SESSION 5A

DISTRIBUTED COMPUTING
A Survey of Blind Search Techniques in Structured P2P Networks

Jamie Furness, Mario Kolberg
Dept. of Computing Science and Mathematics, University of Stirling
{jrf,mko}@cs.stir.ac.uk

Abstract—The ability to perform complex queries is one of the most important features in many of the P2P networks actually deployed today. While structured P2P networks can provide very efficient look-up operations via a Distributed Hash Table (DHT) interface, they traditionally do not provide any methods for complex queries. This can be attributed to the use of consistent hashing, which causes data to be distributed uniformly over the entire network. Since consistent hashing does not preserve locality there is no guarantee, in fact it is highly unlikely, that similar search terms will have their data stored together. This means in a simple DHT it is not possible to perform range queries, wild-card or full-text searching, which limits their application in the real world.

In this work we review the existing methods for performing complex queries on top of structured P2P networks; focusing on methods which allow for full-text search rather than only keyword queries. It should be obvious that to perform blind search with support for full-text queries the query must be processed locally at each node, and as such the problem of blind search is almost identical to the problem of efficiently broadcasting; with the difference that queries need not always reach all nodes to be successful. The majority of existing algorithms for performing complex queries on top of structured P2P networks exploit the structure inherent in DHTs to efficiently broadcast the search query over the entire network; this allows every node in the network to process the query locally, removing the restrictions placed on the complexity of queries. We focus on methods which allow for full text search rather than only keyword based queries, and refer to this method of searching within a DHT as broadcast search. Some alternate methods [2], [3], [4], [5], [6] which do not require broadcasting can still perform wild-card and range queries, but are limited to keyword-based search rather than full-text search so we do not cover them here.

I. INTRODUCTION

In the kind of P2P networks found today locating data without the use of complex queries is almost impossible. For example to locate a song in a specific format with the bit-rate within a certain range without the use of complex queries would require the user to guess the exact file naming scheme used and perform a new query for every possible bit-rate value within the desired range. Obviously this is not a sensible approach, and can be one reason why structured P2P networks are not as popular in the real world.

While structured P2P networks can provide very efficient look-up operations via a Distributed Hash Table (DHT) interface, they traditionally do not provide any methods for complex queries. This can be attributed to the use of consistent hashing, which causes data to be distributed uniformly over the entire network. Since consistent hashing does not preserve locality there is no guarantee, in fact it is highly unlikely, that similar search terms will have their data stored together. This means in a simple DHT it is not possible to perform range queries, wild-card or full-text searching, which limits their application in the real world. Unstructured networks usually implement wild-card search by a form of flooding or random walks, however flooding is inherently inefficient due to the large number of redundant messages sent [1], and random walks tend to be slow with no guarantee of actually finding the data even if it exists.

It should be obvious that to perform blind search with support for full-text queries the query must be processed locally at each node, and as such the problem of blind search is almost identical to the problem of efficiently broadcasting; with the difference that queries need not always reach all nodes to be successful. The majority of existing algorithms for performing complex queries on top of structured P2P networks exploit the structure inherent in DHTs to efficiently broadcast the search query over the entire network; this allows every node in the network to process the query locally, removing the restrictions placed on the complexity of queries. We focus on methods which allow for full text search rather than only keyword based queries, and refer to this method of searching within a DHT as broadcast search. Some alternate methods [2], [3], [4], [5], [6] which do not require broadcasting can still perform wild-card and range queries, but are limited to keyword-based search rather than full-text search so we do not cover them here.

II. BROADCAST SEARCH TECHNIQUES

A. Efficient Broadcast

In [7] an algorithm for broadcasting complex queries, or indeed any data, over DHTs is proposed by El-ansary et al. using Chord [21] as an example. In Chord nodes form a ring structure, in a network of size $N$ each node maintains routing state information for at most $\log(N)$ other nodes, namely the node succeeding it, known as the successor, and a set of finger nodes. The finger nodes are chosen at logarithmically increasing distance around the ring, the $i^{th}$ entry in the table at node $n$ contains the identity of the first node that succeeds $n$ by at least $2^{i-1}$ ($i \geq 1$).

In the proposed algorithm, to initiate a query, a node $n$ will send the query along with a limit to every (non-redundant) node in its finger table. The limit parameter given is the identifier of the next finger in the finger table, and is used to restrict the forwarding space of the receiving node to $[n, \text{limit}]$. The last node in the finger table is given a limit of the originating node. When the message is received by a node it forwards the broadcast to any fingers it has within the defined forwarding space, giving each one a new limit.

Using simulation El-ansary et al. show that there are no redundant messages generated and that exactly $N - 1$ messages are needed to cover every node. The network was
first populated with \(2^3\)–\(14\) nodes and then the broadcast was initiated. During the simulations no nodes ever joined or left the network, so the effects of churn were not taken into account. Due to the tree structure used node failures have a large impact on the performance of efficient broadcast. For example in figure 1 if node \(N4\) was to fail then half the network would not receive the message. To confirm this we ran simulations with a 10,000 node network and lifetime mean ranging from 100 seconds to 10,000 seconds; the maintenance delay of Chord was set to 10 seconds and 120 seconds. As can be seen in figure 2 the success rate drops dramatically when the lifetime mean drops below around 30 minutes.

Since the Efficient Broadcast algorithm was proposed it has been extended and improved in various different ways, outlined below.

1) Self-correcting Broadcast: Extending on their own work, in [8] Ghodsi et al. claim the algorithm proposed will fail to cover all nodes when the routing tables are not up-to-date. A similar algorithm called Self-correcting Broadcast is proposed [9], which uses a correction-on-use technique over DKS [22].

Distributed K-ary Systems (DKS) are configured with parameter, \(k \geq 2\), such that the look-up length is guaranteed to take at most \(log_k(N)\) hops for a network of size \(N\). Each node maintains a routing table, consisting of \(log_k(N)\) levels. At each level \(l\) a node \(n\) has a view of the identifier space defined as \([n, n \oplus N/2^l]\). This means that at level one, the view consists of the whole identifier space, and at any other level, one \(k^{th}\) of the previous level is considered. At every level the view is partitioned into \(k\) equally-sized intervals. This method for partitioning the identifier space is known as the k-ary principle, and is described fully in [9].

In this algorithm a node starting the broadcast iterates through all levels of the routing table, starting at the first level. At each level, the node moves in counter-clockwise direction through all of its intervals, broadcasting a message to each responsible node \(r\). Each broadcast message carries with it the parameters \(l\) (level), \(i\) (interval) and the \(limit\). The message first delivers the intended data to the receiving node. Secondly it serves as a request to cover all nodes in the interval \([r \oplus i \times N/2^l, limit]\). Due to outdated routing tables some intervals might not seem to have any nodes even though they are populated. The responsibility of covering those intervals is delegated to the next interval.

If node \(n\) gives another node the responsibility to cover other preceding intervals and other nodes exist in those intervals, the node will trigger correction-on-use, and the routing information will be corrected at node \(n\).

Via simulation, in [8] it was shown that coverage is always 100% and no redundant messages are ever received. Network sizes of 500, 1000, 2000, 3000, 4000 were used, to start 10% of the nodes were added then the broadcast was initiated while the other 90% of nodes joined. While these results show that

<table>
<thead>
<tr>
<th>Proposal</th>
<th>Section</th>
<th>Reference</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Partitioning Search (RPS)</td>
<td>II-A3</td>
<td>[10], [11], [12]</td>
<td>Proposes using a mixture of efficient broadcast and epidemic communication to achieve a better success rate under churn.</td>
</tr>
<tr>
<td>Dynamic Querying over DHT</td>
<td>II-A5</td>
<td>[14], [15], [16]</td>
<td>Uses an iteratively increasing TTL value to decrease query traffic without affecting success rate.</td>
</tr>
<tr>
<td>Efficient Broadcast in P2P Grids</td>
<td>II-A6</td>
<td>[17]</td>
<td>Introduces a TTL value and other optimizations to try and decrease query traffic without affecting success rate.</td>
</tr>
<tr>
<td>An Efficient Broadcast Algorithm</td>
<td>II-A7</td>
<td>[18]</td>
<td>Attempts to detect and handle failed nodes, but the implementation seems unfinished.</td>
</tr>
<tr>
<td>Pastry’s broadcast mechanism</td>
<td>II-B</td>
<td>[19]</td>
<td>Describes a method for efficiently broadcasting over a Pastry network, which can be used for complex queries.</td>
</tr>
<tr>
<td>CAN broadcast</td>
<td>II-C</td>
<td>[20]</td>
<td>Describes a method for efficiently broadcasting over a CAN network, which can be used for complex queries.</td>
</tr>
</tbody>
</table>
nodes joining the network are handled successfully they do not take into account nodes departing the network. Since this algorithm is based on the Efficient broadcast algorithm, nodes departing are likely to have a similar effect in both.

2) Pseudo-reliable Broadcast: In [9] Ghodsi notes that the current algorithms will be providing best effort delivery in the presence of failures. In many cases a best effort broadcast may not be sufficient. Assuming the initiating node attempts to broadcast and it has $M = \log_2(n)$ pointers (as in Chord), the broadcast will partition the space into $M$ intervals, the last of which will cover roughly half of the identifier space (see figure 1). If the node responsible for the last partition was to fail then roughly half the nodes will not receive the broadcast.

In the Pseudo-reliable Broadcast algorithm every broadcast has a globally unique random identifier associated with it, which is included in every message. Nodes keep track of previously seen identifiers and filter any message which has an identifier previously seen, hence redundant messages are filtered. Each node has an ACK set, used to keep track of the children it is responsible for. In addition to keeping the identities of the children the algorithm also keeps track of the interval delegated to each child. The algorithm is made resilient to failures by using a bulk operation [9] to resume after a failure. Whenever a node suspects that one of its children has failed, it uses the ACK set to determine the interval delegated to the failed node. Thereafter, the bulk operation is used to cover all nodes in that interval.

The failure detector suggested uses timeouts, meaning it could potentially give false-negatives, suspecting that an alive but busy/slow node has failed. The only consequence of this however will be wasted bandwidth consumption from the redundant messages being sent.

The time it takes for a broadcast to complete depends on the depth of the broadcast tree. Therefore, using a timeout when waiting for an acknowledgment from a child is problematic, as the parent does not know the depth of its sub-tree and therefore cannot determine the time to wait before triggering a timeout. Because of this issue the failure detector is assumed to be implemented independently from the broadcast, this can be achieved by the failure detector periodically sending a message to its children and awaiting an acknowledgment.

However, this algorithm still sends redundant messages after a failure, and could be improved. Consider a node who’s children are all done covering their delegated intervals, except one. If the node fails, it’s parent will detect that and reassign the interval to a new node. The new node will attempt to cover all of the original nodes children, rather than just the remaining one. To avoid these redundant messages, nodes could periodically send an update of their current ACK set to their parent. If this was the case, when a node failed only the children it was waiting for would need to be covered rather than all of them.

3) Recursive Partitioning Search (RPS): In [11] a framework called Recursive Partitioning Search (RPS) for blind search over structured peer-to-peer overlays is presented by Vishnevsky et al., with a realization for Chord [21]. RPS is a version of the Efficient Broadcast algorithm from section II-A, however it includes a TTL value as well as a limit, and a node will only broadcast the query to its sub-tree if it cannot satisfy the query itself first. With enough data replication the whole network does not need to be searched to find the data, and once we have found one copy we have no need to continue looking.

Vishnevsky et al. consider ways to restrict the number of nodes visited but without reducing the query success rate dramatically. In a network of size $N$, if the TTL value is set to $\log(N)$ then in a stable network the algorithm results in a 100% success rate, so it is a maximum reasonable value for the TTL. In most cases a smaller TTL value can still achieve high success rates because most resources are replicated among multiple nodes.

A variation of the algorithm to control the query message traffic is as follows: The query originator selects a partial list of its fingers and sends the query to them first. The tag in the message now contains two values, the nodes local search limit and the global search limit. Receiving nodes update the local limit as before, but do not touch the global limit. This variation allows a node to divide the search space and perform RPS sector by sector sequentially, if required.

In [10] realizations of RPS for both Chord [21] and Pastry [23] are presented. To reduce network load the use of go-stop signals (RPS+G) and local indices (RPS+L) are suggested.

Typically blind search algorithms generate multiple query messages that traverse the network concurrently, thus even when one of the query messages finds a result the others continue to search. To reduce the number of these messages some intermediate nodes inquire if the search originator has already found the target resource. The initiator either sends a “go” signal if the search is still pending, or a “stop” signal to terminate the query message. By using go-stop signals the scalability of blind search algorithms can be enhanced.

Vishnevsky et al. performed simulations over both Chord and Pastry, allowing a comparison of the two overlays. Using a network with 1,000 nodes, it was found that performing RPS over Pastry not only provided a higher success rate, but was faster than Chord. For a 95% success rate over Chord a TTL value of 6 was required, giving a mean delay of 99ms and 0.43 messages/sec per node. However, for a success rate of 99% over Pastry a TTL value of 3 was sufficient, giving a mean delay of 42.8ms and 0.27 messages/sec per node. This was mainly due to the Pastry routing table being roughly twice the size of the Chord routing table.

The use of RPS+G and RPS+L showed a reduction in network load by almost 50% (using the Chord implementation), as shown in Figure 3.

Extending this work by briefly looking at the effects of churn [12] notes that when RPS sends a query message to a node, it is expected that the node will propagate that message to other nodes which are within its search sector. If a node fails before doing so, all other nodes within its sector will not receive the query.

