such recording contains accurate information regarding user thinking time that indicates content difficulty and complexity and requests processing time that indicates content delivery effort.

In more detail, the data recording module, is embedded in the web server of the e-learning platform and records specific e-learning platform fields. Specifically, eleven (11) fields (*request_time_event*, *remote_host*, *request_uri*, *remote_logname*, *remote_user*, *request_method*, *request_time*, *request_protocol*, *status*, *bytes_sent*, *referer*, *agent*) from different courses and user requests are recorded with the use of an Apache module, developed in Perl programming language, as a first step.

The development of such a module has the following two advantages: rapid storage of user information, since it is executed straight from the server API and not by the LMS application, and the produced data are independent of specific formulation used by the LMS platform.

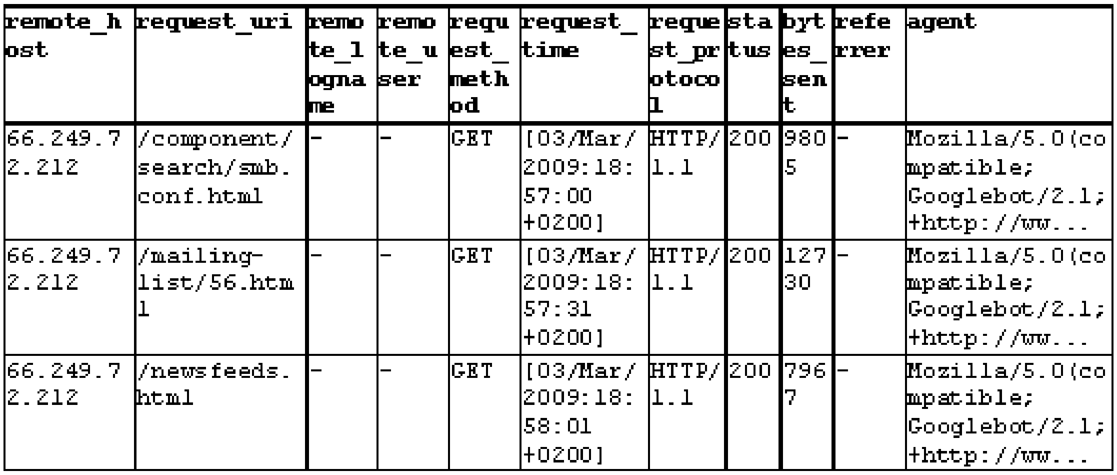
3.2 Data pre-processing

The data of the log file contain noise such as missing values, outliers etc. These values have to be pre-processed in order to prepare them for data mining analysis. Specifically, this logging data step filters the recorded data. It uses outlier detection and removes extremes. This step is not performed by the LMS platform and thus can be embedded into a variety of LMSs. Also, it facilitates data mining analysis methods construction of robust results.

The produced log file, is filtered, so it includes only the following three fields: (i) *courseID*, which is the identification string of each course; (ii) *sessionID*, which is the identification string of each session; (iii) *page Uniform Resource Locator* (*URL*), which contains the requests of each page of the platform that the user visited.

3.3 Study population and context

In detail the dataset was collected from a real LMS environment used in the Technological Education Institute (TEI) of Kavala that uses the Open eClass e-learning platform [6]. The data are from the spring semester of 2009 from the Department of Information Management and involve 1199 students and 39 different courses. The data are in ASCII form and are obtained from the Apache server log file. A view of the collected data in forensic log format is shown in table 1.



remote_host	request_uri	remote_logname	remote_user	request_method	request_time	request_protocol	status	bytes_sent	referrer	agent
66.249.72.212	/component/search/smb.conf.html	-	-	GET	[03/Mar/2009:18:57:00 +0200]	HTTP/1.1	200	9805	-	Mozilla/5.0(compatible; Googlebot/2.1; +http://ww...
66.249.72.212	/mailing-list/56.html	-	-	GET	[03/Mar/2009:18:57:31 +0200]	HTTP/1.1	200	12730	-	Mozilla/5.0(compatible; Googlebot/2.1; +http://ww...
66.249.72.212	/newsfeeds.html	-	-	GET	[03/Mar/2009:18:58:01 +0200]	HTTP/1.1	200	7967	-	Mozilla/5.0(compatible; Googlebot/2.1; +http://ww...

Table 1. eClass data in FLF

The log file which is produced, from the previous step, is filtered and pre-processed in order to include the following fields: courseID, sessionID and page Uniform Resource Locator (URL).

4. Processing the Data

The aforementioned fields of the previous section are not adequate in order to evaluate the course usage. So, some metrics are used for the facilitation of the course usage evaluation (Table 2). First, the indexes Sessions, Pages, Unique pages, Unique Pages per CourseID per Session are computed with the use of a Perl program. Then, the metrics Enrichment, Disappointment, Interest and Homogeneity are calculated.

The number of the sessions and the number of the pages viewed by all users are counted for the calculation of course activity. The metric unique pages measures the total number of unique pages per course viewed by all users. The Unique Pages per Course per Session (UPCS) metric expresses the unique user visits per course and per session; it is used for the calculation of the course activity in an objective manner. Because some novice users may navigate in a course and visit some pages of the course more than once, UPCS eliminates duplicate page visits, since it considers the visits of the same user in a session only once.

Enrichment is a metric which is proposed in order to express the "*enrichment* " of each course in terms of educational material.

Enrichment is defined as the complement of the ratio of the unique pages over total number of course web pages as it was proposed in [21-22].

Index/Metric name	**Description of the index/metric**
Sessions	The total number of sessions per course viewed by users
Pages	The total number of pages per course viewed by users
Unique pages	The total number of unique pages per course viewed by users
Unique Pages per CourseID per Session (UPCS)	The total number of unique pages per course per session viewed by users
Enrichment	The enrichment of courses
Disappointment	The disappointment of users
Interest	It is the one 's complement to the disappointment
Homogeneity	Homogeneity of courses

Table 2. Metrics name and description

The number of the *sessions* and the number of the *pages* viewed by all users are counted for the calculation of course activity. The metric *unique pages* measures the total number of unique pages per course viewed by all users. The *Unique Pages per Course per Session* (*UPCS*) metric expresses the unique user visits per course and per session; it is used for the calculation of the course activity in an objective manner. Because some novice users may navigate in a course and visit some pages of the course more than once, UPCS eliminates duplicate page visits, since it considers the visits of the same user in a session only once.

Enrichment is a metric which is proposed in order to express the "enrichment" of each course in terms of educational material. Enrichment is defined as the complement of the ratio of the unique pages over total number of course web pages as it was proposed in [21-22].

$$\textit{Enrichment} = 1\text{- } (\textit{Unique Pages/Total Pages}) \tag{1}$$

where Unique Pages<=Total Pages.

Enrichment values are in the range [0, 1). When users follow unique paths in a course this is 0 while in a course with minimal unique pages this is close to 1. Since it offers a measure of how many unique pages were viewed by the users, it shows how much information included in each course is handed over to the end user inferring that the course contains rich educational material.

Disappointment is a metric which combines sessions and pages viewed by users and it measures the disappointment of the users in the course, in the sense that when a user views few pages of the course, s/he logs out of the course. Disappointment metric is defined as the rate of sessions per LMS course to total number of course web pages.

$$\textit{Disappointment} = \textit{Sessions/Total Pages} \tag{2}$$

In other words, the disappointment metric reflects how quickly the users discontinue viewing pages of the courses. Disappointment values are in the range (0, 1]. Due to the negative nature of the Disappointment metric, it was replaced by another metric which has positive sounding manner, *Interest*. Interest metric is defined as the one's complement to the disappointment.

$$\textit{Interest} = 1\text{-}\textit{Disappointment} \tag{3}$$

Both disappointment and interest metrics were proposed in [21]. A low interest in a course means that there are not many unique pages viewed per session; therefore the course is not so popular among the students. This may be so either because students were not pleased with the educational material or there are not many pages to visit. High interest indicates that users are interested in course content and continue further with their study. When the quality of the educational material does not fulfil user requirements, the user is led to log out of the course.

Homogeneity metric is another metric that is defined as the ratio of unique visited course pages to the number of sessions that visited that course.

$$Homogeneity = Unique\ pages/Total\ Sessions \qquad (4)$$

where Total Sessions per course >> Unique course pages.

Homogeneity metric value ranges from [0,1), where 0 means that no user followed a unique path and 1 that every user follows unique paths. It is a course quality index and characterizes the percentage of course information discovered by each user participating in a course. The aforementioned metrics contribute to the evaluation of courses usage. The results for the 39 courses are presented in Table 3.

5. Algorithms

In this section, two algorithms which classify the LMS courses and suggest actions to the educators for course improvement are used.

5.1 Classifier algorithm

The first algorithm classifies LMS courses based on poor or rich quantity of course information material. Afterwards, based on LMS courses with adequate information material, it tries to spot how often course information is added or updated by educators based on homogeneity classification or followed by users the updated information. Finally, using the UPCS metric it identifies whether updates of course information can increase the student's interest in the specific course. Classifier algorithm schema is depicted in Figure 1.

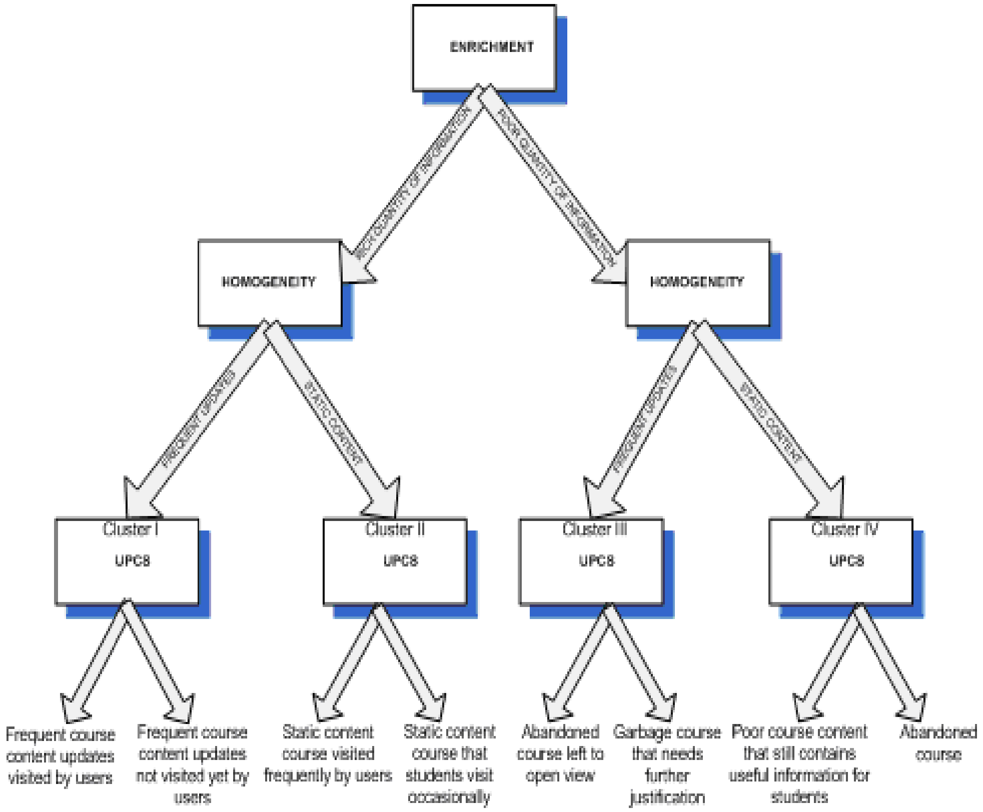

ENRICHMENT
RICH QUANTITY OF INFORMATION
POOR QUANTITY OF INFORMATION
HOMOGENEITY
HOMOGENEITY
FREQUENT UPDATES
STATIC CONTENT
FREQUENT UPDATES
STATIC CONTENT
Cluster I
UPCS
Cluster II
UPCS
Cluster III
UPCS
Cluster IV
UPCS
Frequent course content updates visited by users
Frequent course content updates not visited yet by users
Static content course visited frequently by users
Static content course that students visit occasionally
Abandoned course left to open view
Garbage course that needs further justification
Poor course content that still contains useful information for students
Abandoned course

Figure 1. Classifier algorithm schema

According to the above, the proposed algorithm is based on Enrichment, Homogeneity and UPCS and consists of the corresponding stages.