Simulations using Pastry showed that for a 1% increase
in churn rate (in this case churn rate was taken to mean the ratio of nodes temporarily out of the network to the total number of nodes) there is roughly a 2% drop in the amount of nodes receiving the broadcast. However due to the duplication of documents in the network, RPS exhibits quite a limited dependence on peer churn. In the simulations performed there were 100 unique resources, with duplication ranging from 0.01% to 10% depending on the documents popularity (modeled using a zipf-like distribution). For each simulation, 1,000 queries were sent. What resource to query for was chosen based on the resources popularity. RPS showed to have a success rate of about 99% even if 15% of peers have failed. In [24] we discuss the effects of different data replication strategies on churn.

4) A Partition-based Broadcast Algorithm: Another partition based approach is proposed in [13] which works by hierarchically partitioning the identifier space into two subspaces to construct a binary partition tree. Upon receiving a broadcast message with a limitation value \( l \), each node \( n \) partitions its limited identifier space \((n, l)\) into two subspaces \((n, f)\) and \((f, l)\), where \( f \) is the closest finger node to the limitation value \( l \). Next, the node \( n \), selects the first finger found in \((n, f)\) as its left child, and \( f \) as its right child. Finally the node forwards the broadcast message with the limitation value \( f \) to its left child and with the limitation value \( l \) to its right child.

The main difference between this algorithm and the Efficient Broadcast algorithm is that each node sends to at most two children. This has the effect that the broadcast tree is balanced and hence the branching factor is fixed at 2, but at the cost of increased hop count and hence time taken for a broadcast to complete.

5) Dynamic Querying over DHT: In [14] Talia and Trunfio propose a blind search method known as Dynamic Querying over Distributed Hash Tables (DQ-DHT) which uses the Dynamic Querying technique [15] originally developed for unstructured networks in combination with the Efficient Broadcast algorithm described in section II-A. Dynamic querying is interested in ensuring only the required amount of results are obtained from a search to avoid wasted effort in finding 100% of matches, this is done using a small TTL and iteratively increasing it as required. This approach is converted to work over DKS by Trunfio et al. in [16].


Like the Efficient Broadcast algorithm, the broadcast message contains a limit argument \( l \). When a node receives the broadcast message, it picks the finger nearest to the middle \( (m) \) between its own address and \( l \), and sends the broadcast message to it. The node itself continues the broadcast distribution within an interval between itself and \( m - 1 \). However it is again noted that a failure will make the entire sub-tree under the node unreachable. To solve this, all nodes that have received the broadcast, periodically forward the broadcast information to their randomly chosen neighbours as part of epidemic communication.

Using simulations in [17] it is shown that the combination of efficient broadcast and epidemic distribution will reach more nodes that simply efficient broadcast does under churn, and significantly faster than purely epidemic communication.

Due to its use of epidemic communication, this method obvious does result in duplicate messages being generated, however through simulation it is shown that before failure the algorithm distributes the broadcast almost entirely by means of efficient broadcast, with the message complexity increasing after simulated node failure. It is important to note that the message complexity is always significantly lower than that of purely epidemic communication.

7) An Efficient Broadcast Algorithm: In [18] an algorithm is proposed in which the ring is split into \( d \) equal sized partitions. Each partition has a responsible node, which is the first node encountered in the partition when moving clockwise around the ring. To contact the node a message can be routed using the underlying DHTs routing function. This means that the responsible nodes don’t need to be in the partitioning nodes routing table, however it does mean traversing one level in the broadcast tree is not strictly a one-hop operation like in the efficient broadcast algorithm.

In a similar way to the Pseudo-reliable Broadcast algorithm (section II-A2) each node maintains an ACK set, containing the nodes they have forwarded the broadcast to and their allocated partition. After a node has received an ACK from all of its children (or if its a leaf node, right away), it will send an ACK to its parent node. Nodes maintain a timer for each partition, if a node does not receive an ACK from a child node within the given timeout it will select the next node from the child nodes successor list and retransmit the broadcast message to the this new node. At first this may sound good, but it seems to have a major flaw; if the child node is dead how can its successor list be accessed? At possibly work around could be to just retransmit the message to the partition again using the DHTs routing function; if the first node has died it should eventually be removed from the routing tables by the DHTs maintenance algorithm.

The timeout used by a node at layer \( i \) in the broadcast tree is calculated using \( T_i = n_{suc} \times T_B \), where \( n_{suc} \) is the number of successors a node has (a design choice of the DHT), and the base timeout is similar to that in TCP, defined as \( T_B = 2(\Delta + 4\sigma) \) where \( \Delta \) is the mean time taken for one
hop and the standard deviation $\sigma$ is $0.1\Delta$. In the paper there is no indication as to what $h$ stands for, however for the formula to make sense we assume it must refer to the depth of the broadcast tree. This however is a problem as it is not possible for a node to actually know this value. In [25] an algorithm for network density estimation is presented that could be used to estimate the depth of a broadcast tree, however it only provides a rough estimate and adds extra delay and message complexity to the operation.

B. Pastry’s broadcast mechanism

In [19] a method for broadcast over Pastry is described. While not specifically designed with search in mind, it can still be used to broadcast complex queries.

Each pastry node maintains a routing table, a neighborhood set, and a leaf set. In a network of size $N$ using identifiers with base $2^B$, each nodes routing table is designed with $\log_B(N)$ rows, where each row holds $B - 1$ entries. All the entries at row $r$ of the routing table each refer to a node whose identifier shares the current node’s identifier in the first $r$ digits, but whose $(r + 1)^{th}$ digit does not match that of the current node.

A node broadcasts a message by sending the message to all nodes in its routing table; each message is tagged with the routing table row. When a node receives a message tagged with routing table row $r$ it forwards the message to all nodes in its routing table with rows greater than $r$. This continues until a node receives a message tagged with $r$ and it has no entries in rows greater than $r$.

In [26], using simulations with a Pastry network of size 10,000 and average session times of $\{5, 15, 30, 60, 120, 600\}$ minutes, it is shown that the query success rate starts to drop dramatically when the average session time drops under 30 minutes.

We ran simulations with a 10,000 node Pastry network with lifetime mean ranging from 100ms to 10,000 ms which showed similar results, as can be seen in figure 4.

[27] presents a performance analysis of Pastry’s broadcast mechanism using an analytical model and simulations. The replication load and hop count are studied, with comparisons drawn to Scribe [28].

C. Content Addressable Network (CAN) broadcast

A solution for application-level multicast in CAN is presented in [20]. In their solution multicast is performed by creating a mini-CAN of participating users, then broadcasting over that CAN. While using this technique specifically for complex queries isn’t suggested, broadcasting is the basis of most complex querying techniques.

CAN is a rather unique type of overlay, designed around a virtual $d$-dimensional Cartesian coordinate space. The entire coordinate space is dynamically partitioned among all the nodes in the system such that every node owns an individual zone. Keys are mapped onto a point in the coordinate space using a consistent hash function, and the responsible node is the node whose zone contains that coordinate.

In a CAN with dimension $d$, each node has at least $2d$ neighbours; one to move forward in dimension $d$ and one to move backwards. To initiate a broadcast the source node forwards the message to all its neighbours. When a node receives a broadcast message from a node with which it neighbours along dimension $i$, it will forward the message to those neighbours along dimension $1...i-1$ and the neighbours in dimension $i$ on its other side. To prevent the message from looping back around a node does not forward a message along a particular dimension if that message has already traversed at least halfway across the space from the source coordinates along that dimension. An example broadcast can be seen in figure 5.

It is worth noting that for a perfectly partitioned coordinate space the algorithm ensures each node receives the message exactly once. For an imperfectly partitioned space however, a node may receive the same message from more than one neighbour. For example, in figure 5, node $E$ would receive a message from both neighbours $C$ and $D$.

III. CONCLUSIONS

This paper presents an overview of current blind search methods for structured P2P networks. It should be obvious that to perform blind search with support for full-text queries the query must be processed locally at each node, and as such the problem of blind search is almost identical to the problem of efficiently broadcasting; with the difference that queries need not always reach all nodes to be successful. Broadcast-based schemes allow for processing of any type of query but the total message complexity increases linearly with the network size. While this is much less efficient than standard operations in a DHT, which can usually be completed with $\log(N)$ messages, it is a major improvement over flooding in unstructured networks where large numbers of duplicate messages are generated. It is however important to note that
broadcast-based schemes tend to perform badly under high churn rates [24]. Some methods (II-A2, II-A7) attempt to handle this by detecting failures with a timeout and retransmission, however this adds to both the complexity and search latency; other methods (II-A6) combine efficient broadcasting with flooding, allowing some redundant messages to be generated, resulting in a higher success rate than efficient broadcast but more quickly and efficiently than flooding alone. From this we can see that while it is possible to broadcast with near 100% success rate, all approaches have a negative effect on some other aspect of the system.

Methods for restricting the search space (II-A3, II-A5) show that it is possible to reduce the overall number of nodes queried without altering the success rate, and due to data replication a 100% message success rate is not necessarily required to achieve a 100% query success rate.

We feel there is a notable lack of work on performing complex queries in one-hop networks, and in future work aim to adapt the efficient broadcast algorithm to work on the variable-hop overlay Chameleon [29], which aims at having one-hop performance for high bandwidth peers and multi-hop performance for low bandwidth peers. Instead of trying to recover from failures we aim to reduce the number of failures in the first place by taking advantage of Chameleon’s high routing table accuracy, and mitigate the effect of failures with the use of a novel data replication strategy. Making use of the fact that Chameleon produces a variable-hop network will allow forwarding of messages to $2^N$ nodes depending on available bandwidth, allowing for a heterogeneous network in which low bandwidth peers use less bandwidth than required by efficient broadcast but with higher performance assuming a high enough proportion of high bandwidth nodes.

REFERENCES


Investigating Sharing in Memory for Life Systems

Haseeb Ur Rahman, Madjid Merabti, David Llewellyn-Jones, Sud Sudirman
H.Ur-Rahman@2008.ljmu.ac.uk, {M.Merabti, D.Llewellyn-Jones, S.Sudirman}@ljmu.ac.uk
School of Computing and Mathematical Sciences,
Liverpool John Moores University,
Liverpool, L3 3AF, UK

Abstract: Memory for Life (M4L) systems store and organize life events captured by people in digital form using their cameras, mobile phones and so on. This paper describes M4L systems and the challenges for sharing digital events. Based on the challenges, an investigation is carried out in order to find a suitable technology that allows sharing of digital events according to the social network of a user. For this purpose, Web-based online social networks and peer-to-peer networks are particularly studied. The requirements for a social P2P model for sharing human digital events (HDEs) are suggested as future work.

I. Introduction

Human life is filled with many events which people try to capture or store in different ways. In the era before reading and writing was widespread, the only way to keep these events was though “learning by heart” either in simple story form or in the form of verse. These events were then told to people by a poet who captured them in the form of verse or as story tellers in large gatherings. In some ages, symbolic or pictorial forms were also used to record different events which we see in the form of ancient remains in different regions of the world. After the introduction of pen and paper, monarchs of different kingdoms used to keep annalists/historians in their court to record the events that happened in the period of their ruling. With the start of the modern age, other forms of recording these events were introduced such as through pictures, audio, video etc. and nowadays the development of technology has produced high quality data capturing and huge data storage devices at a low cost.

In fact, it’s not unusual for people in all ages to create memoirs, recalling different events in their lives with different people and at different places and times. Many people want to keep their memories in some form or other: either in written paper form or in the form of pictures, videos etc. Human lives are filled with so many events, but after some time or reaching a specific age, many people find they have forgotten many of them. Now, thanks to technological developments, it’s possible to have a record of the most beautiful events in our lives in the form of pictures, audio, video and so on, all recorded in digital form. However the need remains to allow them to be organized in such a way that reminds us not only of the place and time but also the feelings we were having at those times. M4L [1] is such an effort to enable collecting, organizing and sharing of such events, bringing the computer world and the physical world closer to each other. We refer to an event captured as a memory in digital form as a human digital event (HDE).

M4L is a new research area with the aim to help in storing and managing data properly and extracting different information like lifestyle, stories, medical history, interest etc. about the life of a person. Data will be annotated at the time of capture or storage, automatically via the device that is capturing or storing the data. The parameters for annotating data could include GPS location, time, object names, events, temperature, and so on. This data could be used for a variety of interesting and useful reasons, often allowing extensions beyond the purposes for which it was originally intended [2]. For example, pictures could be used to understand a person’s social life by counting the number of people in a picture, while the same data could be used to establish the state of a person’s health using face detection software to compare different photos. The information stored could also be used to generate daily schedule for the person and stories about their life by collecting connected information. It can help with a child’s education by collecting information about their behavior, different approaches for solving problems across the world and so on. Also, the schedule of elderly people, who have weak memories, might also be predicted by analyzing their previous routines. These digital events which capture the most memorable minutes of life are not only to be stored, but also to be shared with other people.

Consider a scenario of a wedding ceremony in which both wedding bride or bridegroom and guests capture the event using their cameras, mobile phones etc. Each party capture the same event as their memories in digital form, but the point of view of each of them is different while capturing it. For example, a bridegroom might capture it to see who came to their wedding, how many guests there were, how the arrangements went and so on. The guests, on the other hand, might keep the memories to see the expressions and outfits of the bride and groom, and to keep a record of the speeches that were made. This attracts people, even those who have captured the events and also those that have missed the event, to see the point of view of each person through their memories. This behavior results in sharing and/or collecting the memories of other people. This process is also described by Olsson et al. [3].

The aim of this paper is to identify the issues and challenges in the sharing of HDEs. The challenges are illustrated by considering the social priorities and the requirements of M4L systems for sharing data. Currently used technologies for carrying online social activities are investigated. The purpose is to find an appropriate option that has the potential to carry out social activities and allow M4L systems to perform their functions as required.

This paper is organized as follows. In Section 2 we explain the research so far for collecting and organizing HDEs. Section 3
gives an overview of the challenges that need to be overcome for sharing these events. Each issue is explained with examples in the scenario of a M4L system. The following section investigates Web-based online social networks and peer-to-peer networks (P2P) for sharing HDEs, in terms of the challenges. The last section, Section 5, concludes the discussion and describes the requirements for a social P2P model for our future work.

II. M4L SYSTEMS

The idea behind the Memory for Life was originally conceived by Vannevar Bush in 1945 in the article “As We May Think” where it took the form of a machine called Memex: “A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory” [4]. It was posited that only one button push could retrieve all the data you need in a small amount of time. Gorden Bell’s MyLifeBits [5], inspired by the idea of Vannevar Bush, was developed with the intention that with the tools and technology and the relatively-speaking large storage devices available at the time, it would be possible to collect and organize all of our data easily. The MyLifeBits software is able to store text, images, links, videos etc. in a database and annotate. The annotation is currently manual, but in future some hardware or software solutions are expected to annotate the data automatically. Jim Gemmell et al. [6] described the four principles for designing MyLifeBits. First, there should be no strict hierarchy for organizing data. Second, many visualizations of their life bits were desirable to help understand what they would be looking at. Third, the value of non-text media is dependent on annotations. Fourth, authoring tools create two-way links to media that they included with new media.

Another project with similar aims to those of MyLifeBits is Haystack [7], which facilitates users in organizing and managing their emails, documents, appointments, tasks etc. Total Recall [8, 9] collects data through personal sensors such as cameras, microphones etc. and organizes and annotates them for future use under security and privacy constraints so that the collected data cannot be used by the wrong people. Eyetap [10], similar to Total Recall, gathers data using a small camera that is attached near an eye and stores all the things which a human eye sees as part of its normal routine. Memory for Life [11] not only collects and manages data but also analyses in a way that shows different aspects of a person’s life and helps him in his normal routine if he does something unusual or forgets to do it. Azizan et al. [12] describe a Human Life Memory system for collecting, storing and organizing different life events which they call “Seredipitous Moments”, as well as discussing sharing via P2P networks. JXTA is used as the peer-to-peer networking technology for this system.

As an example, an interesting feature of Memory for Life is to generate stories about the life of a person. These stories are generated from the stored data of a user and possibly from the data of his friends and family members, since other people may also be an integral part of the scene or story. Therefore the stored events of other members may also be required to complete the story that a particular user is interested in. Incorporating the stored events of others into a story in this way is possible by allowing the sharing of events between users.

III. CHALLENGES FOR SHARING HUMAN DIGITAL EVENTS

The Memory for Life systems will store every aspect of a human’s life, which in theory should make it easy to know their interests, lifestyle, social activities, schedule, stories and so on. However there are several challenges which must be overcome when using these systems over a computer network. These are given below.

Data privacy: The first challenge is to ensure data privacy; the system must ensure that the personal data stored is accessible only by those people that are allowed to access it, and that it should never be stored on any unwanted system. The user of the system should be able to adjust the level of privacy of his data to allow some people to access, yet limit access to others. The interesting aspect about data privacy of M4L systems is that it requires the information to be kept hidden not only from unknown people but also from different groups of known people. In the latter case, the differences in privacy could be because of age, culture and so on. For example, husband-wife and child-parent relationships are two different trusted relationships. The difference is in part due to age, because usually at an early age the child-parent relationship is a trusted relationship, but after getting married a husband-wife could become more trusted then a child-parent.

User’s control over data: We believe that users’ control over data is an important feature of any social P2P system. Data sharing should be based on a user’s choice, which means that a number of accredited people can be authorized to access some portion of a user’s human digital events, whereas the user can deny access to others. Furthermore, data access is based on a user’s trust level, which implies that a user will only be allowed to access the data when their trust level is equivalent to or above a predefined trust threshold value. The two important elements are that data should be under a sharer’s control: a) while sharing data and b) when data is shared. In the first case, data should be accessible only to those people that are authorized by the user. For this purpose each relationship/connection should be defined in terms of the trust of the user. In the second case, it’s important that data should also be under the control of the user even once it has been shared. In other words, a user should not lose control over access to their data just because it’s been passed on to someone else. This provides a number of technical challenges, and can arise when the trust level of a user is changed to a non-trusted level. In such a situation there should be some procedure to prevent them from further using the data, i.e. accessing the user’s new or already shared data.

Topology: A M4L stores memories that are collected from the social activities of a person, therefore connecting people or collecting information using an M4L in a network will be
influenced by social relationships. This gives rise to the requirement of network organization (or the topology of the network). The network should be organized in a way that allows users to retain their real life relationships. The network should allow a person to connect to people they like and should differentiate people that are closer to them than those that are not. The structure of a network should not impose rules and regulations that restrict a user from carrying out his social activities or result in bad performance of network. Moreover, when sharing data, the priorities of relationships should be considered by the network or system in order to allow a person to share data according to the priority of those relationships. Therefore, it becomes necessary to consider the relationship information first and then the network information, such as speed, bandwidth and so on.

**Searching:** Another challenge is to provide a means to find the correct data from the appropriate people. Searching in a network by an M4L depends on the operation carried out by it. For example, in a more specific scenario to complete an action such as generating a story about a person’s life, M4L systems will not need to collect every piece of data in the network. Instead, only the required data will be collected from people that are part of the story and are needed in order to complete it. As another example, to know about the social gatherings a person has had with other people, the intention of the system will not be to find all data named or recognized as “social gatherings” in a network. Instead only those that are related to the person will be searched. However, in some other cases the scenario may be more general and the system may need to find all data related to some event, location etc. such as all cultural events in Liverpool in 2008. To get the required data, M4L systems require searching data not only on file name but also on its metadata or even the contents of the file. This scenario makes searching data and the search query structure considerably more complex.

The complication of the above scenario of searching data in an M4L system can also be explained by considering the example of a person who wants to generate a story about their happy moments. For this purpose the happy moments from the life memory events of the person and friends and family members of this person will be considered. The required data will not contain all the happy moments of the lives of his friends and family but the happy moments they had with this person at a specific time, age, location, and so on.

**IV. SHARING HUMAN DIGITAL MEMORIES**

In this section we will consider different technologies and their suitability for sharing human digital events. The technologies will be judged based on the above described requirements for sharing HDEs.

**A. Web-based social networks**

Hundreds of Web-based social networks have been introduced each one grabbing some portion of people’s interest. The purpose of each of these sites ranges from child care [13] to aged people care [14] and from social activities [15] to research oriented activities [16]. We will delve into this further by looking at a number of social networking sites in more detail.

YouTube [17] is a Web-based social network site through which people share their videos; usually these videos are only of a short length. If a video exceeds a given length, it is then cut to the maximum length of a video clip provided by YouTube [18]. The site displays a list of videos related to the currently running video and also the videos uploaded by the same user. Orkut [19], launched by some Google employees in the United States in 2004 [20], was initially designed as a photo sharing online community where people could upload their pictures, join communities related to their interests and make friends. In its early days, getting membership was possible only through an invitation from an existing user, but since 2006 everyone has been entitled to sign up and upload their pictures, videos etc. ResearchGATE [16] is a Web-based social network for researchers and scientists, where they share their research material and other members comment on papers and give suggestions to authors.

The positive aspect of these websites is that they facilitate the exchange of ideas or data between people for a specific interest for which the website was developed. These Web-based social networks show the diverse interests of people that exist in the real world. But their existence in such large numbers disperses the interests of their users, which can result in a misinterpretation of the personality of the user and also creates a headache for the user in keeping track of the rules and regulations of each service provider. If the interests of a user change, then the new rules of a new service provider must be accepted, and potentially large amounts of data moved between sites. Other important issues with these websites are data privacy [21] and a single point of failure, which make them an unsuitable choice for M4L systems.

For individual users, it may often be most appropriate to store their M4L data in a single place, such as a desktop computer. This has a number of advantages; for example more coherent and meaningful information can be easily extracted from the data, allowing a user’s personality to be properly represented. However, one of the most appealing aspects of Web-based systems is their ability to facilitate the easy sharing of data between users. Data or extracted information should be shared in a way that maintains the privacy of the user and allows them to find people with shared interests, as and when required without worrying about new rules. We believe that Web-based social networks do not provide the important facilities either to organize a life’s worth of data or to share it in a way that is appropriate for memory for life systems.

Therefore, the limited facilities and uncertain conditions of Web-based social networks cannot guarantee to present every perspective of a member’s life. To overcome these problems a good alternative is to use the personal resources of a user, e.g. their mobile phone, desktop computer or laptop, which are under their control and bring no extra cost to the user. This opportunity can be supported using P2P networking technologies so that people can share their data, as much as they can afford, with friends, family and across the world.
B. Peer-to-peer networks

Data sharing plays an important role for an M4L because the intention of most people is not only to save their serendipitous moments for their own purposes, but also to share these events with others. P2P networks are suitable for sharing data of all kinds, sizes and any interest. Also, a pure P2P network has no single point of failure or a single authority to collect personal information, and this contrasts with Web-based social networks which create a potential threat to the privacy of a user.

P2P networks operate in the form of an overlay that sits above the network stack and avoids consideration of the underlying physical network details. Androutsellis-Theotokis et al. [22] classify peer-to-peer system into three categories: 

- communication and collaboration systems providing the infrastructure for communication and cooperation between peers;
- distributed computing systems taking advantage of available and free peer computer processors and content distribution infrastructure for sharing data among users.
- Furthermore, the content distribution technologies are grouped on the basis of services they provide as follows. Peer-to-Peer Applications provide for searching and transferring files without any fear for security etc. In this case publishing, storing and distributing data must be undertaken in a controlled and secure way. Peer-to-Peer Infrastructure provides a base and framework for carrying out the activities of peer-to-peer applications e.g. routing the information, anonymity and reputation management.

Many structures have been proposed by different authors, each having their own approach with differing merits and demerits. These approaches can broadly be categorized as being either centralized or decentralized. The centralized (e.g. Napster [23]) approach creates a single point of failure, consequently the trend has shifted towards decentralized networks. Decentralized networks are further divided into structured, unstructured and hybrid. The structured [24] approach uses a keying mechanism to allocate positions to nodes and data to these nodes based on their position or key value e.g. Chord [25]. These types of networks create a strict scenario in which file names or a specific attribute via which a file can be recognized plays an important role for storing or searching of files. But M4L systems require far broader the possibilities e.g. file name, metadata, contents of files etc. to allow searching and storage of files. In contrast to structured P2P networks, unstructured [26] approaches have no predefined structure or rules for the topology of the network and peers search for data within the network based on the information given by neighbours or a neighbour’s neighbours, and so on (e.g. Gnutella v4). In unstructured schemes searching generally takes longer and the chances of accessing a desired person, which is a requirement of M4L systems, are low because of the totally unorganized structure. Consequently hybrid approaches such as KaZaA [27] and Gnutella v6 [28], where some peers with high performance resources – called super peers – take the responsibility of controlling the network locally for a group of peers. This approach is considered to be more appropriate in terms of fast searching and low network maintenance. The problem with such networks is that each super peer becomes a single point of access locally for the network and whenever such a peer leaves the network it disconnects its cluster for some time until another super peer replaces it.

It is clear from the above discussion that peer-to-peer networks offer a suitable environment to share data with each other, but they lack the social aspects required to share data with or access the data of other people. Each person, in a social network, connects through some relationship to every other person such as by friendship, family member, job colleague or similar. The establishment of a connection has a social reason behind it which a conventional P2P network doesn’t consider. Alternatively they can be strangers but become friends by having common interests. Also, due to lack of knowledge about connections, a user in a P2P network usually has no control on sharing their data and anyone can access it without limitation. These relationships and the hierarchy of closeness that peers have with one another have a deep effect on most of the activities that occur within a network, such as data sharing. Therefore, we believe this problem must be overcome by application of social concepts in the P2P network.

C. Social P2P networks

Social P2P networks consider the social priorities in order to connect peers. The open nature of conventional P2P networks is controlled by the social network and activities, which make it suitable for M4L systems.

- Social concepts to improve P2P networks

The concept of peer-to-peer social networks first started through the deployment of various social concepts in P2P networks to improve their performance [29, 30]. Social concepts, in terms of online social networks, such as the keeping of a friends list, forwarding queries to known peers, making communities of peers with similar interests etc. improve the performance of the peer-to-peer network and help to find content in a network more easily. An important characteristic of social networks is used by Upadrashta et al. [31] in their work. They utilise the experience of a peer in a network. Peers analyse the queries that they receive from other peers and find and store their interests. In this way each peer stores information about other peers, resulting in the formation of virtual communities. Whenever a search query is received, it is analysed and then forwarded to those peers that have similar interests to those reflected in the search query. Anwar et al. [32] analyzed Orkut and, based on the social relations found among users, an information routing algorithm was implemented in a decentralized environment. Short paths were easily discovered by routing information only to peers with similar interests. This resulted in low network delay and reduced network traffic. Pouwelse et al. [33] designed a P2P system named Tribler and assumed social concepts in their model to improve the usability and performance of BitTorrent. The social concepts considered are friendship, trust and communities of similar interest. Instead of direct content discovery, the search is based on approaching the communities having similar interests. The five challenges: decentralization,
availability, integrity, providing proper incentive and network transparency, are addressed in their model.

In another approach proposed by Modarresi et al. [34], which is developed for social P2P networks and influenced by social communities, peers with similar interests are grouped together to form a community. This approach is similar to semantic overlay networks [35], the difference being that in semantic overlay networks peers having similar data connected to the same super peer. Data lookup is performed by sending queries only to those members that have similar interests. Interest-based communities bring peers with similar data or interests together and avoid peers that do not have the required data.

Soon after using social concepts for improving network performance, social P2P networks were introduced. Social models are not only influenced by the social ideas reflected in them but also the social activities of people that make them more secure, by interacting with people that they know and being cautious with those they don’t. This provides a social layer of security, which improves network security and makes the network more secure.

- **Social P2P models**

The social P2P models developed so far are limited in number and have not yet achieved the desired results because it remains at an early stage of research. Those that have been proposed are described below.