Course ID	Sessions	Pages	Unique pages	UPCS	Disappointment	Interest	Enrichment	Homogeneity
IMD115	18	73	9	42	0,247	0,753	0,877	0,500
IMD62	23	94	11	52	0,245	0,755	0,883	0,478
IMD9	26	105	12	42	0,248	0,752	0,886	0,462
IMD10	17	61	8	28	0,279	0,721	0,869	0,471
IMD111	33	142	9	79	0,232	0,768	0,937	0,273
IMD21	11	25	8	24	0,440	0,560	0,680	0,727
IMD80	14	38	7	34	0,368	0,632	0,816	0,500
IMD98	32	113	9	61	0,283	0,717	0,920	0,281
IMD17	53	206	11	89	0,257	0,743	0,947	0,208
IMD15	11	24	7	20	0,458	0,542	0,708	0,636
IMD133	25	80	7	54	0,313	0,688	0,913	0,280
IMD130	12	30	5	22	0,400	0,600	0,833	0,417
IMD35	87	338	8	179	0,257	0,743	0,976	0,092
IMD8	45	135	8	82	0,333	0,667	0,941	0,178
IMD105	91	297	11	216	0,306	0,694	0,963	0,121
IMD34	38	113	6	56	0,336	0,664	0,947	0,158
IMD36	72	217	7	134	0,332	0,668	0,968	0,097
IMD14	45	122	7	59	0,369	0,631	0,943	0,156
IMD61	32	74	9	64	0,432	0,568	0,878	0,281
IMD66	56	144	9	107	0,389	0,611	0,938	0,161
IMD64	22	47	7	39	0,468	0,532	0,851	0,318
IMD129	75	209	6	131	0,359	0,641	0,971	0,080
IMD50	22	46	7	38	0,478	0,522	0,848	0,318
IMD122	33	71	7	45	0,465	0,535	0,901	0,212
IMD112	30	62	6	46	0,484	0,516	0,903	0,200
IMD60	22	43	5	40	0,512	0,488	0,884	0,227
IMD11	51	108	6	80	0,472	0,528	0,944	0,118
IMD120	38	80	3	46	0,475	0,525	0,963	0,079
IMD49	14	23	5	21	0,609	0,391	0,783	0,357
IMD26	50	90	8	71	0,556	0,444	0,911	0,160
IMD41	98	185	8	129	0,530	0,470	0,957	0,082
IMD44	48	82	10	75	0,585	0,415	0,878	0,208
IMD125	93	164	6	134	0,567	0,433	0,963	0,065
IMD96	20	31	5	30	0,645	0,355	0,839	0,250
IMD114	28	42	4	34	0,667	0,333	0,905	0,143
IMD132	152	230	7	184	0,661	0,339	0,970	0,046
IMD67	18	23	4	22	0,783	0,217	0,826	0,222
IMD23	30	38	4	32	0,789	0,211	0,895	0,133
IMD134	25	27	4	27	0,926	0,074	0,852	0,160

Table 3. LMS data and grade for 39 Courses

In the first stage of the algorithm, the Enrichment metric is involved in order to identify courses with poor or rich educational content (poor equals to small enrichment value while rich to high enrichment value). A set of N courses are placed to an N-ordered table based on Enrichment, where N<=Total LMS platform courses, the courses with the highest Enrichment metric values.

In the second stage, the algorithm classifies the previous set of N courses using the values of Enrichment and Homogeneity. The classification of LMS courses is performed using four clusters as shown at Figure 1. The higher the Homogeneity value the more frequent the course updates or the more dynamic the course content, depending on Enrichment value. The lower the Homogeneity value then the LMS is more static in content or of poor content updates. The classification of the courses depends on the average Enrichment value of the N LMS courses and the average Homogeneity value of the high and low Enrichment clusters accordingly.

The aim of the third stage of the algorithm is to identify whether the content can be characterized as rich or poor, and whether it is static, frequent or dynamic. In order to do this, each cluster's courses are ranked based on the value of the UPCS.

5.1.1 Application of classification algorithm

The 39 courses were initially ranked according to the Enrichment metric. The algorithm was tested by picking the best and worse LMS courses from a list of 39 courses which are shown in Table 4. That is, best and worst cases from students' usage point of view.

Course ID	Sessions	Pages	Unique pages	UPCS	Homogeneity	Enrichment
IMD132	152	230	5	184	0.033	0.978
IMD35	87	338	9	179	0.103	0.973
IMD125	93	164	6	134	0.065	0.963
IMD129	75	209	8	131	0.107	0.962
IMD105	91	297	12	216	0.132	0.960
IMD41	98	185	8	129	0.082	0.957
IMD36	72	217	10	134	0.139	0.954
IMD17	*53*	*206*	*21*	*89*	*0.396*	*0.898*
IMD66	56	144	16	107	0.286	0.889
IMD8	45	135	18	82	0.400	0.867
IMD122	33	71	21	45	0.636	0.704
IMD112	30	62	20	46	0.667	0.677

Table 4. Processed data for 12 Courses with Average Enrichment value of 0.898

Based on the previous order by Enrichment Table 1 of 12 LMS courses, the Classifier algorithm was applied by using an average Enrichment value of 0.898 and average homogeneity value for the high enrichment cluster of 0.09 and for the low enrichment cluster of 0.45. The classification of the algorithm produced four clusters, which are shown in the Table 5.

As shown in Table 6, for each one of the four classes the LMS courses are ordered based on the UPCS metric value. So courses IMD105 and IMD36 are the representatives of high and low UPCS values for cluster I, IMD132 and IMD41 for cluster II, IMD112 and IMD122 for cluster III and IMD66 and IMD8 for cluster IV accordingly.

In Table 4, these courses and the classifier algorithm evaluation feedback for each one of those courses are presented.

5.2 Suggestion algorithm

The goal of the second algorithm is to allow an automated suggestions system for course improvement. The first step of the proposed algorithm is course ranking in descending order by UPCS. A course placed in the first ranking positions is a popular one, either because of exclusive quality of its educational content or quantity of course material.

The first suggestion rule (Figure 2) of the algorithm compares Interest metric of each course with the a*Average(Interest) of all LMS courses, where a is a coefficient parameter. If Interest value is lower than a*Average(Interest), it means that this course either does not have adequate educational content or its content quality does not meet (satisfy) user requirements. In order to distinguish between these two cases a new condition is applied that checks whether course Unique Pages value is less than the

Average(Unique Pages) value for all LMS courses. If this condition is fullfiled, then course content quality is in need of amelioration, while if not, course content is of fine quality and new content additions do not need to be made due to course interest expressed by users.

Enrichment Class	Homogeneity Clusters	Course ID	Sessions	Pages	Unique pages	UPCS	Homogeneity	Enrichment
High	Dynamic Content or Frequently Updated, Cluster I	*IMD105*	*91*	*297*	*12*	*216*	*0.132*	*0.960*
		IMD35	87	338	9	179	0.103	0.973
		IMD36	*72*	*217*	*10*	*134*	*0.139*	*0.954*
		IMD129	75	209	8	131	0.107	0.962
	Static Content with frequent updates, Cluster II	*IMD132*	*152*	*230*	*5*	*184*	*0.033*	*0.978*
		IMD125	93	164	6	134	0.065	0.963
		IMD41	*98*	*185*	*8*	*129*	*0.082*	*0.957*
Low	Dynamic Content with less updates, Cluster III	*IMD112*	*30*	*62*	*20*	*46*	*0.667*	*0.677*
		IMD122	*33*	*71*	*21*	*45*	*0.636*	*0.704*
	Static Content, Cluster IV	*IMD66*	*56*	*144*	*16*	*107*	*0.286*	*0.889*
		IMD17	53	206	21	89	0.396	0.898
		IMD8	*45*	*135*	*18*	*82*	*0.400*	*0.867*

Table 2. Clustering of the 12 courses based on the Classifier algorithm

Cluster ID	Course ID	CCA Evaluation
I	IMD105	**High Activity LMS with updates followed by users**
I	IMD36	**High Activity LMS with frequent educator updates that are not followed by users**
II	IMD132	**High Activity LMS with Static content, frequently updated and followed by users**
II	IMD41	**High Activity LMS with static content, frequently updated but poorly followed by users**
III	IMD112	**Garbage course or Forum with updates- Need for further evaluation**
III	IMD122	**Abandoned course of dynamic content left to open view**
IV	IMD66	**Course of poor static content that still contains information followed by users (or forced to follow)**
IV	**IMD8**	**Abandoned course of poor static content occasionally followed by curious users**

Table 3. Clustering of the 12 courses based on the classifier algorithm

```
IF Interest < a* Avetage(Interest) THEN
        IF UniquePages > Average(UniquePages)
       THEN “Improve Quality”
     ELSE IF Interest > b*Average(Interest) THEN “Add Content”
             ELSE “Add Content and Improve Quality”
ELSE IF UniquePages < AVG(UniquePages)
```

Figure 2. First Suggestion rule

Additional course quality improvements are suggested if Interest is less than *b**Average(Interest), where *b* is a coefficient parameter. Finally, if Interest is more than the *a**Average(Interest), and the number of the Unique Pages is less than the Average(Unique Pages), then there is a slight need for new content addition.

The next suggestion rule (Figure 3), applies the Enrichment metric. A low Enrichment value means that users do not visit course pages due to the lack of course content updates. If Enrichment value of a course is less than *c**Average(Enrichment), where *c* is a coefficient parameter, then the algorithm suggests that it would be a good practice for the author to update course content, so as to motivate users to re-visit his/her course.

```
IF Enrichment<c* Average(Enrichment) THEN
        "Update the course content"
```

Figure 3. Second Suggestion rule

Algorithm's *a*, *b* and *c* coefficient parameters range between 0 and 1. In order to accurately calibrate the coefficient parameters, the algorithm first applied to a reduced set of LMS courses. Course selection was performed based on best and worst case LMS courses, using UPCS ranking. Then a value was calculated by using best to worst course Interest deviation value. b coefficient value is the average best to worst course Interest value and c was calculated as the median value of the first k LMS courses based on UPCS ranging, where *k* =5:

$$c = \text{median}\,(\text{Interest}_i) - k * N * 0.0001 \qquad (5)$$

where *N* is the total number of LMS courses and *k* the number of selected courses with maximum UPCS ranking values.

5.2.1 Application of suggestion algorithm

This experiment had two goals: To determine suitable values for a, b, c parameters and to test the two suggestion rules of the algorithm with respect to their impact on improving the course quality.

The first stage of the algorithm was the ranking of the courses. Table 7, due to space limitation, displays the results for the courses in ranking positions 1-5, 21-39, using the UPCS metric. The ranking of the courses is based on: first UPCS, then Enrichment and then Interest values.

Based on the aforementioned metrics, the courses were classified using the following three steps:

1. Course ranking step: primarily, course evaluation was considered using the UPCS value and LMS courses were ranked in descending order.

2. First suggestion rule step: The first suggestion rule is used in order to evaluate course content in terms of interest as expressed by course users and provides the appropriate suggestions to the instructors, related to the quantity and the quality of their course educational content.

3. Second suggestion rule step: Course content is examined in depth in order to express whether users are satisfied from what they see or course content seems confusing or complex to the end user. Enrichment metric was used to identify courses with poor or rich educational material confirmed by users and provides suggestions for possible course updates.

In order to perform the final two stages of the algorithm, the coefficient parameters values were firstly calculated. According to the experiment outcome, the values for *a, b, c* parameters were 0.9, 0.6 and 0.95 respectively.

The second goal of the experiment was to test the suggestion rules by showing them to the course instructors and receive verification feedback on the suggestion accuracy. When instructors applied the proposed suggestions, their courses improved in UPCS ranking position.

6. Discussion and Conclusion

The proposed method uses existing techniques in a different way to perform LMS usage analysis. It uses the enrichment, homogeneity and interest metrics. It presents the clustering of students and courses. It uses two algorithms for course classification and suggestion actions.

Position	Course ID	Sessions	Pages	Unique pages	UPCS	Interest	Enrichment	Automated Suggestions		
								Add new content	Improve content quality	Update
1	IMD105	91	297	11	216	0,694	0,963			
2	IMD132	152	230	7	184	0,339	0,970	*	*	
3	IMD35	87	338	8	179	0,743	0,976			
4	IMD36	72	217	7	134	0,668	0,968	-		
5	IMD125	93	164	6	134	0,433	0,963	*		
21	IMD120	38	80	3	46	0,525	0,963	-		
22	IMD112	30	62	6	46	0,516	0,903	-		
23	IMD122	33	71	7	45	0,535	0,901	-		
24	IMD9	26	105	12	42	0,752	0,886			
25	IMD115	10	73	12	42	0,863	0,836			*
26	IMD60	22	43	5	40	0,116	0,884	*		
27	IMD64	22	47	7	39	0,149	0,851	-		
28	IMD50	22	46	7	38	0,152	0,848	-		*
29	IMD114	28	42	4	34	0,095	0,905	*	*	
30	IMD80	14	38	7	34	0,184	0,816	-		*
31	IMD23	30	38	4	32	0,105	0,895	*	*	
32	IMD96	20	31	5	30	0,161	0,839	*	*	*
33	IMD10	17	61	8	28	0,131	0,869			
34	IMD134	25	27	4	27	0,148	0,852	*	*	
35	IMD21	11	25	8	24	0,560	0,680			*
36	IMD130	12	30	5	22	0,600	0,833	-		*
37	IMD67	18	23	4	22	0,217	0,826	*	*	*
38	IMD49	14	23	5	21	0,391	0,783	*		*
39	IMD15	11	24	7	20	0,542	0,708	-		*
Average of all courses				7,25	68,41	0.55	0.892			

Table 7. Processed e-learning data for 39 courses

It has the following advantages: (1) It is independent of a specific LMS, since it is based on the Apache log files and not the LMS platform itself. Thus, it can be easily implemented for every LMS. (2) It uses new metrics in order to facilitate the evaluation of each course in the LMS and the instructors to make proper adjustments to their course educational material. (3) It uses two algorithms for analyzing LMS data, classifies the courses and suggests the proper actions to the educators.

Feedback about the method was received by the educators. The educators were informed about the indexing results along with abstract directions on how to improve their courses. Most of them increased the quality and the quantity of their educational material. They increased the quality by reorganizing the educational material with a uniform, hierarchical and structured way. They also improved the quantity by embedding additional educational material. By updating educational material, both quality and quantity were increased. A major outcome through the process of informing the educators about the results was that the ranking of the courses constitutes an important motivation for the educators to try to improve their educational material. Because of their mutual competition, they want their courses to be highly ranked. A few educators complained that their courses organization does not assist them to have high final scores in the ranking list. They argued that for example the metric interest is heavily influenced by the number of web pages used to organize the educational material. Thus, courses that have all their educational material organised in few pages have low interest score. They were asked again to re-organize the material for each course in the LMS according to the order they are taught, in order to facilitate the use by the students.

It should be mentioned that even if the scope of the method is on LMS platforms and educational content, it can be easily adopted in other web applications such as e-government, e-commerce, e-banking, blogs etc. Furthermore, enrichment, homogeneity and interest metrics may also be used for example by e-government applications, since enrichment shows how much information is handed over to the end user, homogeneity characterizes the percentage of information independently discovered by each user and interest indicates whether users are pleased with the material of the site and do not log out.

References

[1] Albeanu, G. (2007). E-Learning metrics. *Proceedings of the 2nd International Conference on Virtual Learning* (ICVL 2007).
[2] Avouris, N., Komis, V., Fiotakis, G., Margaritis, M., Voyiatzaki, G. (2005). Logging of fingertip actions is not enough for analysis of learning activities. *Proceedings of Workshop Usage Analysis in learning systems (AIED'05), Amsterdam.*
[3] Baker, R., Corbett, A., Koedinger, K. (2004). Detecting Student Misuse of Intelligent Tutoring Systems. *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems*, p. 531–540.
[4] Claroline, (2009). http://www.claroline.net.
[5] García, E., Romero, C., Ventura, S., de Castro, C. (2008). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. *Journal of User Modeling and User-Adapted Interaction*, 19 (1-2) 99-132.
[6] GUNet, (2009). http://eclass.gunet.gr/.
[7] Hamalainen, W., Vinni, M. (2006). Comparison of machine learning methods for intelligent tutoring systems, *In*: *Procceedings of Int. Conf. in Intelligent Tutoring Systems*, p. 525–534.
[8] Kazanidis, I., Valsamidis, S., Theodosiou, T., Kontogiannis, S. (2009). Proposed framework for data mining in e-learning: The case of Open e-Class, *Proceedings of Applied Computing 09,* Rome, Italy, p. 254-258.
[9] Lee, J., Hoch, R., Podlaseck, M., Schonberg, E., Gomory, S. (1999). Analysis and Visualization of Metrics for Online Merchandising. *Lecture Notes In Computer Science, Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, 1836, 126-141.
[10] Mazza, R., Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. I. *Journal of Human-Computer Studies*, 65 (2) 125–139.
[11] Mazza, R., Milani, C. (2004). GISMO: a Graphical Interactive Student Monitoring Tool for Course Management Systems. *Proceedings of International Conference on Technology Enhanced Learning '04 (T.E.L.'04)*. Milan, 18-19 November.
[12] Manford, C., McSporran, M. (2003). e-Learning quality: becoming a level five learning organization, in Mann S. and Williamson A. (eds): *Proceedings of the 16th NACCQ, Palmerston North New Zeeland* (p. 343-348).
[13] Mostow, J., Beck, J., Cen, H., Cuneo, A., Gouvea, E., Heiner, C. (2005). An Educational Data Mining Tool to Browse Tutor-Student Interactions: Time Will Tell*!*. *Proceedings of workshop on educational data mining* (p. 15–22).
[14] Romero, C., Ventura, S., García, E. (2008a). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1) 368-384.
[15] Romero, C., Gutierez, S., Freire, M., Ventura, S. (2008b). Mining and Visualizing Visited Trails in Web-Based Educational Systems. *In:* Educational Data Mining, *Proceedings of the 1st International Conference on Educational Data Mining*. Montreal, Quebec, Canada, 182-186.
[16] Romero, C., Ventura, S. (2007). Educational Data Mining: a Survey from 1995 to 2005. *Elsevier Journal of Expert Systems with Applications*, 33 (1) 135-146.
[17] Srinivasa, R. (2005). Data mining in e-commerce: A survey, In Sadhana, 30 (2 & 3) 275–289.
[18] Strazds, A., Kapenieks, A. (2007). Automated satisfaction measurement for e-learning target group identification, *IEEE Multidisciplinary Engineering Education Magazine*, 2 (3) September.
[19] Ueno, M. (2004). Data Mining and Text Mining Technologies for Collaborative Learning in an ILMS, *Samurai*, *In: Proceedings of the IEEE International Conference on Advanced Learning Technologies*, August 30-September 01, ICALT'04, p. 1052-1053.
[20] Ueno, M. (2002). Learning-Log Database and Data Mining system for e-Learning, *Proc. of International Conference on Advanced Learning Technologies, ICALT 2002*, p. 436-438.
[21] Valsamidis, S., Kazanidis, I., Kontogiannis, S., Karakos, A. (2010a, July). *Automated suggestions and course ranking through web mining, In:* Paper presented at the 10th IEEE International Conference on Advanced Learning Technologies ICALT, Sousse, Tunisia.

A Java System for Copy-Protected Display of Photographic Images on the Internet

Willis L. Boughton
Computer Information Systems Department
Business/Social Science Division
William Rainey Harper College
Palatine, IL 60067
847-925-6354. USA
wboughto@harpercollege.edu

ABSTRACT: *When high-resolution, copyrighted photographic images are displayed on the Internet, they must be protected from being copied by screen capture or by file. HTML and a Web browser alone do not provide protection, e .g., screen capture by a standalone program can not be prevented. This paper describes a Java system that provides copy protection for Internet display of photographic images organized into albums. Included is protection aga inst image file copy and against effective screen capture. The system involves three software components, two used by an administrator to build albums and a one used by client viewers. The system supports any number of albums, provides image annotation, zooming, and searching, requires no server-side software, maintains client privacy, and, using Java Web Start, does not require clients to install any software locally. The system is being used to maintain albums of noncommercial high-resolution historical and contemporary photos.*

Keywords: Java, Security, Image, Copyright

Received: 11 November 2011, Revised 26 January 2012, Accepted 30 January 2012

1. Introduction

Most photographic images displayed on the Internet require no copy protection, either because they are in the public domain or because they are low resolution and a copied image is of little value. High resolution, copyrighted images do require copy protection. An image potentially can be copied two ways: by screen capture and by coping the image file, e.g., by URL. Screen capture in some form is always possible since it requires only a screen capture program. Whether file copy is possible depends on the image display client/server software.

The fundamental issue with screen capture is that is can not be prevented by any client software technique. A standalone screen capture program can not be prevented from accessing video memory. To protect a high resolution image, the capture must be made ineffective using one or more of the following techniques:

a) Do not display the image in high resolution at all. Display it only in low resolution. This is probably the most common technique, but it prevents a viewer from evaluating the full quality of the image.

b) Display the image at high resolution but add disfigurement, e.g., text, geometrical pattern, or visible watermark, that practically can not be edited out. Too much disfigurement, though, may be unacceptable to viewers.

c) Display only part of the image at a time. This technique may not fully protect the image since multiple captured, partial images could potentially be combined to produce the full image.

The fundamental issue with file copy is how the client software accesses image files. There are two possibilities;

a) A custom client/server protocol, e.g., Java Remote Method Invocation (RMI) or a custom TCP/IP socket protocol, is used to transfer images from the server directly into client dynamic memory. This fully protects image files because they can be accessed only by the custom protocol and software, e.g., not by Web browser. This technique, though, requires custom software on the server as well as the client, which might not be possible. It might be that only standard Web services are available on the server.

b) URLs are used to access image files from a standard Web server. The URLs must be protected, i.e., not available to viewers, since if they were not, viewers could copy the image files directly from the server with a Web browser. HTML therefore can not be used to provide the URLs, since they would be directly in the HTML and viewable with a Web browser. Even if the "*save image*" feature of the Web browser were disabled, a viewer could get images from the browser's cache, or a viewer could use custom software, rather than a Web browser, to download image files.

Another issue is image zoom. It might be desirable to display high-resolution images in a much lower resolution screen area but give viewers the ability to zoom in, to evaluate image quality. Since screen capture can not be prevented, this zooming would enable a dedicated viewer to construct the high resolution photo from multiple zooms and screen captures, if the zooms were not disfigured. Since the full photo is already displayed in low screen resolution, it would not make sense to disfigure it, but it would make sense to disfigure the zooms.

2. Past Work and Current State

As early as 1998 [1], it was proposed that a Web browser plugin be used to deliver information securely over the Internet. Though the original focus was on documents and secure communication rather than protection of images, the browser plugin concept applies to images for both screen capture protection and image file protection. Various techniques for visual watermarking have been proposed [2, 3, 4]. These watermarks are designed to distort the image as little as possible while being difficult or at least tedious to remove. To minimize image distortion, the pixels in the watermarked areas must depend on the original image pixels. The watermark does not completely hide the original image in the watermarked areas. The point of view is that the viewer has access to only that one image, so the watermark must simultaneously protect the image while not distorting it too much. Whether the watermark can be removed successfully depends on the watermarking technique and the image itself, e.g., to what extent it has regions free of details [4]. Because the watermark pixels reflect the original image pixels, the watermark pixels might be used to recover, or at least attempt to recover, the original image.

The development [5, 6] of object removal algorithms and their incorporation in commercial photo editing programs has made effective visible watermarking more difficult. These algorithms can, depending on the image and region, remove even large objects in a single operation. The algorithms may make use only of pixels outside the region [5], so they might be effective against even complex image disfigurement. The algorithmically-constructed pixels in the region will not match those of the original (not disfigured) image, but this may not be apparent to someone not familiar with the original image. The algorithm work suggests that to make object removal apparent and protect the image, any disfigurement must, regardless of complexity, be spatially thick. This is demonstrated later in this paper.

3. Approach and System Requirements

The Java system described in this paper is an extension of the Web browser plugin concept combined with dynamic, opaque, zoom-dependent, size-selectable disfigurement of displayed images. This approach is distinctive as follows:

a) No custom plugin is required. The only plugin required is the Java plugin and Java Web Start, which are typically installed on computers. The viewer program, started by Java Web Start, provides the screen capture protection and file copy protection.

b) Image disfigurement is applied only when the image is maximized or zoomed. Images are displayed with no disfigurement at an initial resolution and disfigured only when the viewer maximizes the image or zooms in. A viewer is not immediately presented with a disfigured image, since this may repel some viewers. When an image is zoomed, the viewer can, by scrolling across the image, view all of it in detail even in the presence of the disfigurement.

c) The disfigurement completely hides the image in the disfigurement areas. The disfigurement does not depend on the image pixels. and the disfigured pixels can not be used to recover the original image, even if multiple images with the same disfigurement are available. The disfigurement also can be made as large as desired, to better protect against object removal filters.

d) All protection features are configurable, and images are viewable as albums with full search capability.

The system protects against screen capture and file copy and does not require custom server-side software, client software installation, or administrator programming, The specific system requirements were the following:

a) The only software required on the server must be a standard Web server.

b) The only installed software required on a client's desktop must be a Web browser and the Java runtime, including the Java browser plugin.

c) The client program must work with any Web browser and operating system, by using Java features only.

d) An administrator must be able to build and deploy photo albums to a server entirely by running programs, without writing any software, including scripts.

e) An administrator must have control, by option selection, of image screen capture protection, including initial image display resolution, whether images can be maximized, whether images can be zoomed, and the type of image disfigurement, if any. If disfigurement and zoom are both enabled, the disfigurement must be applied to zoomed images. If disfigurement and image maximize are both enabled, the disfigurement must be applied to maximized images.

f) A client user must not be able to copy a photo by file URL. Image URLs must be kept hidden and images must be saved only in dynamic memory.

g) An administrator must be able combine albums together into separate groups, e.g., put albums with historical photos into one group and albums with contemporary photos into another group. It also must be possible to put an album in multiple groups and to password protect a group.

h) An administrator must be able to specify the resolution of all image files in an album, and it must be possible to use different image resolutions for different albums.

i) A displayed photo must have annotation information, and a client user must be able to select photos by specifying annotation criteria. This search must span all albums in a viewing group.

j) A client user must be able to run a slide show of photos.