Chen et al. [36] describe the Maze system which is a centralized social P2P network introduced in China. Initially it was designed to improve the Tianwang search engine which has been famous in China since 1997. The system is designed around the idea of social relationships among peers. Social relationships are used to find peers in the network and then direct transfer of files can occur between peers. Peers share their friends lists and also the status of their friends. These friends lists automatically bring people with similar interests to a group where they can enjoy the resources of each other. For security purposes, a server issues tickets to each peer whenever it needs to request resources from other peers. The network can work for some time without the central server. However, this still represents a bottleneck for the network because of the great responsibility of the central server in the network in terms of facilitating peers finding information, issuing security tickets and so on.

PeerSoN [21] is an online social P2P network which emphasizes data privacy and security. These properties are achieved through a decentralized architecture and direct exchange of encrypted data between users. Social links are used to interconnect peers to achieve better performance. PeerSoN has been implemented using a structured P2P approach similar to Chord [25]. Structured approaches contain complications for M4L systems as stated in section B. Also, decentralizing network doesn’t achieve the required privacy according to which private information is intended for the eyes of specific audiences only.

MyNet [37] proposes a middleware solution for personal and social networking which organizes the personal resources of a user and shares them in their social network; this can be called a personal social P2P network (PSN).

D. **Requirements for a social P2P model for sharing HDEs**

From the above discussion, we believe that social P2P networks have significant potential for use in sharing human digital events. However, the process of designing a social P2P network is not straightforward and the following are the important challenges we need to overcome in order to develop such a system.

- The network should be decentralized to avoid any single point of failure.
- Personal information should not be accessible to anyone except those that are allowed access as assigned by the data sharer.
- The network should be organized in a way that provides real world social relationships to the user.
- A user should be able to search data using a broad selection of parameters as chosen by the user e.g. using metadata or the content of files. This kind of searching is not only helpful for users in a social P2P network but is also particularly important to implement the M4L systems for carrying out its own automated fine-grained searching in order to fulfill various tasks, such as suggesting interest groups, building up profiles, summarising memory threads and so on.
- Data sharing should be allowed only according to the choice of the user and there should be defined boundaries for each user or life memory system in terms of access to data by other users.
- Users should be provided with security measures to ensure that no one can use their data other than those that have been granted access.
- Each user should have an identity that is socially acceptable and recognizable by other peers.

V. **Conclusion**

The dramatic increase in the size of personal data, in the form of digital events, has made it difficult for people to properly organize their data. M4L systems facilitate this, taking the responsibilities of storage and annotation, and try to present a personality from the user’s data. Sharing data for M4L systems has become important due to the fact that people are increasingly capturing data to share within their social network. We described the challenges involved in sharing these digital events. These include: data privacy, users’ control of data, the topology of the network and the use of appropriate searching techniques. We also discussed Web-based online social networks and P2P networks for carrying out social activities. Web-based social networks are not a suitable choice for sharing human digital events due to the various issues discussed. P2P
networks have an open nature and peers have less control over their data. To control it, the communication of peers should be restricted using careful criteria. In the case of M4L systems, this communication is the social network of the user. We described the requirements for a social P2P model which are necessary for sharing HDEs, and which we intend to develop into a testable design in our future work.

REFERENCES


Ontology-Based Service Description and Discovery

Gilbert Cassar
Center for Communication Systems Research
University of Surrey, UK
Email: G.Cassar@surrey.ac.uk

Abstract—In pervasive environments users could move around with Personal Agents operating on their portable devices. As mobile devices become more powerful, users are also able to host services on their mobile devices. However the common technologies used to describe, publish, and discover services offer very limited freedom of expressiveness to the service designers. Service description frameworks such as WSDL provide descriptions of what a service is and what it can do; however with these kinds of descriptions machines still cannot easily interpret different concepts described through the technical attributes. If more expressive technologies are used to describe a service, machines will be able to identify service descriptions by concepts and data becomes machine-interpretable for reasoning purposes.

In this project we investigate the current technologies used for the Service Oriented Architecture (SOA) and discuss the limitations imposed by these technologies in terms of service description and discovery. We also look at other available technologies which are not used in the standard SOA but which could be embraced to provide a more versatile architecture for future services. A proposed solution for semantic descriptions of services and a discovery mechanism based on that is also described in the paper.

I. INTRODUCTION

Service discovery enables service clients to submit their request (i.e. goals) to a directory server or a discovery engines and obtain a list of matching services that can potentially help fulfilling their goals. The service clients could be system developers (i.e. human user) or software agents (i.e. machine user). The users specify their goal(s) to the discovery engine (usually in the form of a syntactic query) and the search engine looks through a service repository to find the services which are most appropriate to fulfill the clients’ goal. However, as we migrate towards more pervasive computing environments, it becomes clear that the current technologies applied for publishing services and discovering them are limiting the functionality that a discovery mechanism could offer. Most of the Service Oriented Architecture (SOA) solutions currently used on the Internet are centered around web services and usually do not include other kinds of services which can be found in pervasive systems. In pervasive environments users are mobile with Personal Agents operating on their portable devices. As mobile devices become more powerful, it is possible to host services on the mobile devices [14].

The search mechanisms offered by current discovery mechanisms rely heavily on matching keywords given by the user to the textual meta-data describing the services in the registry. This category of search and discovery mechanisms cannot interpret the concepts defined in the service descriptions and therefore cannot use logical reasoning to rank and/or recommend relevant services. In order to enable implementation of more complex search and discovery mechanisms and also adding enhanced functionalities such as ranking, recommendation, and service composition, a more sophisticated approach is required. The migration towards a semantically enabled platform aims to enrich the web with machine-processable descriptions. The main concept is to access data by content rather than by keywords. Using Semantic Web Services allows service description in machine-accessible way and helps to solve the interpretation of services using machine agents. “Semantic Web services aim at an integrated technology for the next generation of the Web by combining Semantic Web technologies and Web services, thereby turning the Internet from a information repository for human consumption into a world-wide system for distributed Web computing.” [2].

The enhanced service discovery mechanisms in our research are the building blocks for providing Service Personalisation. Rather than having a standard service or content presentation which is accessed by all users, we look at a service as a set of functionalities and content as a collection of data and we choose which functionalities and data are most suitable for the user. Each user is given an enhanced experience by being presented with a service and content tailor made for her/him as an individual. “Personalisation is about building customer loyalty by developing a meaningful one-to-one relationship with clients, by understanding the needs of each individual and helping satisfy a goal that efficiently and knowledgeably addresses each individual’s needs in a given context” [5].

The process of service personalisation starts at the service discovery stage when a user defines a goal and a discovery engine starts looking for the best service (or services) that can accomplish that goal. A service will always have a core functionality, however it can also define some side functionalities which are optional and adaptable for every user. Service personalisation exploits this property and assembles together the different optional functionalities of a service in order to customise the service for the user. Service person-
alisation deals with the functionality and the accessibility of a service rather than the presentation of the content once the service is accessed. In more complex scenarios, different services can be composed by a composition engine in order to present the user with an automatically composed service.

The main focus of this paper is description and discovery of the services for pervasive computing. We discuss the related technologies and methods and propose a design for service description and discovery mechanisms which are used for service personalisation in pervasive environments. In pervasive environments, mobility and continuity of services are also key issues in dealing with and also updating the services. However, to keep our discussion focused, in this paper we focus on description and discovery aspects of the services and the mobility and continuity issues are not discussed.

The paper is organised as follows. Section 2 describes the state-of-the-art technologies in service descriptions and service discovery mechanisms. Section 3 presents a proposed Service Description framework in order to support other kinds of services that can be present in a pervasive environment such as Mobile Services and Mobile Web Services. Section 4 describes a clustering-based service recommendation mechanism that is proposed to organise services into clusters based on related concepts thus facilitating concept-matching. Section 5 describes an Architecture to provide Service Personalisation in pervasive environments, and Section 6 concludes the paper and describes the future work.

II. STATE-OF-THE-ART

This section provides an overview of the service description and publishing in pervasive computing environments. The technologies taken into consideration include both the technologies that are commonly used and also some technologies that are still under research but which have a good potential for future use in pervasive systems.

A. Service Description

The future of mobile communications is envisioned to provide users with an Intelligent Personal Mobile Lifestyle Assistant which may take many forms and at the same time provide services which can be accessible by the smart devices of other users (assuming all privacy and security mechanisms are in place). In order to support this interaction between users and devices, a rich distributed service framework is required to describe the different concepts needed for personalization [3]. A detailed and machine-readable service description is the fundamental building block for providing an architecture in which advanced service discovery mechanisms can be performed [3]. Service descriptions are defined using a Service Description Language (SDL). Once it is properly defined, the service description becomes a template that service providers will use in order to publish and advertise the services they offer.

An SDL should be able to fully describe the functional and non-functional attributes of a service in a semantic, machine-interpretable form. A service description mechanism proposed in [11] distinguishes between Functional and Non-functional requirements as shown in Figure 1. Functional requirements include interface information such as input and output parameters, preconditions and effects. Non-functional requirements include performance characteristics (i.e. QoS) such as response time, accounting, and reliability [11].

In addition to service description, discovery mechanisms are also required to support finding and accessing the services based on different requirements. Universal Description, Discovery, and Integration (UDDI) is one of the most popular technologies adopted for service publishing and discovery over the Internet. UDDI defines a data model which is used to define services and save the descriptions as entities in the main registry. The data model is an XML schema for describing Businesses and Web Services and consists of five main data structures:

- **businessEntity**: Top level UDDI element containing the name, description, identifiers and classifications for the provider of a service.
- **businessService**: Abstract definitions which say what the service does. A businessEntity may contain one or more businessService definitions.
- **bindingTemplate**: Specifies where the businessService is located through a Universal Resource Locator (URL)
and defines how to invoke it. A businessService may have more than one bindingTemplate if it can be accessed using different protocols.

- **tModel**: Meta-data entries whose function is to describe the actual data. WSDL, XSD, or other objects being described by a tModel are stored in any Internet location identified by a URL in the tModel definition.

- **publisherAssertion**: A relationship structure putting into association two or more businessEntity structures according to a specific type of relationship.

The main problem with the UDDI data model is that it does not provide the means to describe in detail the capabilities of a service and the different processes a service is made of. "Service discovery with UDDI, only enables business discovery because the limited keyword-matching search is also based on a weak description of a particular service" [3].

One of the common standards to describe the technical aspects of a Web service is Web Service Description Language (WSDL). WSDL is an XML-based language for describing Web Services and how to access them. A WSDL document has a structure composed of five major elements. "WSDL separates services defined in abstract terms from the concrete data formats and protocols used for implementation, and defines bindings between the abstract description and its specification realization" [3].

The WSDL documents are used by service providers to describe their service and then make the service discoverable by publishing the WSDL document in the UDDI repository. When the service is discovered by a client, UDDI sends the description to the client in a SOAP envelope and the client uses the information contained in the WSDL document to locate and use the service.

The current WSDL standard operates at the syntactic level and lacks the semantic expressiveness needed to represent the requirements and capabilities of Web Services. Semantics can improve software reuse and discovery, significantly facilitate composition of Web services and enable integration of legacy applications as part of business process integration [6]. Formal semantic models exist outside WSDL documents and are referenced in WSDL through WSDL extensibility elements (e.g. through SAWSDL [7]).

The following describes some of the recent work to provide semantic description to services. Web Service Modelling Ontology (WSMO) is designed to support automated Service Discovery, Service Selection and Service Composition. Ontologies are used to provide a mechanism to describe and represent services making the data machine-interpretable. WSMO also introduces the concept of "Goals" which is a way of describing what the user wants to achieve. "In the WSMO approach to service discovery, a requester first chooses an appropriate goal from a set of pre-defined goals and further refines it according to her requirements" [12]. Mediators are then used to ensure interoperability in cases where there is an inconsistency in the data that is used for communication between two services.

The main limitation with WSMO is that it focuses on describing 'web services' rather than the concept of a 'service' which can be of any type. This makes it difficult to use WSMO to describe other types of services which can be provided in a pervasive computing environment; services which would not be classified as 'web services'.

OWL-S is another effort to create a machine-interpretable Service Description Framework. It holds both rich expressive power and well-defined semantics.

The parent concept in the OWL-S ontology is Service. Any instance declared in OWL-S needs to be an instance of the Service concept. Although this concept is created as a starting point for defining web services, it happens to be ideal for defining services available in a pervasive environment where instead of just web services there are other Mobile Services and Mobile Web Services which would still derive from the main Service concept.

The three major characteristics of a service that need to be specified by a detailed service description ontology are:

- What a service provides for prospective clients.
- How it is used.
- How a client interacts with it.

OWL-S describes these three major characteristics by using three top-level concepts, namely ServiceProfile, ServiceGrounding, and ServiceModel. ServiceProfile provides the information needed for an agent to discover a service. ServiceGrounding and ServiceModel provide enough information for an agent to make use of a service, once it is found.

The Profile of a service is meant to provide a concise but meaningful description of the service capabilities in order to advertise the service in a registry. However, once the service has been selected from the registry, the Profile is no more of use and the information contained in the ServiceModel is then used to interact with the service.

The ServiceGrounding describes how to access the service. In OWL-S, ServiceProfile and ServiceModel are abstract representations of a service and only the ServiceGrounding contains information about protocol and message formats, serialisation, transport, and addressing.

The concepts of Input, Output, Preconditions, and Effects are all well defined in OWL-S both in the ServiceProfile and in the ServiceModel. Thus OWL-S satisfies the main criteria required for describing services and their functional attributes.

Another effective aspect of OWL-S is that it still relies on already existing standards for service invocation and discovery. Service invocations are still carried out using WSDL definitions and OWL-S is designed so that it can extend
proposed in [11]. In this model, the services are characterised by five elements: Web Service, InputParameter, OutputParameter, Precondition, and Effect. These characteristics are stored in a database. The Web service characteristic model is then used to compute a semantic match between the concept used in the goalWeb services requirement description and the concept stored in the Web services characteristic database. The Functional Characteristics match and the Non-functional Characteristics match functions are carried out separately.

Another semantic modelling approach for Semantic Service Discovery is proposed in [7]. This approach classifies four semantic modelling aspects:

- Functional Semantics: Logical conditions that must hold true before and after the successful execution of a service.
- Data Semantics: Input/Output messages.
- Behavioural Semantics: Protocol that must be followed in order to use the service.

This approach identifies that semantics of inputs and outputs are not sufficient to properly match a service with a user’s goal. In order to have a more accurate discovery mechanism, the pre-conditions and post-conditions of a service must be also taken into consideration.

Recommender systems are a new approach towards providing users with the most appropriate content based on their personal preferences. There are different approaches used to provide Recommendation Systems. These approaches can be classified in two main classes: Content Filtering and Collaborative Filtering. A comprehensive study of Recommendation System strategies is presented in [9]. Four degrees of matching are identified in [9]: Exact match, Plug-in match (the output of the service subsumes the output of the request), Subsumes match (the output of the request subsumes the output of the service), and Fail.

Content Filtering methods build profiles on users and objects and subsequently use these profiles to associate users with matching objects. These methods generally require some external data from the users which might not be easily obtained.

Collaborative Filtering methods can be classified as Neighbourhood Methods and Latent Factor Models [9]. Neighbourhood methods establishes relationships between objects. This approach uses an item’s nearest neighbour that a user actually rated to predict what rating the user would give to this item. “The user oriented approach identifies like-minded users who can complement each other’s ratings.” [9]. Latent Factor Models classify the objects and users on a number of factors. The user’s rating for an object can be predicted using these known factors.
Collaborative Filtering methods give more accurate results than Content Filtering methods [9]; however Collaborative Filtering suffers from the Cold Start problem where it cannot process new objects or users because no previous information is available on them. Content Filtering does not suffer from the Cold Start problem because it will build a profile on new objects or users as soon as they are introduced in the system.

III. AN ENHANCED SERVICE DESCRIPTION AND DISCOVERY FRAMEWORK

OWL-S already provides a very detailed structure for describing a service, however in a pervasive environment where user terminals are also mobile and dynamic in nature, some further characteristics need to be defined in order to accommodate the publishing and discovery of Mobile Services.

In order to describe these characteristics, we extend the OWL-S Service description and create the concepts MobileServices and WebServices. These two concepts inherit all the characteristics of a Service but also specify extra characteristics that are required to fully describe a service that is provided by a mobile terminal. We also recognise the fact that some users may provide services from their terminals to users in the local network only while some others may provide the service for anyone on the Web.

Mobility brings into focus additional characteristics that need to be defined in the service description (i.e. Mobility characteristics) such as; location and availability (time). Thus new properties hasLocation, and hasAvailability are added to the ServiceProfile and the ServiceModel of MobileService as shown in Figure 2. The concept of location is defined by a co-ordinate which specifies where the service provider is. The concept of velocity gives the speed with which the service provider is moving since if it is traveling at too high speeds, the quality of service may decrease considerably. As other issues related to mobility such as updates, continuity, and session handling are not in the scope of this paper and are not discussed here.

The concept of availability describes the time window within which the service will be available. This characteristic is very important when it comes to mobile services since a user terminal has limited battery and resources which may very easily run out at some point making the service unreachable. Availability is described using ‘temporal things’ adapted from the OWL-Time ontology [1].

IV. CLUSTERING-BASED SERVICE RECOMMENDATION

Clustering has already been successfully used in Information Retrieval applications where the main focus is the automatic storage and retrieval of documents [8]. Clustering can organise data in groups (clusters) based on the similarity between one data entity and the other. Every data entry that needs to be clustered is usually represented in the form of a vector of measurements (Feature Vector) where each scalar component of the vector represents a particular attribute describing the data entry as shown in the following equation:

\[ x = (a_1, a_2, a_3, \ldots a_d) \]  

Where \( x \) is a Feature Vector, \( a_n \) is an attribute describing \( x \), and \( d \) is the dimensionality of the vector.

In this context, a Matrix Factorisation Model used to model users and shopping items for a recommender system is described in [3]. Items and users are represented by a vector of dimensionality \( f \) corresponding to the number of attributes that an item can be described with. The interaction between a user and an item is then measured by the dot product of the user and item vectors.

We propose a clustering based service discovery and recommendation using the attributes defined in the service description framework. The service descriptions can be expressed in Feature Vector form and thus enable the use of a pattern proximity measure in order to organise the services into clusters. A pattern proximity measure defines a distance function which is used to determine the dissimilarity between two feature vectors.

When the services are organised in clusters, the clusters will reflect what services are similar and related to each other through the concepts in their description. This will make it possible to discover services by the use of concepts rather than by matching the text in a user’s query with the text in the service description. The mechanism also makes
it possible to recommend services to the user since when a service is selected for the user, the services closest to it in vector space should be very closely related and thus valid recommendations to the user.

V. SERVICE PERSONALISATION ARCHITECTURE

The enhanced service description framework and the clustering-based service ranking mechanism described in the previous sections are designed as part of a Service Personalisation Architecture which supports mobile services and mobile Web services along with normal Web services. The architecture uses the ontology based service descriptions and machine-intelligence techniques to provide personalised ranking, recommendation, and composition of services. The architecture consists of the following components:

- Service Description Ontology
- Service Repository
- Service Personalisation and Composition Engine (SPACE)
- Personal Agents
- Content personalisation

We choose ontology-based descriptions to represent services and their attributes and thus employing a united Service Description framework. The Service Repository will then allow service providers to publish their services in the repository, as long as they abide by the rules of the Service Description Ontology. A clustering algorithm will keep the service entries organised in clusters in order to facilitate service discovery when the Service Personalisation and Composition Engine (SPACE) queries the Service Repository. Thus the Service Repository is an extended implementation of a UDDI with semantic search and discovery mechanism.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we provide an overview of the technologies available for describing and discovering services. The limitation with most of the current approaches is that they are all centered around web services and they do not support the integration of other services which can be found in pervasive environments. We propose an enhanced service description framework based on the OWL-S definition in order to accommodate services used in pervasive environments and a clustering-based service recommendation mechanism. Future work will focus on investigating discovery mechanisms based on our service description model and using advanced search functions such as service recommendation, service composition, and service personalisation.

ACKNOWLEDGMENT

This PhD project is supervised by Prof. Klaus Moessner and Dr. Payam Barnaghi at the Centre for Communication Systems Research, the University of Surrey.

REFERENCES

[1] Time ontology in owl, 'http://www.w3.org/tr/owl-time/ '.
A Proposed Matching Algorithm for the Direction and Relevance of Information in Knowledge Assets

J. Watthananon * and A. Mingkhwan**

* Faculty of Information Technology, King Mongkut’s University of Technology North Bangkok, Thailand
** Faculty of Industrial and Technology Management, King Mongkut’s University of Technology North Bangkok, Thailand
Email: watthananon@hotmail.com, anirach@ieee.org

Abstract - Knowledge Assets in knowledge management is part of the process used by information technology to assist in the data classification of organizations. Good knowledge assets are an effective way to create work successfully. Therefore, to have only data is not an important issue as in the past, so in this research, we applied the emphasis on connecting relationships of represent data, and described the archives of knowledge with the framework of standard DDC (Dewey Decimals Classification) to classify data and we propose a new efficient algorithm for the multiple matching knowledge. The experiment results, format emphasis of weight and quantity of relationships can be recommended by the direction of similar documents, this can be referred to in nature and inherited knowledge (i.e., parent and child nodes). Information in knowledge assets application using the matching algorithm is described to the core of knowledge in the fourth part of this paper.

Keyword: Knowledge Assets, Knowledge Direction, Knowledge Relevance, Multiple Relations, Relationships

I. INTRODUCTION

In recent times, many organizations focus on knowledge management to provide users with access and retrieval of information in a collected database or KA (Knowledge Assets). Because in the past, everyone would focus on quantity of data and thought if they had more data they would have a advantage over competitors. But the aspect of Knowledge Management is to have better quality data, also Knowledge Assets aims not to be too large in size [9], this means it does not collect everything in the database. KA: Knowledge Assets must be well designed, appropriately structured and divided into various topics for easy access. In addition, it must be able to identify and represent the content of knowledge that has relationship or degrees of relevance required to analyze and extract content hidden within the data. Research studies [6, 18, 10, 14 and 19] found that, they used minimal amounts of knowledge, which is especially hard for high-level users and specialized users because data in the database is basic and less useful for classification of knowledge. The problems of clear classification of information from users are different experience levels and basic understanding of the system.

Knowledge mapping is a process in knowledge management [18, 8] to define the direction of relationship and representation to link the hidden relationships within the information. In this paper, we focused on development method of calculating the weight of knowledge link relationships and describing the ability of relevant information to define the direction of relationship to link the hidden relationships within the information. Therefore, this method shows the conceptual overview that is easy to understand and shows the sub-issues of each relevant area. The classification not only represents and shows how to use the road map information but, also the key factor used in Knowledge Assets as "The standard information analysis and the technique applied of knowledge".

The rest of this paper is organized as follows. In the next section we provide a brief review of related works. Section 3 presents the method of calculating the weight of knowledge link relationships and describes ability of the relevant information to define the direction of relationship and representation to link the hidden relationships within the information. Section 4 presents experiments with some discussion on the results. In Section 5, we conclude the paper and put forward the directions of our future works.

II. RELATED WORKS

Many researchers have studied and design methods of presentation from the information retrieval format that allows users to access and easily retrieve similar information, such as in research Sriram et al. [16] presented how to find similar documents that aims to present visualization data from the Web, and used calculation techniques to find the relationship matrix similarity. Information provided includes name, URL, Keyword and elements of data. The results showed that, the format presentation should be smooth and easy to understand. Saad et al. [13] presented a concept maps-based approach for
knowledge visualization by using the Boolean value associated with centrally to support primary education. The results showed that the prototype can adjust the concept of users to present and support in-depth level understanding. Marshall et al. [7] proposed a method concept mapping system to store and compare the knowledge of organization, this research applied element anchoring mechanism similar to flooding (SF) algorithm to match nodes and substructures between pairs of simulated maps and student-drawn concept maps. The experimental results showed significant improvements over simple string matching algorithm which combined recall accuracy of 91% for conceptual nodes. Smolnik et al. [15] studied the comparison between model systems in Groupware-based organizational memories with the period storage of knowledge in organizations, they found that knowledge had relevant consistently and distributed accordingly to basic functions but, cannot support user's search queries. So, they a used technique to create knowledge structures by relationships with topic maps, the experimental results showed that topic associations can describe the relationships between topics by presenting topic keywords. Hlaoui et al. [4] presented a new graph-matching algorithm for the representation of text contents for similar documents by Dice coefficient method with new elements introduced by conceptual graphs. The experimental results showed that the system of information retrieval by a graph approach is especially good for short texts, it shows relationships between the elements from texts since its information retrieval works by comparison of at least one of the two elements. Hlaoui et al. [4] presented a new graph-matching algorithm for the inexact matching problem in K phases. The experimental results showed that content-based image retrieval was efficient correlating randomly generated graphs and can focus on a reduced set of model graphs.

From the researches above, the most effective technique to present data was calculation of the value similarities. In this paper, we also apply the DDC (Dewey Decimal Classification), base on classifying keywords and calculations to improve the classification model. Because, DDC is the standard library of hierarchical classification or family tree, it has a sufficiently wide topic for classification information and has an established relationship between class, sub-class and division by assigning notation to each call number [1]. However, we are adapting this measure to use as a concept hierarchy by the user, i.e. a direction of hierarchy, the weight of knowledge link relationships and to consider links of hidden relationships within the information. Recently, the Dewey Decimal Classification-Multiple Relations (DDC-MR) has introduced retrieval of unseen information for a set of documents [5, 18], for example, meaning and synonymy. The basic idea behind this approach is that documents are represented and linked to the hidden relevant data within the information. Each node is represented and directed to knowledge relating to the subject.

III. PROCESS TO DETERMINE THE WEIGHT RELATIONSHIP OF THE KNOWLEDGE LINK

Process to determine the weight relationship of the knowledge link is the method of calculating and analyzing proportion value of information, to quantify the weight of knowledge link which affects the direction and relevance of knowledge from the collection data. In this paper we focused on direction of knowledge link between the information and the direction between different groups of information, this can be described in detail as follows.

![Figure 1: Procedure the weight relationship of the knowledge link](image)

Step 1: Calculate proportion of relationships. From Figure 1 is a procedure of determining the weight relationship of the knowledge link, the first step, import data from research documents are divided into 4 parts: Title, Abstract, Keyword and Year published. Then data analysis with DDC-MR [1, 5] process to compute using the following equation:
Class Percentage:

\[ CP = \frac{\sum_{i=1}^{n} x_i \times 100}{\sum_{i=1}^{n} x_i} \]  

(1)

where \( x_i \) is value proportion of document relationship, as shown in Table 1.

Table 1: Proportional relationship of each document.

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>...</th>
<th>#6</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>...</td>
<td>0.0000</td>
</tr>
<tr>
<td>001</td>
<td>1.0031</td>
<td>0.6180</td>
<td>0.3799</td>
<td>0.8069</td>
<td>...</td>
<td>0.2633</td>
</tr>
<tr>
<td>002</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>...</td>
<td>0.0000</td>
</tr>
<tr>
<td>003</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>...</td>
<td>0.0000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>999</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>...</td>
<td>0.0527</td>
</tr>
</tbody>
</table>

Table 1 data shows proportional relationship between each of the class by DDC-MR [5] process that can classify all categories until 1000 classes. The value of the table proportion is 0 (zero), this is an arbitrary of relative zero which means relationship of the document still has relevance but is less relevant.