4. System Design

Figure 1 shows the system schematically. There are three system components:

a) Photo Manager program. An administrator runs this Java Swing GUI program to put photos into albums, define annotation for photos, and upload albums to a Web server by FTP.

b) Album Group Builder program. An administrator runs this non-GUI Java program to build an album group file and corresponding Java Network Launcher Protocol (JNLP) file. Each group file specifies albums viewable as a group; each JNLP file is the Java Web Start file for viewing that group. An administrator uploads these files to the Web server.

c) Viewer program. A client user runs this Swing GUI program to view an album group. The user specifies the album group JNLP file by URL in a Web browser, e.g., http://server.com/albums.jnlp. Java Web Start downloads the program specified in the JNLP file to the client's computer and starts program execution. From the Web server the program obtains the album group file specified in the JNLP file and the corresponding albums, and enables the user to select and view them.

4. Photo Manager Program

The Photo Manager program uses the Java Image Management Interface (JIMI) for image operations. All user functionality is available by GUI operations; no programming is required. The administrator puts photos into albums by specifying annotation such as location and topic; an album is a query to a custom database to retrieve photos that match specified annotation criteria. A photo can be in any number of albums. To build an album for Internet viewing, the user selects the parameters for the Internet

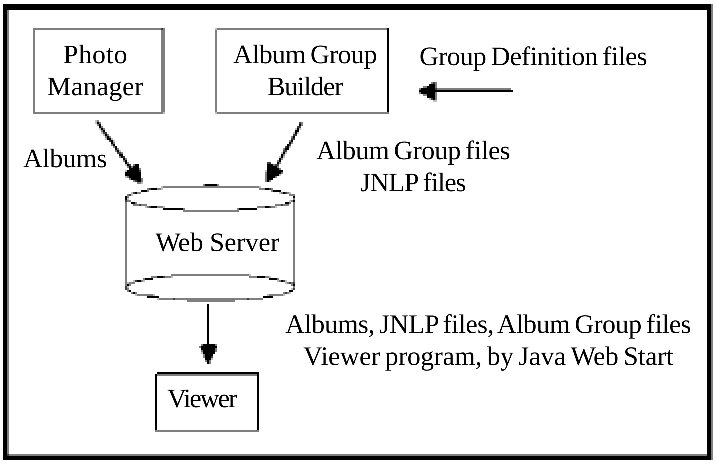


Figure 1. System Schematic

```
<?xml version= “1.0”?>
<!DOCTYPE albumGroup SYSTEM “albumgroup.dtd”>
<albumGroup>
       <name> usaphotos </name>
       <dbPath> c:\apps\bpm </dbPath>
       <codebase> http://www.bluemontsw.com/ </codebase>
       <vendor> Willis L. Boughton </vendor>
       <viewSize> 0.75 </viewSize>
       <maxViewSize> 700 </maxViewSize>
       <canMaximize> true </canMaximize>
       <password> xyz123 </password>
       <copyrightMsg> Copyright % Willis L. Boughton </copyrightMsg>
       <copyProtectMark>
               <labelMark>
                      <text> Copyrighted </text>
                      <stroke> 3 </stroke>
                      <relFontSize> 0.1 </relFontSize>
                      <count> 2 </count>
                      <xPos> 0.0 </xPos>
                      <yPos> 0.0 </yPos>
               </labelMark>
       </copyProtectMark>
       <album>
               <name> America </name>
               <useMark> true </useMark>
               <zoomable> true </zoomable>
       </album>
       <album>
               <name> FSA-OWI Website </name>
               <useMark> false </useMark>
               <zoomable> true </zoomable>
       </album>
</albumGroup>
```

Figure 2. Sample Album Group Definition File

album and clicks a build button. Among the parameters are image resolution, thumbnail resolution, thumbnail annotation, and image ordering. The program uses JIMI to build all images in a folder on the administrator’s computer. To upload an album, the administrator specifies the FTP parameters and clicks an upload button. For each album the program uploads the thumbnail image files for all photos, the image files for all photos, and the annotation information for all photos. The annotation information is in a custom binary file. For each album the program also uploads an empty index.html file into the specified Web server folder. This is done to prevent image file access by Web browser; see the URL security section of this paper.

6. Album Group Builder Program

Input to this program is an album group definition file specifying albums viewable as a group and information required for a matching JNLP file. The definition file is in XML format. Figure 2 shows a sample definition file, and Table 1 describes the fields. For each definition file the program writes an album group binary file and matching JNLP file. These files are uploaded to the server by FTP. Figure 3 shows the JNLP file that matches the Figure 2 definition file. The resources fields in the JNLP file specify the Java version, the memory required by the client viewer program, and the name of the client program file, bpmviewer.jar. The JNLP application-desc fields are the two input parameters to the client program. They are the root URL on the Web server for images and the name of the album group file. The full URL for the album group file is the root URL plus the album group file name. This URL is derivable from the JNLP file, which has consequences discussed in the URL security section of this paper.

```
<?xml version=’1.0' encoding=’UTF-8’ ?>
<jnlp spec=’1.0’ codebase=’http://www.bluemontsw.com/’
href=’usaphotos.jnlp’>
<information>
   <title> Photo Album Viewer </title>
   <vendor> Willis L. Boughton </vendor>
</information>
<resources>
   <j2se version=’1.5+’ initial-heap-size=’256M’ max-heap-size=’384M’/>
   <jar href=’bpmviewer.jar’ main=’true’/>
</resources>
<application-desc main-class=’viewer.Viewer’>
   <argument> http://www.bluemontsw.com/ </argument>
   <argument> usaphotos.bpm </argument>
</application-desc>
</jnlp>
```

Figure 3. Sample JNLP File

7. Viewer Program

The Viewer is a standalone program executed by Java Web Start. The user starts the program by entering a URL in a Web browser, but the program executes in its own window, separate from the Web browser. Figure 4 is a screen capture of the Viewer displaying a high-resolution photo with no disfigurement. The program executes with Java applet security; it has no access to the local filesystem or clipboard and can only read files from the Web server that supplied the viewer program. There are no privacy issues for the user. The program obtains all images from the Web server by HTTP. Image files are read directly into dynamic memory, and the user has no access to image files. When the user selects an album, the program obtains the thumbnails for all photos from the server and displays them in user-selectable tabbed pages. To display a photo the user double-clicks the thumbnail, which opens the full image in a dialog window. The dialog window has buttons for moving to the next or previous photo and starting a slide show at the photo. Depending on how the administrator defined the album group, the dialog also may have buttons for maximizing the image display size and resizing it, and the user may be able to zoom and scroll an image. The user always can select how the photos are ordered, can select a set of photos by tag matching from the currently-viewed album, and can do a search with “*” wildcards across all albums in the group. Figure 5 shows an example of a search, in which the user is selecting only photos taken in Illinois in the year range 1940 through 1960. The Viewer uses multiple threads to maximize responsiveness. The user can display a full image while thumbnails of other images are being loaded, and double-buffering is used to retrieve each full image. When a photo is displayed, the program starts retrieving the next one, so it may be immediately displayed when the user clicks next. Annotation values which are common to all photos in an album and which more appropriately belong in the album's name are excluded from the displayed annotation for all photos in that album, to avoid repetition. For example, if all photos in an album were taken by the same person, that person's name does not appear in any image annotation for that album.

Field	Count	Description
name	1	The name for the Album Group file and JNLP file
dbPath	1	The folder containing the Photo Manager database
codebase	1	The JNLP codebase, which specifies the server folder containing the Viewer program
Vendor	1	The name displayed when Java Web Start starts the viewer.
viewSize	0 or 1	The initial maximum dimension of the image display window, as a fraction of the screen size, defaulted to 0.5. Images displayed larger than this (maximized) or zoomed in this window are disfigured if disfigurement is specified.
maxViewSize	0 or 1	The maximum dimension of the image display window, in pixels, defaulted to 700. If this value results in a size less than viewSize, it overrides viewSize.
canMaximize	0 or 1	If true, the viewer can maximize the image display window. If false, the maximum window size is set by viewSize and maxViewSize.
password	0 or 1	A password enabling a viewer to override the canMaximize setting..
copyrightMsg	0 or 1	A copyright message to be displayed at the bottom right corner of displayed images, defaulted to none
copyProtectMark	0 or 1	A disfigurement to be displayed , in gray color, on each image. The options are a labelMark, the default, or a geometricalMark.
labelMark	0 or 1	A text mark with a specified text, line stroke, repeat count, font size relative to the image display window, and relative position in the image display window. If either position is zero, the text is centered in the image display window. If the repeat count is 1, the text is drawn once in gray. If it is 2, the text is drawn twice, once in gray and once in black, slightly offset.
geometriclMark	0 or 1	A specified number of ellipses drawn on images, either outward from the center or inward from the edge, with a specified stroke.
album	1 or more	An album, consisting of name in the Photo Manager database, whether a copyProtectMark should be used, and whether the album images are zoomable.

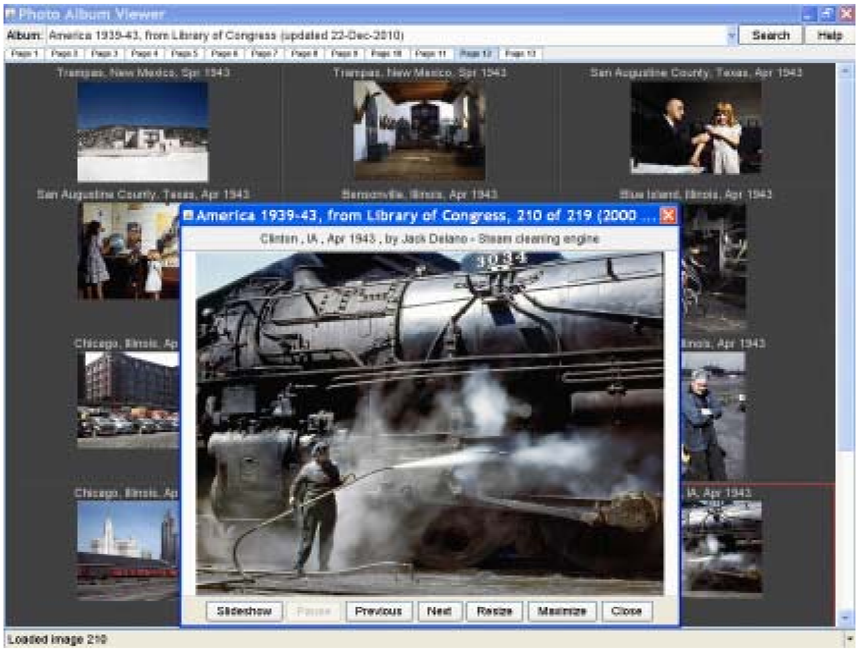
Table 1. Album Group Definition Fields

8. Image Disfigurement

Figure 6 shows the three disfigurement options: text, outer elliptical marks, and inner elliptical marks. The font size, stroke size, position, and repeat count can be specified for the text, and the stroke size and number of lines can be specified for the elliptical marks. The magenta text in the bottom right corner is the optional copyright label. Figure 7 shows the effectiveness of a large and thick text disfigurement against the object removal filter of a commercial photo editing program. On the left is the original, disfigured image, and on the right is the image with the disfigurement removed with the filter. The removal was done in one operation by selecting the entire disfigurement. The filter effectively removes the disfigurement in the region with no features (sky) but severely distorts the regions with features (fence and building). Similar distortion occurs when only the disfigurement on the building is removed.

9. Image URL Security

A Viewer user can not obtain image file URLs from the program itself since they are not displayed and there is no HTML. This



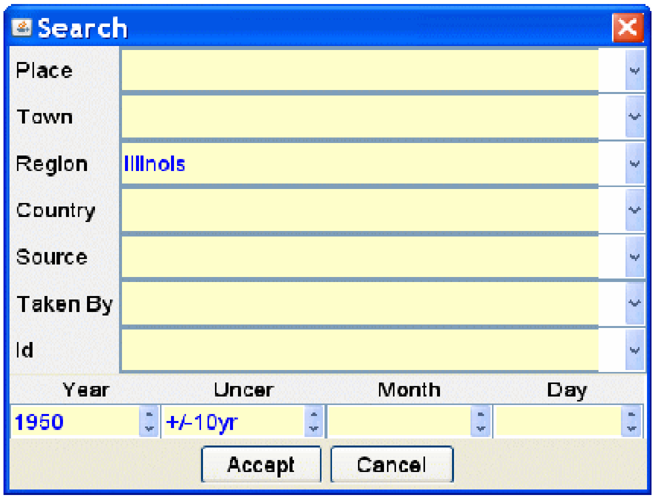
Album: America 1939-43, from Library of Congress (updated 22-Dec-2010)

America 1939-43, from Library of Congress, 210 of 219 (2000 ...