Step 2: Similarity computation. To compare and rank each document from values in Table 1, according to categories and sub-categories. The calculation has many methods but Lertmahakiat et al. (2009) presents the Pearson correlation coefficient – PCC as the best to predict similarities [6] so, we used this to compute the following equation:

Pearson correlation coefficient – PCC:

\[ r_{xy} = \frac{n \sum XY - \sum X \sum Y}{\sqrt{n \sum X^2 - (\sum X)^2} [n \sum Y^2 - (\sum Y)^2]} \]  

(2)

Where \( \sum X \) is sum of data measured by the variable 1 (X), \( \sum Y \) is sum of data measure by the variable 2 (Y), \( \sum XY \) is sum of multiple data between variables 1 and 2, \( \sum X^2 \) is sum square data measured by the variable 1, \( \sum Y^2 \) is sum square data measured by the variable 2, and \( N \) is size of sample. After that, we use value from PCC to compute merge document, ranking class to sub-class and division, if the document has the same class it would calculate the number of classes and calculate degree of relationship between classes, as shown in Table 2.

Table 2: Table relevance metrics

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>...</th>
<th>#6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>147</td>
<td>11.7788</td>
<td>236</td>
<td>11.1373</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>147</td>
<td>9.5207</td>
<td>0</td>
<td>177</td>
<td>8.3530</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>236</td>
<td>15.2850</td>
<td>177</td>
<td>14.1827</td>
<td>0</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>223</td>
<td>14.4430</td>
<td>166</td>
<td>13.3013</td>
<td>397</td>
<td>18.7353</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table 2 shows the Number of class relevance and Quantity of relationships in each document, this can be described as #1 number of class relevant to #2 in class 147, #3 in class 236, #4 in class 223 and #5 in class 183 and quantity between relationships as 9.53%, 15.28%, 14.44% and 11.85%, while document #2 related to #1 in class 147, #3 in class 177, #4 in class 166 and #5 in class 149.

Step 3: Calculate the weight of knowledge link. This represents the size of knowledge link relationships in each node, as computed in previous methods. That proportion cannot be used to show the weight of knowledge link because it shows only data. So, to show the relationship between 3 nodes: For example with document A had relations class 100 is Philosophy, class 300 is Social sciences and node class 600 is Technology, had proportion relationship is 8%, 30% and 10% order class, as shown in Figure 2-(a).

Figure 2: Preview the knowledge link relationship between 3 nodes.
Figure 2-(b) shows range of relation between documents A with 3 classes, this can describe the different relationship but we must calculate the new weight of knowledge link to provided meaning and appropriate proportions, which can be computed by using the following equation:

**Weight value of the knowledge link:**

\[
\text{Weight} = \frac{\text{value of proportion}}{\text{total value of proportion}}
\]

where value of proportion of each knowledge link is the sum of proportion of the destination such as: document A → 100 = 8%, document A → 300 = 10% and document A → 600 = 30%, the proportion displayed is in the range of proportion relationship, users can set conditions and total value of proportion of raw proportion as 8 + 10 + 30 = 48.

\[
W_1 = \frac{(8 * 100)}{48} = 16.667
\]

\[
W_2 = \frac{(10 * 100)}{48} = 20.833
\]

\[
W_3 = \frac{(30 * 100)}{48} = 62.5
\]

Step 4: **Compare the size of knowledge link with criteria.** To compute the appropriate size of knowledge link based on weight values calculated from 4.3 compares proportion with every knowledge link in the displayed data to show the meaning and description of the quantity of relationship. So, the criteria used in this paper is 0.1 mm per 1 proportion, this means if the knowledge link has a proportion of 40 when compared with criteria, size of knowledge link is 4 mm. In this method, the size of relation knowledge link has to change as shown in Figure 2-b, size of knowledge link in document A → 100 = 1.67 pt, document A → 300 = 2.08 pt and document A → 600 = 6.25 pt.

Step 5: **Direction of the relationship.** Generally relationships between variables are statistical methods to create a scatter plot which analyzes direction and proposes the relationship [12]. In this paper, the relationships are arising links between node and thickness or thinness part of related knowledge link. This consists of the number described as quantity of relationship on the knowledge link and can be applied to link similar information. In Figure 2-b we describe direction of relationship between document A and 3 knowledge’s as: Technology, Social sciences and Philosophy, end of the arrow will point to the controlled knowledge. This shows the maximal direction relevant to Technology (6.25pt), while the minimal direction relevance was Philosophy (1.67 pt).

Last step: **Quantity of relationships.** To described the knowledge of one node associated with another node, and presenting quantity of the center position knowledge link relationships. Defined as a number proportion such as 18% means that the relationship between document and nodes, has values equal to 18, which if quantitative is not configured or cannot be shown, this means the quantitative relationships are uncertainty.

In Figure 2-(d) and 2-(e) shows relation weight of knowledge link between 3 classes: For example the DDC standard can classify by keyword, so if we want to know the relation between 3 nodes in document A, we must used equation in 2-(e) to computed the same of keyword between classes. After that, calculate the weight of knowledge link in step 3, 4, 5 and 6 to represent the weight in class 100 → 300 = 5.55 pt, class 100 → 600 = 3.33 pt and class 300 → 600 = 1.11 pt and defined as a number proportion on knowledge link, as shown in Figure 2-(e).

**IV. THE PROPOSE MATCHING ALGORITHM**

In this section, we present a new algorithm for matching knowledge. The goal is to find the core of knowledge assets mapping between their nodes that leads to the most important class. The matching knowledge between the reach of class is a function of the similarity between each classes or nodes. It can be viewed as the distance between the knowledge’s [19]. The basic idea of the new algorithm is iterative of the best possible class mapping and creates direction of the best knowledge at each iteration phase by considering both the weight of knowledge link with criteria and rank of quantity of relationship. The advantage of this algorithm is that this iterative process often allows finding the optimal core of knowledge mapping within information by classify data into 1000 classes. In the first phase, the algorithm selects the third possible rankings from 1000 ratings that minimize the multiple induced by node create knowledge link matching only. In the second phase, the algorithm examines the knowledge links that contain percentage between relationship of document and knowledge and then again computes that weight of knowledge link mappings that class to show quantity of relationships. After that, we receive that similarity measures associated with the basic relations have been defined; i.e. document number 1 have already been relation with 1000 classes of nodes and edges, selection of nodes and edges, etc. We used this to compute the following algorithm:

**Algorithm and Complexity:**

**Input:** 1000 attributed classes of each documents

**Output:** matching between classes or nodes in documents from the knowledge assets

1. Initialize K as follows: For each \( K_i = d(f_i, (v_i), (v_j)) \).
2. Initialize R as follows: For each \( R_{ij} \), set \( R_{ij} = 0 \).
3. While Relation_Phase < N
   - If Relation_Phase = 1, Then
     - For \( i = 1, \ldots, n \)
       - Set the value 1 to elements of R corresponding to the most value in i row of K.
     - Call Matching_Classes(R).
   - Else If Relation_Phase = 2, Then
     - For \( i = 2, \ldots, n \)
       - Set the value 1 to elements of R corresponding to the second rank value in i row of K.
     - Call Matching_Classes(R).
   - Else If Relation_Phase = 3, Then
     - For \( i = 3, \ldots, n \)
       - Set the value 1 to elements of R corresponding to the third rank value in i row of K.
     - Call Matching_Classes(R).
   - Else For all \( i = 1, \ldots, n \)
Set \( R' = R \)
For all \( j = 1, \ldots, m \) set \( R_{ij} = 0 \)
Select the element with the most value in \( K \) that is not calculated 1 in \( R' \) and set it to plus 1 in \( R \) and \( R' \);
Call Matching\_Classes\( (R) \);
Set \( R = R' \).
If all the elements in \( R \) are calculated 1, Then
Set Matching\_Classes = \( N \)
Else plus 1 to Matching\_Classes.

**Matching\_Classes\( (R) \):**
For each valid mapping in \( R \)
1. Compute the matching relation induced by nodes.
2. Compute the weight of knowledge link.
3. Create the direction between document and nodes.
4. Show the percentage relation on each knowledge link that means Quantity of relationships.

**Description with the algorithm:**

\( K = (K_{ij}) \) is Knowledge’s, where \( n \) is the numbers of classes or nodes. Each element \( K_{ij} \) in \( K \) demotes the relation between node \( i \) in \( K_1 \) and node \( j \) in \( K_2 \) and \( R = (R_{ij}) \) is rank. The first step is to initialize \( K_{ij} = d(\mu_1(v_i), \mu_2(v_j)) \)[19]. The second step consists of initializing \( R_{ij} = 0 \). The third step contains \( N \) phases. In the Relation\_Phase = 1, the elements of \( R \) corresponding to the most value in each row of relevance matrix. In the Relation\_Phase = 2, the elements of \( R \) corresponding to the second rank value in each row of relevance matrix and in the Relation\_Phase = 3, the elements of \( R \) corresponding to the third rank value in each row of relevance matrix.

**V. EXPERIMENTS AND DISCUSSION.**

**5.1 Data used in the experiment**

In our experiments, we use collection knowledge from academic articles of the participants in a national conference disciplines in the computer and information technology amounting to 620 articles, which were published during 2005 to 2009. In this paper, we provided one document for research, so the dataset used 4 sections: Title, Abstract, Keyword and Year published, this pattern can classify documents in a short period of time.

**5.2 Discussions**

We used DDC-MR process provided by the relationship analysis to run our experiments. DDC-MR is a hierarchy process of classify content based on DDC in level 3 (1000 classes). Thus, we used this to classify unseen data in knowledge, calculate proportion of relationships in each class, calculate the weight of knowledge link and display size of knowledge link referring to the direction and relevance of information in knowledge assets. However, the research aims to develop the method for calculating the weight of knowledge link relationships and describe ability of the relevant information to define the direction of relationship and representation link the hidden relationships within the information. This can show us the direction relation of class and what proportion is linked to another knowledge or different content, as shown in Figure 3.

**Figure 3:** Shown direction and proportion relevance of each document.

Figure 3 shows an example to link each document amounting to 9 articles that assigns documents represented by a square symbol and class represented by planet symbol. The aim is to link the size of knowledge link which was computed to the defined details of the proportion relationships, then represent it simply to help everyone.
understand the data. Each document shows the direction and proportion relevance by 3 sequences because we cannot displays any class, as it would not be able to read the relationship. Shown in Table 3, descriptions of document #1 is associated with class 620 = 4.81%, class 621 = 4.12% and class 005 = 3.41%, while the document #2 is associated with class 620 = 4.37%, class 658 = 3.61% and class 621 = 3.61%. The weight of knowledge link used to display the direction of relation had different size depend on the computed value of proportion.

<table>
<thead>
<tr>
<th>Name</th>
<th>#01</th>
<th>#02</th>
<th>#03</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank1</td>
<td>'620'</td>
<td>4.824</td>
<td>'620'</td>
<td>4.372</td>
</tr>
<tr>
<td>Rank2</td>
<td>'621'</td>
<td>4.118</td>
<td>'658'</td>
<td>3.607</td>
</tr>
<tr>
<td>Rank3</td>
<td>'005'</td>
<td>3.412</td>
<td>'621'</td>
<td>3.607</td>
</tr>
<tr>
<td>Rank4</td>
<td>'658'</td>
<td>2.941</td>
<td>'005'</td>
<td>3.388</td>
</tr>
<tr>
<td>Rank5</td>
<td>'515'</td>
<td>2.824</td>
<td>'676'</td>
<td>3.169</td>
</tr>
</tbody>
</table>

The experimental results show the direction and quantity of relationship between documents, which is relevant in Computer Programming (005) and the second, is class General Management (658) by analyzing the amount of links and the weight of knowledge link relationships. However, we can refer to the relevant documents in-depth when searching for content such as document #3 and #4 which have relevance to class General Management (658) and class Labor economics (331) according to the computed value of proportion. The document #4 can be referred by document #8 in class Computer Programming (005). Thus, if we only focused on direction, the displayed data would not go to the core of knowledge or the vice of knowledge.

VI. CONCLUSIONS AND FUTURE WORKS

The direction and relevance of information are applied to link knowledge that is extracted from Knowledge Assets in a manner known as Knowledge mapping. This is supported and represent by contributing information from 620 articles. This method can represent the conceptual map of relation and open dimensional thinking of the total system by linking the correct path. In addition, when considering the benefits, quantity of knowledge or the thickness weight of the knowledge link can correlate to continuous data. We can analyze all relevant data and display it, so the problem is not the searching of knowledge but the issue understands the knowledge clearly.

In our future work, we plan to develop and apply our proposed method with a variety of information retrieval such as star relationship or triangle relationship etc.

REFERENCES

Supporting Elastic Cloud Computation with Intention Description Language

T. Baker, A. Hussien, M. Randles, A. Taleb-Bendiab,
School of Computing and Mathematical Sciences
Liverpool John Moores University
Byrom Street, Liverpool, L3 3AF, UK

Abstract - The continuing shift towards cloud computing is a major influence in today’s industry. The proliferation and availability of cloud computing facilities, including on-demand platforms, on-premises platforms and Platform as a Service (SaaS), is a key part of that shift. Theoretically, therefore, it ought to be straightforward to create a new elastic Cloud-Based Application (CBA) because of the tools and techniques within the available platforms. Moreover, at design time, an application can be enhanced by accessing/composing services provided by the cloud rather than developing entirely new services. Practically and programmatically speaking, however, there is still huge gap between the users and the cloud, as the clouds’ users are still not able to dynamically update their applications’ components and services at runtime in an effective manner. Attaching a service into a running CBA to enhance the application behaviour requires some knowledge of at least the service and the application source code so that code composition can be take place. This paper demonstrates a new Intention Description Language (IDL) that can be used via Provision, Assurance, and Accounting (PAA) modelling approach, to weave a new service to the elastic CBA at runtime. The IDL aims at bridging (how, why and what) is done by representing the business processes managed by business rules. While the PAA description is out of the scope of this paper, a full explanation of IDL is presented here.