Clinton , IA , Apr 1943 , by Jack Delano - Steam cleaning engine

Loaded image 210

Figure 4. Viewer Program Screen Capture



Place	
Town	
Region	Illinois
Country	
Source	
Taken By	
Id	

Year	Uncer	Month	Day
1950	+/-10yr		

Figure 5. Sample Multi-Album Search

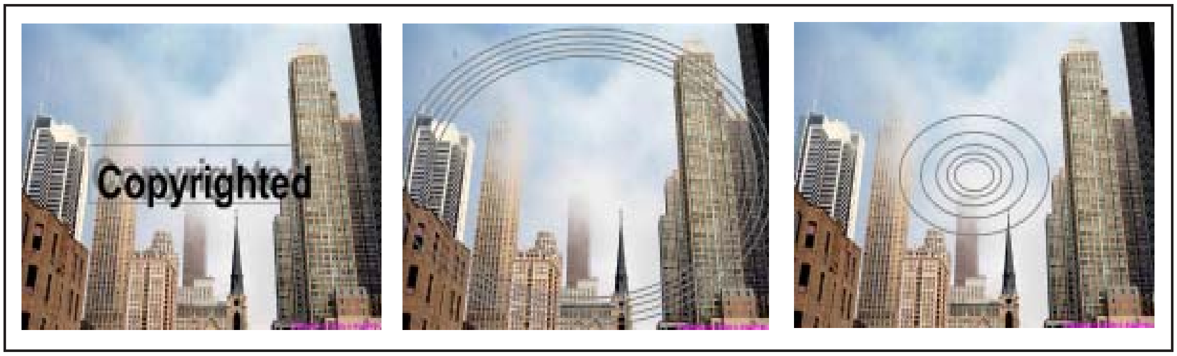


Figure 6. Disfigurement Options

does not, however, fully protect the URLs. The complicating issues are:

a) If the URL for a Web server folder is entered in a browser, it will display a file listing for that folder if it does not contain an index.html file. If a Viewer user could obtain the URL for a Web server folder that contains images but does not contain an index.html file, he could get the image files by a save directly from the browser's file listing. The Viewer user needs to know only the folder URL, not the image URLs.

b) The URL of an album group file is known to all clients from the JNLP file, as described previously. A Viewer user can download this file by entering the URL in a browser and doing a save. All contents of the album group file that identify image folders therefore must be unreadable. If they were not, a user could combine these folder names with the root URL on the server to obtain the image folder URLs.

c) A Viewer user could obtain the image URLs by using a proxy server that displays the HTTP requests.

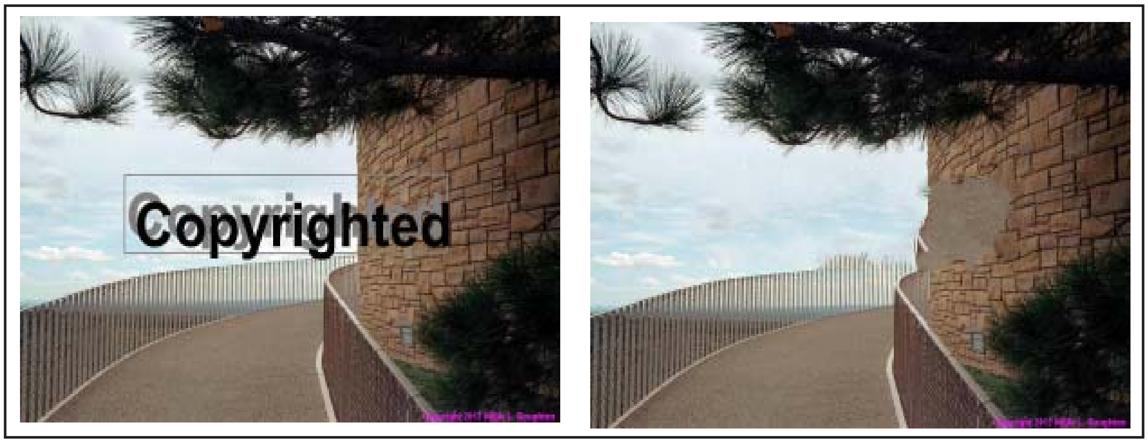


Figure 7. Disfigurement Removal with Photo Editing Filter

The system addresses these issues as follows:

a) The administrator uses long, password-like names for the image folders on the server, to minimize the chance of a Viewer user obtaining a folder URL by guess alone. In addition, when the Photo Manager uploads an album it uploads an empty index.html file into the folder containing the images. Even if a Viewer user were able to obtain the image folder URL, the browser would display only an empty Web page, not the folder listing and image file names. The Photo Manager also uploads an empty index.html file into the server root folder, if the folder does not already contain an index.html file. This prevents a Viewer user from entering the root URL in the browser and seeing folder names from the browser file listing. This is a secondary protection, since even if a user obtains the folder names, the index.html files in the image folders prevent the browser from displaying image file names.

b) Server folder names in the album list file are encrypted.

c) The HTTP connections for reading images are opened with the Java "*no proxy*" parameter, which forces direct connections even if a proxy is being used.

10. Use

The system is being used to maintain a database of several thousand noncommercial photos, with a few hundred available on the Internet. They are arranged in multiple groups, each generally consisting of multiple albums. The image files have resolution of 2000 or more pixels in the larger dimension, higher than typically used on the Web. Albums containing images in the public domain are viewable full screen, at full resolution, zoomable, without disfigurement. Copyrighted images are viewable in a combination of limited resolution, no zoom, and disfigurement, depending on the album group. A group of two example albums, one of public domain photos and another of copyrighted photos, is at http://www.bluemontsw.com/usaphotos.jnlp.

References

[1] Jenkin, M., Dymond, P. (1998). A Plugin-based Privacy Scheme for World-wide Web File Distribution, *In:* Proceedings of the Thirty-First Hawaii International Conference on System Sciences. 7, 621-627.

[2] Braudaway, G, Magerlein, K, Mintzer, F. (1996). Protecting Publicly Available Images with a Visible Image Watermark, *In:* Proc. SPIE 2659, 126.
[3] Kankanhalli, M. S., Rajmohan, Ramakrishnan, K. R. (1999). Adaptive Visible Watermarking of Images, IEEE International Conference on Multimedia Computing and Systems, 1,568-573.
[4] Mohanty, S. P., Ramakrishnan, K. R., Kankanhalli, M. S. (2000). A DCT Domain Visible Watermarking Technique for Images, IEEE International Conference and Expp on Multimedia, 2, 1029-1032.
[5] Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C. (2000). Image Inpainting, Proceedings of ACM SIGGRAPH, p. 417-424
[6] Avidan, S., Shamir, A. (2007). Seam Carving for Content-aware Image Resizing, *ACM Transactions on Graphics*, 26(3).

Author Biography

Willis L. Boughton has a Ph.D. in astronomy from the University of Illinois at Urbana-Champaign and twenty years of business experience in medical imaging, real-time embedded systems, and statistical process control. His last business position was Director of Process Measurements for Ameritech Corporate Information Services. Since 2000 he has been on the faculty of the Computer Information Systems Department at William Rainey Harper College in Palatine, IL. He is the author of a two-volume Java programming textbook. His last publication was in the Journal of Instruction Delivery Systems.

REVIEW:
Semantic Web; Ontology Specific Languages for Web Application Development

Zeeshan Ahmed, Thomas Dandekar, Saman Majeed
Department of Bioinformatics
Biocenter, University of Wuerzburg
Am Hubland 97074, Wuerzburg, Germany

ABSTRACT: *This paper is based on the findings of a literature review on the field of World Wide Web. A list of some key publications and explicitly provided literature is briefly included in this paper; addressing the importance of the field of Semantic Web and describing its contributions as the mechanism for structuring the information over the web in a format so that machines can understand the semantic context. Highlighting the technological innovation of Semantic Web, this paper presents Ontology some domain specific languages for Ontology construction: eXtensible Mark-up Language, Resource Description Framework and Web Ontology Language; offering different ways of explicitly structuring and richly annotating Web pages. Furthermore this paper discusses how Ontology is contributing to the semantic based web system development with some example of real time applications and concludes with some existing limitations needing to be overcome.*

Keywords: Ontology, Semantic Web, World Wide Web

Received: 11 October 2011, Revised 3 December 2011, Accepted 8 December 2011

1. Introduction

World Wide Web is a global information sharing and communication system made up of three standards: Uniform Resource Identifier (URL), Hypertext Transfer Protocol (HTTP) and Hypertext Mark-up Language (HTML) developed by Tim Berners-Lee to effectively store, communicate and share different forms of information [1]. The information is provided over the web in text, image, audio and video formats using HTML, considered unconventional in defining and formalizing the meaning of the context. Currently most of the data over the web (or attached to web applications) is not well structured; it is easy to go for scattered extensive information by looking into bookmarked web pages but quite difficult to extract a piece of needed (particular) information. HTML documents are formatted such that these cannot be processed semantically because these are only available in unstructured readable format. This deficiency leads to the problems of *intelligently searching, extracting, maintaining, uncovering and viewing the knowledge based information over the web*. Moreover deficiency becomes the major cause of some semantic oriented problems e.g. Meta data extraction. There is a current need to have an approach which can publish data over the web not only readable but also in machine understandable and processable formats.

Most of the search engines are promising enough as they require excessive manual pre-processing e.g. designing a schema, cleaning raw data, manually classifying documents into taxonomy and manual post processing e.g. browsing through large result lists with too many irrelevant items [11]. Furthermore some search engines and screen scrapers are also there but are insufficient in creating a rich multi domain information environment [2]. Such search engines use full text query to search information but can only return unstructured contents not the actual structured information stored in the attached database whereas screen scraper extracts and reassembles fragments from the web pages.

To increase data integration and interoperability over the web the concept of "*Web Service*" was introduced [28, 51, 52]. Initially due to their dynamic nature the web services became very popular in industry in very little time but later on an enormous increase in the number of web services with end-to-end service authentication, authorization, data integrity and confidentiality problems were identified which are still alive and not handled by existing web technologies [53].

If the data will be structured over the web only then it will improve the process of search, extraction and maintenance of particular information over the web. As an advanced version of Web, Web 2 was introduced [1] to improve interactive information sharing, interoperability and user centred design web application development. Later on Web 2 was also updated to a new concept Web 3. The concept of Web 3 is to transform Web (online data) into a database to provide accessibility of the contents by multiple non browser applications [19]. Continuing the streak of advancement in existing web and to cope with the currently existing web problems: *Information filtration, security, confidentiality and augmentation of meaningful contents in mark-up presentation*, the concept of "*Semantic Web*" (SW) was proposed by Tim Berners Lee [3] (renowned as the modified version of Web3).

SW is a mechanism for presenting information over the web in such a form so that humans as well as machines can understand the semantic of the context. It is a linked mesh of information which could be processed [5]. The aim of SW is to produce technologies and domain specific languages capable of reasoning on semi structured information [4]. SW is an intelligent conception and advancement in World Wide Web to collect, manipulate and annotate information independently by providing effective access to the information. It provides categorization and uniform access to the resources and advances the transformation of World Wide Web into semantically modelled knowledge representation systems with a common framework which allows data to be shared and reused [7]. It also gives the concept of semantic based web services for dynamic composition of service based applications. SW research depends on a number of key methodologies: *Knowledge Representation Languages* and *Reasoning Algorithms* [32]. Currently SW is standing on a very important building block of Ontology [6], aims of structuring data into processable semantic models [8], as the collection of interrelated semantic oriented concepts (see section 2).

One of the most recent developments is the integration of SW concepts in agent and multi agent based (communication) system development. Some agent based SW systems have been developed using SW technologies. A new SW specific language i.e. *Meta-language of the agent* (AgentML), is also introduced to formulate the agent by discussing its (agent) components [54]. Agents together with SW can provide many practical online benefits e.g. web indexing agents can turn documents into formal knowledge, personal agents can be used for reservations (e.g. holiday's trips. doctor appointments etc.), multi agent system can be used to build and maintain additional Linked data sets etc [55].

The ultimate goal of semantic web is to structure the meaningful contents of unstructured published data over web to take advantage in improving the data extraction processes [3] and to involve knowledge management in creating an advanced knowledge modeled management systems. Without a doubt SW has contributed in the progress of web but still there are some limitations and due to them SW is not currently successful in attaining the actual goal of completely structuring the information over the web making advanced knowledge modelled system. The need is to enhance the existing semantic web technologies and proposition of new domain specific languages for better SW application development because all the theories can be fruitful if the implementation is possible.

The remainder of this review paper is organized as follows: targeting the challenges of implementing a SW application capable of providing semantic based search to extract desired information from attached repositories over the web; ontology is explored and discussed in section 2. Section 3 presents some ontology (domain) specific languages and section 4 describes some example of SW applications. Section 5 provides some limitations and section 6 conclusions.

2. Ontology

Ontology is the explicit representation and description of already available finite sets of terms and concepts used to make the abstract model of a particular domain. Ontology has become a favorite subject for different research communities e.g. computer science, philosophy, bioinformatics, knowledge engineering & management, natural language processing, information retrieval, cooperative information systems and information incorporation etc., because of its interdisciplinary nature [9]. With variability in its usage, ontology has different definitions with respect to the different fields e.g. in computer science it is defined as the combination of concepts and relationships for domain modelling [37], in philosophy it is known as the study of "what there is" [38], or a mathematical formulation of properties and relationships of certain entities [39], and so on.

Ontology is known as the major building block of SW to structure data in machine processable semantic based models for knowledge base systems implementation. Ontology has two kinds of representation schemes i.e. *informal* and *formal* [40]. Informal representation method provides broad range of entities and relations (object, attribute, value triples) whereas formal representation method is based on family of description logics [41]. Ontology produces the abstract modeled representation of already defined finite sets of terms and concepts involved in intelligent information integration and knowledge management [9]. It is basically categorized in three different categories: *Natural Language Ontology* (NLO), *Domain Ontology* (DO) and *Ontology Instance* (OI). NLP provides the relationships between generated lexical tokens of statements based on natural language [25]. DO models precise domains and OI is to generate automatic object based web pages [9].

Ontologies are constructed and connected to each other in a decentralized manner to clearly express semantic contents and arrange semantic boundaries to find out required needed information [10]. It is mainly the combination of following elements: *classes, properties, values, relations between classes, restrictions on properties* and *characteristics of slots* e.g. during the development of natural language search engines (in most of the cases) natural language based information (e.g. queries, grammar rules, vocabulary etc.) are treated as the input to the ontology construction process, which first parses the text in nouns and verbs. Nouns are represented as "*classes*" and verbs as "*properties*" containing values, relationships with other properties and some constraints. Classes are further divided in main and sub class categories maintained in taxonomical hierarchy. The size of ontology varies due to the increase in number of classes and instances.

Ontologies can be made manually from scratch e.g. by extracting information from web and by merging already existing ontologies into new ontologies. But this manual process sometimes becomes very complex and time consuming especially when dealing with a large amount of data. Moreover, to support the process of semantic enrichment reengineering for the building of the web consisting of Meta data depends on the proliferation of ontologies and relational Meta data. This requires production of Meta data at high speed and low cost. So in these cases machine learning approaches can be very helpful in generating ontologies automatically because they provide real time schemes like classification rules, instance based learning, numeric predictions, clustering, Bayesian networks and decision trees for the generation of ontologies.

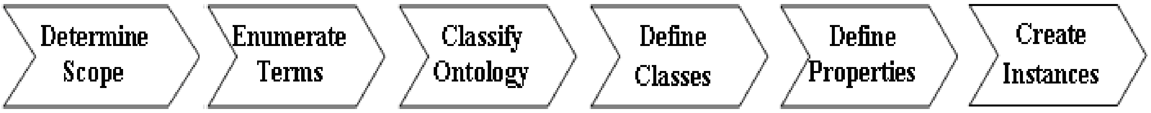


Figure 1. Ontology development activities [12]

Figure Legend. Six Ontology development activities i.e. Determine Scope, Enumerate Terms, Classify Ontology, Define Classes, Define Properties and Create Instances.

Ontology development is an iterative process based on six main activities: *Determine Scope, Enumerate Terms, Classify Ontology, Define Classes, Define Properties* and *Create Instances* as shown in Figure 1. In the beginning of an ontology development process it is very important to determine the scope otherwise it will be both time and effort consuming [12]. Then enumerated terms are needed to be identified to classify ontologies within their respective types. Classes and their respective properties along with their relationships and constraints are defined using identified enumerated terms. In the end only the instances are created and used. To implement ontology development process some experience, a powerful user friendly ontology supporting tool and communication between domain experts and developers is required e.g. LibraryWine [12].

3. Ontology Specific Languages

First step in building ontologies is to create the nodes and edges. Once the concepts (nodes) and relationships (edges) of graph based ontology are constructed then the next step is to quantify the strengths of semantic relationships [11]. Ontologies can be constructed manually and automatically by using some ontology supporting languages: *eXtensible Mark-up Language* (*XML*), *Resource Description Framework* (*RDF*) [5] and *Web Ontology Language* (*OWL*); all offer ways of more explicitly structuring and richly annotating Web pages.

• XML is one of the fundamental contributions towards middleware technologies [13]. It is a markup Meta language which allows sharing of information between different applications through markup, structure and transformation. As the major contribution towards semantic web, XML uses Data Type Definitions (DTD) and depends on data types, attributes, both internal and external elements structure documents and provide syntax serialization and abbreviation for data modelling [17]. The XML schema restricts the syntax to be only used for the structured documents, because of this XML has two main problems in process of information extraction; first it is without semantics and second is the arbitrary naming and structuring of elements [20].

• RDF, a URL based syntax data representation provides a secure and reliable mechanism for the exchange of metadata between web applications. RDF processes Meta data by making an abstract data model based on three object type attributes: *Resource, Property and Statement* [24]. Resource is an expression, property is an attribute to describe a resource and statement is a resource having some property and value. RDF uses three containers: *object bag, sequence, and alternative*, to keep multiple available and alternative values arranged in an order in resources and properties. The "object bag" contains resources, "sequence" contain resources along with their properties having single or multiple values arranged in order and "alternative" contains resources having alternate value(s) of a property [18]. RDF provides syntax serialization and abbreviation for RDF data modeling. Serialized syntax expresses the full capabilities of data modeling in a very regular fashion and abbreviated syntax includes additional constructs to provide a more compact form in representing a subset of the data model. RDF is more useful than XML in ontology construction because it provides semantic based features for data including domain independency, vocabulary and privileges in defining terminologies used in schema language. Furthermore it also provides syntax based on reification (statements about statements), data types, attributes, nesting, elements, element types, element container and no restrictions in structuring document like XML. RDF has its own grammar but it is not complete, it relies on the support of XML to fulfill its need. Moreover, the RDF modeling mechanism is insufficient in expressing various logical statements [17].

• The Web Ontology Language OWL was proposed by the W3C proposal in 2004 [35]. OWL is derived from American DARPA Agent Markup Language (DAML) [22]. OWL is based on ontology, inference and European Ontology Interchange Language (OIL) [23], and is intended to be an extension in RDF in expressing logical statements [21]. It provides an Application Programming Interface (API) for the development of Semantic Web application using ontology [31], influenced by XML Document Object Model (DOM) [36], can easily be used in any OWL supporting language editor e.g. Protégé 4. OWL API provides number of classes and interface for OWL based ontology modeling [34]. It is rich in vocabulary because it not only describes classes and properties but also provides the concept of namespace, import, cardinality relationship between the classes and enumerated classes. OWL has some specific limitations like only one "*Namespace*" per project is allowed, "*Import*" is not currently supported, no database backend and Multi-User support and a few OWL Language features are also missing [12].

Using above mentioned ontology specific languages, an extensible and customizable toolset is available i.e. *Protégé*, for the construction of ontologies. Protégé is with some excellent features towards automatic generation user-defined and modeled graphical interfaces for the acquisition of domain instances, extensible knowledge modelling and embedding standalone applications [12]. Furthermore Protégé provides plug ins acceptability to enhance its functionality e.g. Some new graphical user interface related features, new visualization libraries, import and export formats etc.

4. Semantic Web Applications

Residing in the domain of Semantic Web many products have been developed and and several approaches have been introduced by many researchers providing values in the implementation of semantic based applications with use of ontology [26]. Newly proposed approaches are helping in structuring data over the web to take advantage in implementing efficient web based information retrieval search mechanism. In this section we are discussing some of the recent SW based approaches e.g. Semantic Desktop [44], Reisewissen [30], Intelligent Semantic Oriented Agent based Search (I-SOAS) [27, 29] and Meta Data Search Layer[46].

Semantic Desktop; stepping into user's mid frame by implementing Personal Information Model (PIM). PIM is designed to improve the process for the identification of documents and retrieval of required document. The design is based on ontologies and classes. The relationships of classes and ontologies are predefined and the information can be accessed using RDF graphs. Four rules based on forward changing principle are defined to retrieve the information. This information is divided into three parts: *author* (single or team), *relevant project*, and *relevant solution*. The system works in the following way: first query runs aiming to find out the project and if project is found then it moves to find out the related documents of the project. The proposed architecture mainly consists of three main components: *receiver, interpreter and analyzer*. Receiver is used to provide index services and obtain the information about the structure of indexed files with the help of so called brainFiller. Interpreter first retrieves information (structure / unstructured) using full text search and then uses so called LiveLink. The contents of obtained information is structured with the help of manual annotation and meta data (based on their properties and preferences). As the last step, the analyzer queries (using Jena inference engine) the created RDF models to infer runs and uses F-Logic to integrate rules. Authors designed four case scenarios for proposed approach to share searched information: *Local Search*, *Group Search, Closed Community and Open Community*. Local search scenario only deals with the search mechanism and can only be applied to a personal desktop. Group search can be applied with *n* particular network domains. Closed community consists of a number of users having different roles but same topic where as open search consists of users with different roles and different topics.

Reisewissen is a hotel recommendation engine and travel information system. It provides quality services by semantically connecting, organizing and sharing the isolated pieces of information by transpiercing to data sources, caching & fetching of data, transforming data from heterogeneous to RDF models, mapping of ontologies between database and triples, matching RDF and non RDF based information [30]. Reisewissen is implemented by manually mapping ontologies (RDF models). Reisewissen identifies potential relevant knowledge sources and provides quality service by semantically connecting, organizing and sharing the currently isolated pieces of information in an online portal to anticipating customer behavior. The design of Reisewissen (Figure 2) is composed of three main components: *Data Connectors (DC)*, *Evaluation Framework (EF)* and *Evaluation Engine (EE)*. Data connectors are used to provide transparency to data sources and transformation of data from heterogeneous to common data format (RDF and Java objects), moreover it also provides the caching and fetching of data. Evaluation Framework is a workbench to test the quality of data and rules by providing functions and filters to map resources and return result in decisive format (Boolean or float value). Evaluation Engine combines individual filters to rank and filter information by weighting and yielding the overall score. Information is obtained using Simple Object Access Protocol (SOAP) based web services and stored in both RDF and non RDF formats, which are then matched to find out the desired result. Data stored in RDF format is based on developed ontologies mapped between database and RDF triples. Moreover Reisewissen uses Prolog to capture ex-pert's knowledge which can be formalized and can generate new data by implementing the customer request in evaluator encapsulated rules. Data is matched semantically by combining data properties to ontology and similarities between two concepts are determined by distance reflecting their respective positions in hierarchy. As output a list of selected results are generated to customer.

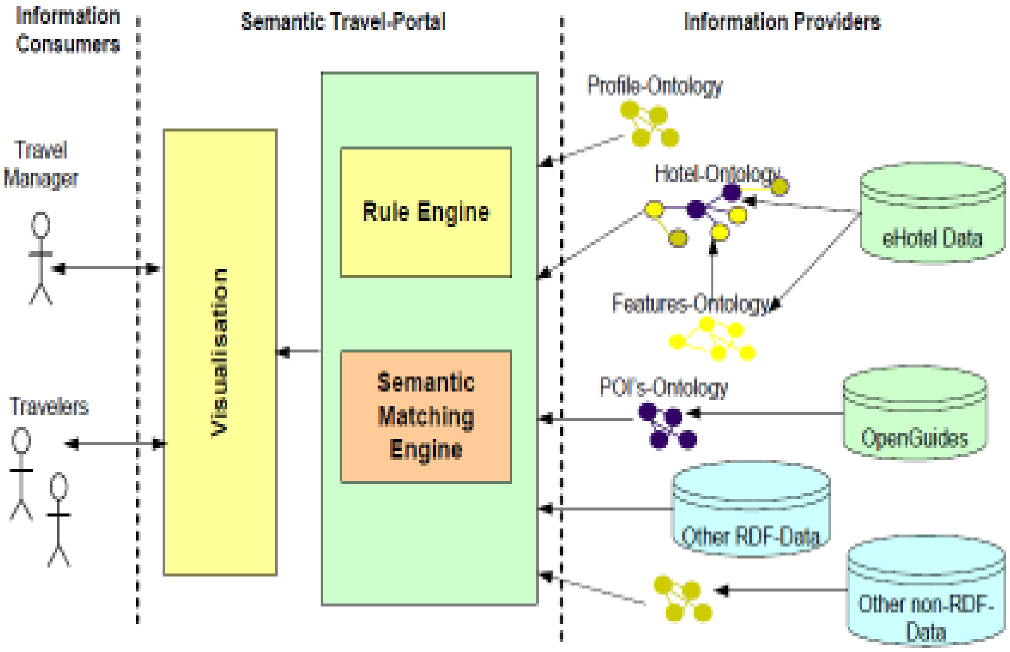


Figure 2. Reisewissen hotel recommendation engine [30]

Figure Legend. Reisewissen hotel recommendation engine consists of three major parts: The data connectors handling the transparent access to data sources and transformations, the evaluation framework providing functions (evaluators) to evaluate and filter resources and the actual evaluation engine combining the individual evaluators to rank and filter a set of hotels as well as yield evaluation results to a customer user interface.