I. Introduction

The continuing move to cloud computing platforms represents a major upheaval in the computer industry. These kinds of platforms are becoming widely used in dynamic economic environments, where a company’s survival depends on its applications’ status and the applications admin’s ability to focus on core business needs and adapt the application quickly at work time by adding, removing and modifying any of the web services or components based on the BPEL and WSDL [1, 2].

However, the Web Service Description Language (WSDL) [3, 7] was developed to describe the technical details, (partners, partner link types and port types), to show how a web service can be accessed and invoked remotely over the web to enhance or expand systems functionalities. In the same regard, state of the art business process automation systems are based on Business Process Execution Language (BPEL) [4] as services composition language, which relies on WSDL for describing and automating business behaviour. Since the WSDL was made to be used by computers, it describes only how something can be done technically [5], e.g. how a web service can be invoked.

From the elasticity and scalability perspectives, the dynamic Cloud-Based Application (CBA) has different requirements from the ones fulfilled with WSDL and BPEL. While the technical description of services is important in the cloud computing [6, 7], describing why and what is done are also important.

In this paper, we proposed a new Intention Description Language (IDL) that can be used via Provision, Assurance, and Accounting (PAA) modelling approach, to weave the CBA at runtime. The IDL therefore aims at bridging how, why and what is done by representing the business processes managed by business rules. While the PAA description is out of the scope of this paper and can be found in [8, 9, 10], a full explanation of IDL is presented here.

The remainder of this paper is structured in five main sections. Section 2 gives more details about the problem statement this paper intended to solve. The proposed solution is given in section 3 with a precise description about the IDL meta-model. A case study, PAAPetShop, is available in section 4 to aid a clearer understanding of the proposed approach from a practical point of view. However, to, evaluate this approach at runtime, it is necessary to demonstrate its practicality and reliability, this will be explained in section 5. The paper ends with some conclusions and future work.

II. Problem Statement

The main characteristic of Elastic Compute Cloud (EC2), which has promoted its use amongst many of the largest businesses for most, if not all, of their business applications, is the elasticity of this computing paradigm [11]. Elasticity means the CBAs, especially long running applications, should be flexible and adaptable to accept new changes and requirements from different parties at runtime in an
automated manner. The requirements might need to invoke and inject services and components from other platforms using different description languages (e.g. WSDL, BPEL…etc) to achieve the desired behaviour.

WSDL is designed to provide the web services technical details that are used and needed by BPEL to invoke and execute the web services. While the technical details alone are not enough to achieve the full cloud elasticity, the BPEL itself is too rigid to cope with changing business demands instantly [5], without the need for taking the system offline to modify the business demands.

However, the elastic cloud computing applications need formal/standard description language support to allow these applications to be dynamically updated by integrating services and components from other languages and frameworks at runtime.

III. PROPOSED SOLUTION

To overcome the above problem, we propose a more pragmatic weaving approach that is mainly composed of two parts: A CBAs IDL, that sits in the cloud and is accessible and easily adapted, by cloud users, at runtime, and the PAA framework (SaaS) that also sits in the cloud that ought to efficiently read and execute IDL and the emergent requirements expressed though the using of IDL. While the PAA framework has been widely explained in [8, 9, 10], this paper focuses on detailing the IDL.

A. Intention Description Language (IDL)

Tangible products usually have a well-defined set of possible variants for customization. For example, if a customer requires a faster PC, more powerful CPU and increased RAM can be designed, built and attached to the computer motherboard.

However, the same cannot be straightforwardly achieved for intangible cloud applications/services/components. This makes the need for a comprehensive description language one of the most important undertakings for the elastic cloud computing. Thus, the cloud elasticity requires combining and correlating business, operational and IT aspects into CBA IDL.

B. IDL Approaches

IDL is an xml CBA description language. It uses three main approaches to model and automate business behaviour on cloud applications.

The first one focuses on flows of activities that generate values. This part of IDL is called the flow model. The flow model is used to describe “what” is happening; representing this in the xml using the Flow attribute, which is encapsulated within each task to direct the execution to the next task. The flow, as shown in Listing 1 has two main subclasses move to and the decision. Where <moveto> directs the IDL to the next task, whereas <Decision> is used to direct the IDL to a sudden task that arises according to the user’s emergent requirements. The two types are shown in Listing 1.

```
Listing 1: IDL flow model
--------------------------------------------------------------
<PetShopIntention>
  <process>
    <startpoint id="StartPoint1">
      <moveto>Checkout</moveto>
    </startpoint>
    <action id="Checkout">
      <input type="text">
        <message><![CDATA[checkout.aspx]]></message>
      </input>
      <moveto result="[Not set]">ValidateAccount</moveto>
    </action>
    .......
    <Decision id="Valid" NeptuneFunction="process">
      <input type="text">
        <validation><![CDATA[1]]></validation>
        <message><![CDATA[0]]></message>
      </input>
      <moveto result="1">Billing</moveto>
      <moveto result="0">EndPoint1</moveto>
    </Decision>
    .......
    <endpoint id="EndPoint1" />
  </process>
</PetShopIntention>
--------------------------------------------------------------
```

The second approach uses rules/requirements to describe desired business behaviour. This so called Business Rules/Requirements (BR) approach uses rules/requirements to describe “why” something has to happen according to the rules and the requirements, and uses technologies to automate decision logic. The NeptuneScript is used in the IDL to describe and create executable functions that describe why something is done according to the requirements.

```
Listing 2: IDL BR
--------------------------------------------------------------
<neptunefunction>
  define OrderShipping as ProcessStep
  {
    requires
      {
        saveOrder : SaveOrderShipping
        order : PetOrderDetails
      }
    features
      {
        feature shipOrder for order; }
    actuation
      {
        order.ShippingDate = TODAY;
        onceValid : saveOrder(order);
      }
  }
</neptunefunction>
--------------------------------------------------------------
```

The actuation keyword shown above helps in understanding why something is happening.
The third approach is the technical specification of the services and components that should be used to accomplish certain behaviour. The technical specification of the service is, for example, the code it uses; the name of the service, the type of the service, the url address of the service, the execution engine…etc. which is shown in the following, Listing 3.

Listing 3: IDL technical specification

```
<complete id="2" code="# WSDL" execution_engine="#BPEL">
  <nblo name="_default" type="executable" service="#sellPets" />
  <nblo name="_default" type="executable" service="#buyPets" />
</complete>
```

In this case, the IDL can be described as a comprehensive description language due to its nature of integrating the three different approaches (what, how and why).

Thus, the IDL schema defines three core types of information that provide the descriptions that a consumer can use to discover, select, invoke services and have a view on services’ behaviour at execution time.

C. Formalizing IDL Meta-model

In order to establish a proper base for IDL to be used in the design and creation of CBAs, we provide a formal specification for it. This serves for purposes such as communication and implementation for integration with other specifications and languages like WSDL and BPEL.

As explained in the previous sections, the IDL consists mainly of a set of processes that describe the business behaviours. Some of these processes are composed of sub processes which both include a task or set of tasks that should be executed to achieve the desired behaviour of that process. The process model information is attached to the IDL to provide full information about the current IDL, for example, the IDL creator, IDL name, IDL owner, and the guide key.

As shown in Figure 1, there are different kinds of requirements that are defined in the IDL to give a clear view to the user and to satisfy the three different approaches explained above. The Provision requirements show the specific user requirements and the individual services and components the user wants to use in the application at runtime. However, the provision requirements can either be added manually by the user or extracted automatically by the provision generator at runtime during the application execution.

The Assurance requirements describe properties that are fundamental for the characterization of a service. We rely on a set of non-functional properties such as validity, availability, service type…etc. In order to provide a suitable language that can be understood by business stakeholders and consumers, this requirement has been extended to include Concept Aided-Situation Prediction Action (CA-SPA) policy, which can be used to add new prediction and action at runtime according to the current situation; more information about CA-SPA can be found in [12].

The Accounting requirements represent the statistical side of the application and supply the dashboard with the necessary information to allow better managing of the application at runtime.

The following in Listing 4 shows the IDL Provision Assurance and Accounting requirements which have been added at runtime to the PetShop IDL, described in the next section, to add a new process (check delivery) and validate the address of the shipped order.
Listing 4: PAA requirements

```plaintext
define ValidPreShippingOrderPAA as PAA {
    provisioning
    { order as PetOrderDetails }
    assurance
    { order.ShippingAddress is VALID
      order.SelectedPet is INSTOCK
      order.ShippingDate is NULL }
    Accounting
    { /*error handling or logging defined here*/ }
}
```

---

IV. CASE STUDY: PETSHOP IDL

PetShop is an architectural blueprint developed by Microsoft based on the original Sun Microsystems PetStore benchmark enterprise architecture for e-commerce and enterprise systems [13]. The application produced by the PetShop blueprint builds a web-based application for browsing and purchasing pets.

In particular, the PetShop application is implemented from Web Services composition that perform the main processes outlined in Figure 2, which for instance enable administrators of the system to add, remove, and modify the animals available for sale, with their data stored in a database.

![Figure 2: The original PetShop Order Process]

To this end, this section outlines the adaptation processes when a new process or service can be injected (or simply added) at runtime to the standard PetShop process model.

The standard PetShop blueprint does not support EC2, as it is a static design approach. In other words, the process model cannot be adapted at runtime (e.g. add a new service to the PetShop process model like PetRefund). The whole system would need to be taken offline before the adaptation, and then re-uploading it again online after the adaption. Of course, the system administrator should undertake this. So the user will not be able to add any new function or service to the original process model on the cloud.

Hence, there is a new way for designing the PetShop via PAA, using the PetShop cloud IDL which can be accessed and modified by the user via the PAA editor service; which allows the user to modify the entire process model and upload the modified version at runtime. The PAA PetShop user interface is shown in Figure 3.

![Figure 3: PAA PetShop-based-Cloud]

The user will be mainly dealing with the above interface, s/he can move from this page to another PetShop page using the available buttons. The movement from this page to the next one is managed and controlled by IDL flow model. The IDL might be saved on another virtual machine on the cloud, and can be accessed and used via the intermediate model, shown in Listing 5.

Thus, updating Figure 2, the PetShop process model, can be completed using the Edit button, which displays another web page for the PAA Wiki Editor for IDL, as shown in the Figure 4.

![Figure 4: IDL runtime editor]

More information about the PAA wiki Editor and how to update the process model can be found in [14].
A. PetShop CBA and IDL interconnection

The PAA PetShop itself can be saved on another Virtual Machine (VM) on the cloud. All the information needed by PAA PetShop to access and use the IDL is saved in another intermediate model, which is part of the PAA PetShop. This intermediate model saves all the IDL information that is shown in the IDL meta model (the IDL creator, IDL name, IDL owner, and the guide key). This information is used to differentiate the requested IDL from the other IDL models on the cloud.

Listing 5: IDL intermediate model

```xml
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfIDLModelStub
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <IDLModelStub>
        <Name>PetShop</Name>
        <UID>2a6a736c-62c8-4a4a-b82e-7f11d2905c08</UID>
        <IDLModelFileName>c:\inetpub\wwwroot\neptune\DecisionModel\PetShop.xml</IDLModelFileName>
        <Organisation>Liverpool JMU</Organisation>
        <Creator>Thar Baker</Creator>
        <DateAdded>2009-05-07T13:56:17.8118415+00:00</DateAdded>
        <DateAccessed>2009-05-07T13:56:17.8118415+00:00</DateAccessed>
    </IDLModelStub>
    <IDLModelStub>
        <Name>Simple Triage</Name>
        <UID>f14821fe-a131-4a5f-b3c5-0b3a7b4fedf3</UID>
        <IDLModelFileName>c:\inetpub\wwwroot\neptune\DecisionModel\swellingsimple.xml</IDLModelFileName>
        <Organisation>LJMU</Organisation>
        <Creator>Thar Baker</Creator>
        <DateAdded>2009-05-07T09:44:45.9107592+00:00</DateAdded>
        <DateAccessed>2009-05-07T09:44:45.9107592+00:00</DateAccessed>
    </IDLModelStub>
    <IDLModelStub>
        <Name>Simple Triage Extended</Name>
        <UID>d537d371-ae1e-4b09-9c7a-d48bae851282</UID>
        <IDLModelFileName>c:\inetpub\wwwroot\neptune\DecisionModel\extended.xml</IDLModelFileName>
        <Organisation>LJMU</Organisation>
        <Creator>Thar Baker</Creator>
        <DateAdded>2009-05-07T09:51:06.4708236+00:00</DateAdded>
        <DateAccessed>2009-05-07T09:51:06.4708236+00:00</DateAccessed>
    </IDLModelStub>
</ArrayOfIDLModelStub>
```

Hence, the interconnection between the PAA PetShop and the IDL is established; whether they are both saved on the same cloud, or another public/private cloud, can be determined through the intermediate model. The following Figure 5 shows a high level overview of the proposed elastic CBAs via the use of IDL.

As shown in Listing 5, the intermediate model is an xml-based model. The agility behind this model is it provides important data in the form of adaptable xml tags. These tags can be adapted at runtime, by the user, using PAA Wiki Editor [8, 14]. Hence, accordingly, the IDL model and the other information can be adapted at runtime depending on the location of the requested IDL. On the other hand, this can provide insights to the user or the system admin to track user history by considering the date that the IDL has been used and the time of the last access into it.

V. EVALUATION: SYSTEM SCALABILITY

A test was conducted to ascertain the scalability of CBA PAA PetShop during its operation in comparison to the original Microsoft PetShop. By introducing a new behaviour to PAA PetShop via IDL, the performance impact of the further interpretation of the new behaviour could be contrasted against that of the performance impact of the introduction of the same behaviour in the Microsoft PetShop model.