I-SOAS is proposed to solve the problems of implementing semantic oriented information models, Meta data extraction and multiuser access in Product Data Management (PDM) Systems [48]. It provides a flexible multi user graphical interface [49], intelligent search [50], and knowledge management [45]. Without going into the details of all modules, we discuss only the most relevant one i.e. *intelligent search*. The overall job of intelligent search module is divided into five main iterative sequential steps i.e. *Data reading, Tokenization, Parsing, Semantic Modelling* and *Semantic based query generation* [47]. The main concept behind the organization of these five steps is to first understand the semantic hidden in the context of natural language based set of instructions and generate a semantic information processable models for the system's own understanding and information processing. At first the Data Reader reads and organizes input data from GUI into initial prioritized instructions list. Then the Data Tokenizer tokenizes instruction one by one, which are then treated by the Data Parser for parsing and semantic evaluation with respect to the grammar of used natural language. Then the Semantic Modeler filters the irrelevant semantic less data and generate Meta data based semantic model. Then in the last step the Semantic Based Query Generator is supposed to generate a new query used for further data storage and extraction of desired result. Following the concepts of semantic web and ontology construction, authors have created the ontology (Figure 3) using RDF with respect to the structure of proposed natural language (English) grammar of I-SOAS [42, 43].

Meta Data Layer; a process for Meta data search based on three questions for information extraction: *what user needs, where it lies,* and *how it can be retrieved*. The targeted objective is to identify the location from set of locations contained by a document and avoid looking into non-specific document. Scalability and efficiency of this approach is determined using simulation of documented Meta data keywords, location pointers, node connections and node knowledge. The whole process of identifying target location and search consists of nine procedural steps (Figure 4).

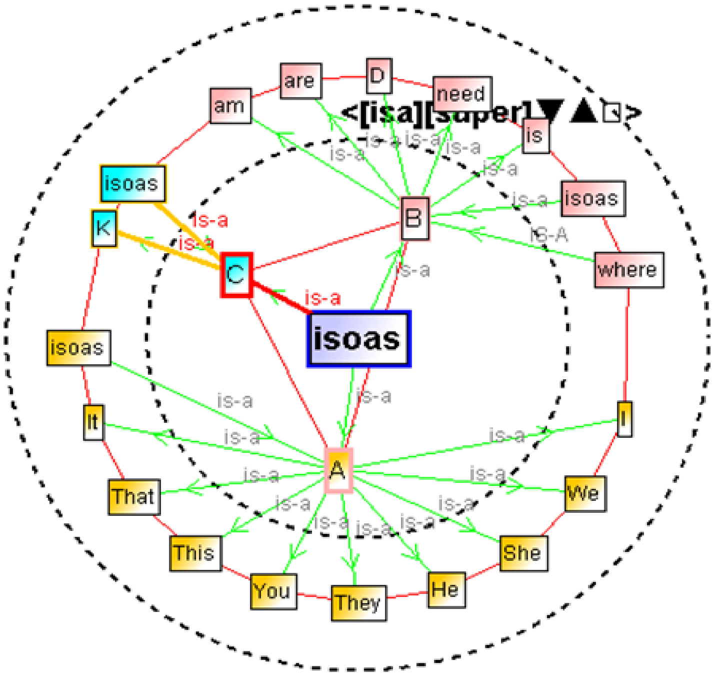


Figure 3. I-SOAS Ontology; Class relationships [43]

Figure Legend. I-SOAS Ontology; Class relationships consists of one main Class ISOAS, then three sub classes*: A*, *B* and *C*. All three subclasses contain their further subclasses and the relationships of this subclass with each other.

- Select target document from network having at least one keyword.
- Use keywords contained in document for the construction of a search query.
- Start with free node (not already containing any target document).
- Make a record of start node.

- Start node’s knowledge treated as basic knowledge for the selection of other sub nodes.
- If number of nodes is equal to forwarding degree select those nodes.
- If number of nodes is less than forwarding degree select additional nodes.
- If number of nodes is more than equal to for-warding degree select subset of nodes.
- For each selected node, if node contains target document then update connectivity and if it doesn’t then continue search using nodes.

Authors have explicitly mentioned that this search mechanism is good but there is still room for improvement in examining the path length of searches for different and same users characterized by their different query distributions. Moreover regarding the time to converge to a stable network, this can meet ambiguities and needs to have more realistic simulation using parameters and distributions.

Other than the mentioned ones, there are also some more beneficial ontology based approaches such as Google translate [54] join different language processing engines to predict, Bioinformatics related approaches e.g. protein interaction database such as i-Hop [55] and XplorMed [56].

5. Limitations of Semantic Web; Ontology

The development of ontology driven applications is difficult because of some disadvantages, limitations and principal problems which are as follows:

- Natural language parsers (used to parse the information to construct the ontology) are limited because they can only work over a single statement at a time [13].

- Not possible to define the boundaries of ontology based particular domain’s abstract model.

- Not possible to automatically handle the increase in size of ontology (due to the increase in number of classes and instances).

- Creating ontologies manually is a time consuming process which becomes very complex when there is a large amount of data to create large number of ontologies. To take advantage in creating large number of ontologies by reducing the complexity and time, an automatic ontology creation mechanism is required. Some mechanisms are already proposed and implemented to create ontologies automatically but they are in-sufficient and less qualitative. While creating nouns based classes using existing automatic ontology creation mechanism is automatically possible now, it is quite impossible to identify the possible existing relationships between classes to draw the taxonomical hierarchy [14]. Furthermore it is also quite impossible to perform automatic emergence of ontologies to create new ontologies [16].

- Currently available ontology validators are restricted and not capable of validating all kind of ontologies e.g. based on complex inheritance relationship [53].

- Domain specific ontologies are highly dependent on the domain of the application and because of this dependency domain specific ontologies contain specific senses which are not possible to find in general purpose ontology [15].

- The process of semantic enrichment reengineering for web development consists of relational meta data required to be developed at high speed and in low cost depending on proliferation of ontologies, which is currently also not possible.

- Handling the dynamically raised calculations caused by the comparison of big complexities of similar ontologies is also not possible [16].

- Only one namespace per project is allowed during the ontology creation using OWL [12].

- Import is not currently supported during the ontology creation (using OWL) [12].

- No database backend support is available during the ontology creation (using OWL) [12].

- No multi user support is provided by any ontology supporting language [12].

Regardless of above mentioned limitations, using ontology is beneficial in and tantamount structuring data and implementing data extraction process for efficient information search.

We mention here furthermore, that as currently there is no fully automated ontology creation for the web possible, the method of highly user based structured web pages is also powerful, notably the Wiki movement (www.wikipedia.org) for instance to structure web pages of different bacterial genomes (e.g. Subti-Wiki, Staph-Wiki). Such approaches are complementary to automatic semantic web efforts and both profit from each other.

6. Conclusions

Introduction of World Wide Web brought new meaning to information sharing and communication. However advancement in Data formats made data accessibility much more complex. With the concept of Web Service, a new and innovative filed of Semantic Web came into being. A review research has been conducted in the field of Semantic Web and its importance has been elaborated in detail in this paper. In this paper, we have presented a major building block of Semantic Web i.e. *Ontology*, with some implementation technologies: *XML, RDF and OWL*, along with a brief concluding description of some Semantic Web applications: *Semantic Desktop, Reisewissen, I-SOAS and Meta Data Layer*. Furthermore, concluding the research review we have mentioned some ontology limitations needed to be overcome.

7. Acknowledgement

We (authors) are thankful to the University of Wuerzburg Germany for giving us the opportunity to work on this research. We are thankful to the blind reviewers and publishers for being publishing this manuscript. We thank DFG (TR34/Z1) for funding.

References

[1] Tim, B. L. (1994). Universal Resource Identifiers used in the World Wide Web. RFC 1630. Publ. Internet Society, Internet Engineering Task Force.

[2] David, H., Stefano, M., David, K. (2005). Piggy Bank: Experience the Semantic Web Inside Your Web Browser. Lecture Notes in Computer Science, V. 3729, Oct, p. 413 - 430.

[3] Tim, B. L., Hendler, J., Lassila, O. (2012). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American Special Online Issue. Publ. Scientific American. 24-30.

[4] Sebastian, R., Ryszard, K., Mariusz, C., Piotr, P., Krystian, S., Adam, W., Stefan, D. (2006). Building a Heterogeneous Network of Digital Libraries on Semantic Web, *In:* Proceedings of Semantic Systems from Visions to Applications 2006, Vienna Austria.

[5] Sean, B. P. (2007). The Semantic Web: An Introduction, last reviewed February, 28. <http://infomesh.net/2001/swintro>

[6] Wernher, B (2005). Ambient Intelligence Semantic Web or Web 2.0. *In:* Proceedings of Semantic Content Engineering.

[7] Witold, A., Tomasz, K., Krzysztof, Wêcel (2005).How Much Intelligence in the Semantic Web?.Volume 3528/2005, 1-6.

[8] Heiner, S. (2002) Approximate Information Filtering on the Semantic Web. *In:* Proceedings of the 25th Annual German Conference on AI: Advances in Artificial Intelligence, 114-128.

[9] Fensel, D. (2001). Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag. ISBN 3540416021, p. 147.

[10] Okkyung, C., SeokHyun, Y., Myeongeun, O., San-gyong, H. (2003). Semantic Web Search Model for Information Retrieval of the Semantic Data. *In:* Proceedings of the 2nd international conference on Human.society@internet, 588-593.

[11] Gerhard, W., Jens, G., Ralf, S., Martin, T. (2004). Towards a Statistically Semantic Web. *In:* Proceedings of 23rd International Conference on Conceptual Modeling. 3-17.

[12] Holger, K., Mark, A. M., Natasha, F. N. (2003). Tutorial: Creating Semantic Web (OWL) Ontologies with Pro-tégé. *In:* 2nd International Semantic Web Conference.

[13] Grigoris, A., Frank, V. H. (2003). A Semantic Web Primer. The The MIT Press Cambridge, Massachusetts London, England

[14] José, S., Paulo, Q. (2004) A Methodology to Create Legal Ontologies in a Logic Programming Based Web Information Retrieval System. Artif. Intell. Law. 12(4) 397-417.

[15] Amalia, T., Laurent, R., Dalila, B. (2002). Vulcain – An Ontology-Based Information Extraction System, *In:* Proceedings of 6th International Conference on Applications of Natural Language to Information Systems-Revised. 64-75.

[16] Marc, E., York, Sure, (2004). Ontology Mapping - An Integrated Approach. *In:* Proceedings of 1st European Semantic Web Symposium. 76-91

[17] Klaus, T., Herman, M. (2006). Semantic technologies - An Introduction. Semantic Technologies Showcase the Austrian Situation. 15-20.

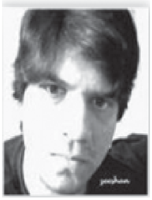
[18] Ora, L., Ralph R. S. (2007). Resource Description Framework (RDF) Model and Syntax Specification. reviewed February 2012. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>

[19] Sachin, R. (2009). Web2 to Web3 moving ahead with web technologies - lots of resources and articles on web2 and web3. reviewed 01 October. <http://sachinkraj.wordpress.com/2007/10/10/web2-to-web3-moving-ahead-with-web-technologies>
[20] The W3C Extensible Markup Language (XML). reviewed February 2012. <http://www.w3.org/XML>
[21] OWL; Web Ontology Language. reviewed February 2012. <http://www.w3.org/TR/owl-features>
[22] DAML. reviewed February 2012. <http://www.daml.org>
[23] Welcome to OIL. viewed February 2012. <http://www.ontoknowledge.org/oil>
[24] What Is An RDF Triple?. reviewed February 2012. <http://www.robertprice.co.uk/robblog/archive/2004/10/What_Is_An_RDF_Triple_.shtml>
[25] Borys, O. (2001). Learning of ontologies for the Web: the analysis of existent approaches. *In:* Proceedings of the International Workshop on Web Dynamics, held in conj. with the 8th International Conference on Database Theory.
[26] Galia, A. (2005). Language Technologies Meet Ontology Acquisition. *In:* Proceedings of the 13th International Conference on Conceptual Structures ICCS 2005, Springer. 3596. 367-380.
[27] Zeeshan, A. (2009). Intelligent - Semantic Oriented Agent Based Search (I-SOAS). *In:* proceedings of Doctoral Symposium at ACM International Conference on Frontiers of Information Technology.
[28] Zeeshan, A., Saman, M. (2011). Review Middleware Technologies, Chain Web Grid Services. *International Journal of Web Applications*, 3 (4) 197-205.
[29] Zeeshan, A. (2009). Proposing Semantic Oriented Agent and Knowledge base Product Data Management. *Information Management and Computer Security Journal*. 17(5) 360-371.
[30] Magnus, N., Malgorzata, M., Robert, T. (2006). Improving Online Hotel Search What Do We Need Semantic For? *In:* Proceedings of Semantic Systems from Visions to Applications.
[31] Horridge, M., Bechhofer, S. (2009). The OWL API: A Java API for Working with OWL 2 Ontologies. *In:* 6th OWL Experienced and Directions Workshop.
[32] Kathrin, D., Ronald, C., Annette, t. T., Nicolette, d. K. (2011). Comparison of Reasoners for large Ontologies in the OWL 2 EL Profile. Semantic Web. 2 (2) 71-87
[33] Pascal, H., Krzysztof, J. (2011). Semantic Web Tools and Systems. Semantic Web, 2 (2) 1-2.
[34] Matthew H., Sean, B. (2011). The OWL API: A Java API for OWL Ontologies. Semantic Web. 2 (1) 11-21
[35] Peter, F. P., Patrick, H., Ian, H. (2004). OWL Web Ontology Language semantics and abstract syntax. W3C Recommendation. 10 February.
[36] Bob, D. (2009). OWL 2 Web Ontology Language Document Overview. W3C Recommendation, World Wide Web Consortium, reviewed February. < http://www.w3.org/TR/ owl2-overview/>.
[37] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge acquisition. 5(2) 199-200.
[38] Quine, O. (2004). On what there is. Cambridge Belknap Press, Harvard University.
[39] Hofweber, T. (2009). Logic and Ontology. Stanford Encyclopaedia of Philosophy.
[40] Schulz, S., Stenzhorn, H., Boeker, M., Smith, B. (2009). Strengths and limitations of formal ontologies in the biomedical domain. RECIIS Rev Electron Comun Inf Inov Saude. 3 (1) 31–45.
[41] Baader, F., Calvanese, D., Mcguinness, D., Nardi, D., Patel, S. P. (2007). The Description Logic Handbook Theory, Implementation, and Applications (2nd Edition). Cambridge University Press.
[42] Zeeshan, A., Thomas, D., Saman, M. (2012). Role of Ontology in NLP Grammar Construction for Semantic based Search Implementation in Product Data Management Systems., *International Journal of Management, IT & Engineering*. 2(2) 1-40.
[43] Zeeshan, A., Saman, M., Thomas, D. (2010). Towards Design and Implementation of a Language Technology based Information Processor for PDM Systems. International Science & Technology Transactions of Information Technology- Theory and Applications, 1 (1) 1-7.
[44] Leo, S., Gunnar, A.G., Thomas, R. (2008). The Semantic Desktop as a foundation for PIM research. *In:* Proceedings of the Personal Information Management Workshop.
[45] Zeeshan, A. (2011). Designing Knowledge Base towards PDMS, *International Journal of Information Technology and Engineering*, 2(1) 9-12.
[46] Sam, J. (2003). P2P MetaData Search Layers. *In:* Proceedings of Second International Workshop on Agents and Peer-to-Peer Computing. 101-112.
[47] Zeeshan, A., Detlef, G. (2009). Design Implementation of I-SOAS IPM for Advanced Product Data Management. *In:* Proceedings of The Second IEEE International Conference on Computer, Control & Communication, 1-5.
[48] Zeeshan, A., Detlef, G. (2007). Contributions of PDM Systems in Organizational Technical Data Management. *In:* Proceedings of The First IEEE International Conference on Computer, Control & Communication, November.
[49] Zeeshan, A. (2011). Designing Flexible GUI to Increase the Acceptance Rate of Product Data Management Systems in Industry. *International Journal of Computer Science & Emerging Technologies*, 2(1) 100-109.

[50] Zeeshan, A. (2010). Proposing LT based Search in PDM Systems for Better Information Retrieval. International Journal of Computer Science & Emerging Technologies. 1(4) 86-100.
[51] Wainewright, P. (2002). Web Services Infrastructure, The global utility for real-time business. A white paper.
[52] Thomas, M., Stefan, T., Isabelle, R. (2002). Transactional Attitudes: Reliable Composition of Autonomous Web Services. *In:* Proceedings of Workshop on Dependable Middleware-based Systems, International Conference on Dependable Systems and Networks. IEEE, ISBN: 0-7695-1597-5, p. 792.
[53] Grit, D., Son, N., Andrew, T. (2004). OWL-S Semantics of Security Web Services: a Case Study. The Semantic Web: Research and Applications, Publ. Springer, V. 3053, 240-253.
[54] Visit, H., Vuong, T. X. (2005). Semantic Web Agent Communication Capable of Reasoning with Ontology and Agent Locations. World Academy of Science, Engineering and Technology. 10.
[55] Daniel, J. L. (2008). Intelligent agents and the Semantic Web; Developing an intelligent Web. IBM Technical Library.
[54] Google Translate. Last reviewed 1st March 2012, <http://translate.google.com/translate_tools>.
[55] iHOP. Last reviewed 1st March 2012, < http://www.ihop-net.org/UniPub/iHOP/ >.
[56] Perez-Iratxeta, C., Pérez,AJ, Bork, P., Andrade, M. A. (2003). Update on XplorMed: a web server for exploring scientific literature. Nucleic Acids Research. 31, 3866-3868.

Authors Bibliography

Prof. Thomas Dandekar is the Chair of the Department of Bioinformatics, Biocenter, Am Hubland, University of Wuerzburg Germany, and interested in combining genome-based bioinformatics with systems biological modelling.



Zeeshan Ahmed is the software engineer, researcher at the Department of Bioinformatics, Biocenter, Am Hubland, University of Wuerzburg, Germany, and has interestes in developing intelligent software systems towards product data analysis, visualization and management.

Saman Majeed is the doctoral scientist the Department of Bioinformatics, Biocenter, Am Hubland, University of Wuerzburg, Germany.

Call for Papers

The Fourth International Conference on the Networked Digital Technologies (NDT 2012)
April 22-24, 2012
Canadian University of Dubai
UAE
http://www.ndtconf.org

Proceedings will be published by Springer CCIS series

The proposed conference on the above theme will be held at the Canadian University of Dubai, UAE from April 24-26, 2012 which aims to enable researchers build connections between different digital applications.

Currently, a number of institutions across the countries are working to evolve better models to provide collaborative technology services for scholarship by creating shared cyberspace thro expert collaboration, but this is a challenge for the institutions for a number of reasons. In the last few years, the landscape of digital technology applications projects for the various disciplines in humanities, social sciences, and sciences appears induced by many initiatives. For the creation of research clusters, the research community has thousands of databases, websites, local computing clusters, and web-based tools around individual themes, interests and projects. In most cases, these tools and resources are and were created to meet the specific needs of a particular community. In many cases, the funding and support for these critical initiatives is fragile and temporary, and directed in piecemeal fashion. There is a need to provide concerted efforts in building federated digital technologies that will enable the formation of network of digital technologies.

- Information and Data Management
- Data and Network mining
- Intelligent agent-based systems, cognitive and reactive distributed AI systems
- Internet Modeling
- User Interfaces, Visualization and modeling
- XML-based languages
- Security and Access Control
- Trust models for social networks
- Information Content Security
- Mobile, Ad Hoc and Sensor Network Management
- Web Services Architecture, Modeling and Design
- New architectures for web-based social networks
- Semantic Web, Ontologies (creation , merging, linking and reconciliation)
- Web Services Security
- Quality of Service, Scalability and Performance
- Self-Organizing Networks and Networked Systems
- Data management in mobile peer-to-peer networks
- Data stream processing in mobile/sensor networks
- Indexing and query processing for moving objects
- User interfaces and usability issues form mobile applications
- Mobile social networks
- Peer-to-peer social networks
- Sensor networks and social sensing
- Social search

- Information propagation on social networks
- Resource and knowledge discovery using social networks
- Measurement studies of actual social networks
- Simulation models for social networks
- Green Computing
- Grid Computing
- Cloud Computing

All the papers will be reviewed and the accepted papers in the conference will be published in the “Communications in Computer and Information Science” (CCIS 136) of Springer Lecture Notes Series (www.springer.com/series/), and will be indexed in many global databases including ISI Proceedings and Scopus.

The NDT proceedings are indexed by dblp (http://dblp.uni-trier.de/db/conf/ndt/)

In addition, selected papers after complete modification and revision will be published in the following special issues journals.

1. Journal of Digital Information Management (JDIM)
2. International Journal of Information Studies (IJIS)
3. International Journal of Green Computing (IJGC)
4. International Journal of Web Applications (IJWA)
5. Journal of E-Technology

(http://www.dline.info)

General Chairs
Rachid Benlamri
Lakehead University, Canada
Driss Guerchi
Canadian University of Dubai, UAE

Program Chairs
Abdallah Shami, University of Western Ontario, Canada
Eric Monacelli, Versailles University, France
Lorna Uden, Staffordshire University, UK

Workshop Chairs
Russel Pears, Auckland University of Technology, Auckland, New Zeland
Sun Aixin, Nanyang Technological University, Singapore
Hiroshi Ishikawa, Shizuoka University, Japan
Boubaker Boufama, Windsor University, Canada

Publicity Chair
Essa Ibrahim Basaeed, Khalifa University, UAE

Local Arrangements Chair
Emad Eldin, Canadian University of Dubai

Local Arrangements Co-Chairs
Talal Kursany, Canadian University of Dubai
Sherif Moussa, Canadian University of Dubai
Siwar Rekik, Canadian University of Dubai
Hussam Abuazab, Canadian University of Dubai

The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2012)

August 27-29, 2011
University of Cyprus, Nicosia, Cyprus
http:/www.dirf.org/diwt

Technically co-sponsored by IEEE Communications Society

Proceedings will be published by IEEE Xplore

The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2012) is a forum for scientists, engineers, and practitioners to present their latest research results, ideas, developments and applications in the areas of Computer Communications, Communication networks, Communication Software Communication Technologies and Applications, and other related themes.

This conference (ICADIWT 2012) will include presentations of contributed papers and state-of-the-art lectures by invited keynote speakers.

This conference welcomes papers address on, but not limited to, the following research topics:

- Computer Communication
- Communication Technologies
- Forensics, Recognition Technologies and Applications
- Communication Software
- Fuzzy and neural network systems
- Signal processing, pattern recognition and applications
- Digital image processing
- Speech processing
- Computational biology and bioinformatics
- Computer networks
- Information retrieval and internet applications
- Software engineering
- Data Communication
- Digital Communication
- Databases and applications
- Distributed Computing
- Data mining
- Real time systems
- Computer and network security
- Multi-Agent systems

Program Committees

General Chair
George Angelos Papadopoulos, University of Cyprus, Cyprus

Program Chairs
Weimin He, University of Wisconsin Stevens-Point, USA

Joel Rodrigues, Institute of Telecommunications/University of Beira Interior, Portugal

Modified versions of the selected papers of the conference will be published in the following peer reviewed journals

1. Journal of Digital Information Management
2. International Journal of Web Applications
3. International Journal of Information Studies
4. Journal of E-Technology
5. Journal of Information Technology Review

Submission of papers	**January 31, 2012**
Notification of Acceptance	**March 31, 2012**
Camera ready	**May 31, 2012**
Registration early bird	**June 30, 2012**
Registration late	**July 20, 2012**
Conference dates	**August 27-29 2012**

Conference Mail: diwt@dirf.org

Seventh International Conference on Digital Information Management (ICDIM 2012)

August 20-22, 2012
University of Macau
Macau

Technically co-sponsored by IEEE Technology Management Council

Proceedings will be indexed in IEEE Xplore

General Chair
Simon Fong, University of Macau, Macau
http://www.icdim.org
Email: conference@icdim.org

000000000000000000000000000000000000000000000000000000000000000000000000000000000000