Figure 6 shows two systems executed to produce the same behaviours over a time-scale of cycles of a process execution. After 50,000 executions, a new task requirement is introduced in the IDL model, and a task description updated accordingly. As new semantic linking needs to take place, performance is reduced, such that the time to complete the process increases. It should be noted, however, that as only one new requirement is needed to be linked, the performance impact is less than that of the original initiation at cycle 0, where many new requirements are introduced at the same time. After the reconfiguration, linking and execution, performance returns to a new standard, slightly slower than the original behaviour from 10,000 to 50,000. This is due to the added time needed to execute the action by Neptune.
On the other hand, there is a slight increase in execution time of the original PetShop (given in green) from 50,000, due to the new behaviour introduced to the PetShop code to be executed. It can be noted that after the execution of the new behaviour, however, the execution performance of PAA PetShop compared to Microsoft PetShop is largely similar, thus scalability issues can be said to be introduced.

![Figure 6: Scalability testing of PAA PetShop vs. Microsoft PetShop](image)

**VI. CONCLUSIONS AND FUTURE WORKS**

This paper has argued for the design and use of an adaptable intention description language for cloud-based applications to achieve the cloud elasticity characteristics through the using of the PAA approach. This means it will be straightforward, quick and reliable to amend the cloud-based applications at run time without the need for the recoding and republishing of the application.

The perceived value of using IDL via PAA Neptune has driven the re-development of PetShop to be a Cloud-Based Application through this approach.

In future work, the IDL description language is to be rigorously tested and evaluated to ensure the design of IDL and PAA tools is correctly envisaged. But there is still a large body of outstanding research issues, which have to be resolved before the full benefits of exploitation can be realised.

**REFERENCES**


Abstract—In today's world web content play an important role of feeding information in to our society. Unlike books that are systematically classification using a standard like DDC or LC system. Web is classified by owner of the site which means no any standard between one another. In our work we propose the idea and a preliminary results of classifying the article of the Online Encyclopedia content like WIKIPEDIA in to a DDC library standard system with the extension of the MR (Multiple Relation) which allow user to find the ontology like topic and minor classed of the article. With the standard and extension befits of DDC-MR user of Wikipedia will be able to related their researches topic with the standard classification of material in the library.

Keywords – Dewey Decimals Classification, Multiple relations, Information retrieval, WIKIPEDIA

I. INTRODUCTION

Today’s society is a kind of Information Society. Therefore, everyone needs all news and information from different sources in order to develop their own livelihood, make a living including level up the country. Internet is one of the important tools, which present information correctly, quickly, efficiently and collect all information. However, Internet service is classified without any standards support and specified into the group based on only one group as shown in Fig. 1.


Wikipedia classification group is without any international standards support and specify into the group based on only one group as shown in Fig. 1.

The popular classifying information systems used in the library are Library of Congress Classification (LCC) of DR has divided into 20 main classes which use an alphabetic is a symbol. Herbert Putnum, Dewey Decimal Classification (DDC) by Melvil Dewey has 10 main classes and use an alphabetic is a symbol [1], [3].

In the present, the classifying of articles or books is categorized only into a group which is proposed in the previous research as Dewey Decimal Classification Multiple Relations (DDC-MR) [2], [3]. So, that it can recognize the contents of other groups hidden in the articles or books. In this paper, the researcher is using the Dewey Decimal Classification Multiple Relations (DDC-MR) for analyzing the contents of the articles in Wikipedia. In order to, synchronize the search of the contents in the Wikipedia system and the articles in the library with better format and speed. As a result, this connection of the content of the article in Wikipedia is international instead of search by Keyword. In addition, this connection is able to search the article from the library in which the same contents precisely and systematically.

II. RELATED WORK

In many research, the DocMine Algorithm [4] has sets the major items of categorizing the documents and makes the warehouse for them, which increasing the efficiency of finding the big size of information. For the research using ODP (Open Directory Project) [5] for classifying into Meta Search, it increases the efficiency of database connection. In addition, there is a studying of library knowledge in order to find the automatic categorizing [6] by making the group of architectural words by human for connecting the groups. There is a research which creates ontology [7] for classifying into groups as standard system of the library. As a result, the created ontology is able to precisely classify into major groups.

For Golub (2006) [8], he used the technique of creating the group of words which related to engineering so that it can be able to classify in the website automatically. However, such classifications have not been in the categorizing the contents of the articles in Wikipedia. The researches which present the methods and analysis of classification of documents, articles and contents in the
Internet by the Dewey Decimal Classification Multiple Relations (DDC-MR), the result is shown by radar graph with the Mapping Method [9],[10] and is compared the relationship of Multiple Relations Data [1], [2], [3],[11] which makes the classification clearer and more accurate. Therefore, this paper analyzes the contents of the articles in Wikipedia for the Dewey Decimal Classification Multiple Relations (DDC-MR) into same standard as the classification of the library which is the international standard. It can connect the contents of the articles with the documents in the library and it makes benefit to Wikipedia in term of reference group which can be determined.

III. DDC-MR WIKIPEDIA CLASSIFICATION MODEL

In this paper, we uses the Dewey Decimal Classification Multiple Relations (DDC-MR) for analyzing content in Wikipedia in order to create a connection of information synchronization in the online system and documents in the library. This makes the information transfer faster and better and makes the connection of the contents in Wikipedia to international standard which has the steps of processing as shown in Fig. 2.

Step 1: Information Retrieval for extraction title article and categories keywords content from Wikipedia, using calculated weight keywords.


Step 3: Calculated percentage DDC-MR classes.

Step 4: Display percentage DDC-MR classes with Radar graph.

Step 5: Calculated X, Y point of Article Wikipedia with sum vector.

Step 6: Display X, Y point of Article Wikipedia with Scatter plot.

Step 7: Calculated angle degree of Article Wikipedia.

Step 8: Display angle degree of Article Wikipedia with Scatter plot.

Last step: Compare Wikipedia DDC-MR Classes with Library Book DDC Classes

Illustrates the steps of processing classification model of Wikipedia start from reading the Article title and Category keyword from Article Wikipedia. Then, using the Article Title and Category Keyword to be classified as the Dewey Decimal Classification Multiple Relations (DDC-MR) and calculated the percentage of each classes DDC-MR in Eq. (1) to store in the database.
\[ P_n = \frac{N_n \times 100}{\sum_{n=0}^{N_n}} \]  
(1)

\[ P_n = \text{Classes Percentage} \]
\[ N_n = \text{Keywords Number} \]
\[ n = \text{Classes Number} \]

Note: \( N \) is Keywords number of classes form Article Wikipedia.

\( n \) is order of classes divide into 10 value as follow:
- 000 class is order of class equaling 0,
- 100 class is order of class equaling 1,
- 200 class is order of class equaling 2,
- 300 class is order of class equaling 3,
- 400 class is order of class equaling 4,
- 500 class is order of class equaling 5,
- 600 class is order of class equaling 6,
- 700 class is order of class equaling 7,
- 800 class is order of class equaling 8,
- 900 class is order of class equaling 9

\( P \) is percentage of class from Keywords number.

Using the percentage of each class creates radar graph in order to show the relationship ratio of contents of the articles in each class and use the relationship in each class of content for calculating for point values in X-axis and Y-axis with Sum Vector equation as in Eq. (2)

\[
\begin{align*}
X &= \sum_{n=0}^{9} P_n \cos(36n) \\
Y &= \sum_{n=0}^{9} P_n \sin(36n)
\end{align*}
\]  
(2)

\[ X = \text{value X-axis} \]
\[ Y = \text{value Y-axis} \]
\[ \hat{P}_n = \text{Percentage of class} \]
\[ n = \text{Classes Number} \]

Using values of \( X \) and \( Y \) plot in the graph as Scatter plot in order to find the position of the overall contents by equation 3.

\[ \theta = \tan^{-1}\left(\frac{X}{Y}\right) \]  
(3)

\[ \theta = \text{Angles are measured in degrees} \]
\[ X = \text{value of X-axis} \]
\[ Y = \text{value of Y-axis} \]

IV. Classification Results

This paper result using DDC-MR for analyzing the contents of the articles in Wikipedia with 350 samples of Article in Wikipedia. The result of classification and overall analysis of content is shown in Table 1-3 and Fig 3-5.
In Table 3, shows percentage of DDC-MR with Radar Graph, after using their percentage for searching the value of X-axis and Y-axis by Sum Vector equation, X-axis and Y-axis and the content’s angle which the value of X-axis and Y-axis and position’s degree will be the representative of the overall contents. The result in Fig. 3 is calculated from Eq. (2) – Eq. (3), that shows value of X-axis and Y-axis plots of the content position of article from 350 examples which is calculated from Eq. (2). This result presents the overall contents of each class.

In Fig. 4, presents scatter plots of the content position of article from 350 examples which is calculated from Eq. (3). This result presents the overall contents of each class. If the contents of the articles in Article Wikipedia are similar, the position’s degree of each will be closed. From Figure 3, the position’s degree of the content from Scatter Plot.

In Fig. 5, depicts the mark point of content from Fig. 3. There are 4 mark points which is used too many content of article Wikipedia. The mark point is represented Id 10, Id 17, Id 57 and Id 62, respectively. Each point form scatter plot representation Article content Wikipedia have 9 classes of DDC-MR. This research is using DDC-MR for analyzing.
the contents in Wikipedia so that it is able to classify the contents of each article. Therefore, we can use such result to apply in the Wikipedia with the connection of the contents in Wikipedia and the books in the library to be able to identify the name of the books, prints and documents in the library which are in the same group or similar to the content in the Wikipedia. In addition, this research add the recommendation of the group for documents in the library which is related to the contents in Wikipedia in order to give the users the variety of information which is more reliable, quick, international and accurate for the needs of the users in table 4.

<table>
<thead>
<tr>
<th>Article Id</th>
<th>Article Title</th>
<th>Wikipedia Categories</th>
<th>Library Book</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Row vector</td>
<td>-Linear algebra -Matrices</td>
<td>000 - Data structures, algorithms, and software(005.133) -Adaptive image processing (006.6). -Digital graphics(006.6) etc.</td>
</tr>
<tr>
<td>400</td>
<td>-Vectors</td>
<td>400</td>
<td>-English in mechanical engineering(428.240) -The big picture : idioms as metaphors(458.24) etc.</td>
</tr>
<tr>
<td>500</td>
<td>-MATLAB : an introduction with applications(519.402) -Introductory and intermediate algebra(512.9) -Linear algebra and its applications(512.55) etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>-Engineering economic analysis(658.15) -Mathematics for retail buying(658.87) etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>-Architectural drawing(720.284) -Design bars(725.72) etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In table 4 we using article Wikipedia Title name is Row vector. That the results Wikipedia classification this article the major group is Mathematics and Logic. The system can suggestion category content keywords into 3 sub groups as follow: (i) Linear algebra (ii) Matrices and (iii) Vectors. But if we used classification with DDC-MR, that can defines 5 classes as follow: (i) 000: Generalities, (ii) 400: Language, (iii) 500: Pure sciences, (iv) 600: Technology and (v) 700: Art. Each main class is divided into sub classes, and each sub classes have several divisions, and each division has several sections, and sections can be linked to a book in library as follows.

For the experimental, we can describe the many of class classification from DDC-MR such as: class 000 (Generalities) can related to a book in library as (i) Data structures, algorithms, and software (005.133), (ii) Adaptive image processing (006.6) and Digital graphics (006.6) etc. Class 400 (Language) can related to (i) English in mechanical engineering (428.240), (ii) The big picture: idioms as metaphors (458.24) etc. Class 500 (Pure sciences) can related to (i) MATLAB: an introduction with applications (519.402), (ii) Introductory and intermediate algebra (512.9) and (iii) Linear algebra and its applications (512.55) etc. Class 600 (Technology) can related to (i) Engineering economic analysis (658.15), (ii) Mathematics for retail buying (658.87) etc. And class 700 (Art) can related to (i) Architectural drawing (720.284) and (ii) Design bars (725.72) etc.

In the previous results, can be seen that the Wikipedia had classification in major group only and the result in information of article content not had relationship or not supported in each group. But in this paper, we use the DDC-MR classification that the results had more relationship and more complex widely of content. Therefore, the classification with the DDC-MR is content analyzes with a recognized standard and widely used in Library. That can related to the content of articles in WIKIPEDIA to the name of book in Library, its can help the user received the reliable of data and the reference of data is acceptable.

V. CONCLUSION

This research is using DDC-MR for analyzing the contents in Wikipedia so that it synchronizes the information on online system with documents in the library which is more quick and better in the format of information transfer as following;

1) Benefit to Wikipedia in term of more reliable and clearer searching because Wikipedia is online information resource and is open source which does not limit the editor and it open for the users to check and edit the information by themselves. The good point is that the information can grow very quickly but the bad point is that there is no editor or specialist for checking the correction of all information in Wikipedia.

Using DDC-MR for analyzing the content in Wikipedia is able to compare the contents in Wikipedia with Text books in the library within the group so that this can be checked for the accuracy of the information and can ensure that the contents in Wikipedia is more reliable. In addition, DDC-MR will make the connection of the contents in Wikipedia within the international standard instead of the connection with only Keyword.

2) Benefit to the library in term of increasing value of information in the books, regarding the information is created by the academic to the books in the library is not utilized efficiently because the current users prefer to use the information which is accessed easier than the reliable one which reference to the academic work. Especially the
online system which is not able to connect to the information which is checked by process of the editor or specialist before published to the prints.

Using DDC-MR for analyzing the contents of Wikipedia will be able to search for the documents in the library which have similar contents and will give the details with clearer and more accurate than in the prints. This makes the prints in the library more utilizable.

3) Benefit to the development in each technology. The connection between Wikipedia and the library by DDC-MR gives users the quicker receiving information from Wikipedia and be able to check the accuracy of the information from the library more convenient. This results to the technology or research having the accurate reference and makes it more reliable and fast.

REFERENCES


