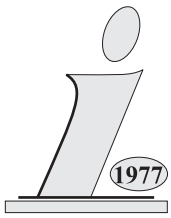
Volume 31 Number 1 March 2007 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics



The Slovene Society Informatika, Ljubljana, Slovenia

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

A Novel Roll-Back Mechanism for Performance Enhancement of Asynchronous Checkpointing and Recovery

Bidyut Gupta and Shahram Rahimi
Department of Computer Science, Southern Illinois University
Mail Code 4511, Carbondale, IL 62901-4511, USA
{bidyut, rahimi}@cs.siu.edu

Yixin Yang
Department of Biological Sciences, Emporia State University
Emporia, KS 66801, USA
yyang@emporia.edu

Keywords: asynchronous checkpointing, recovery, maximum consistent state

Received: May 26, 2006

In this paper, we present a high performance recovery algorithm for distributed systems in which checkpoints are taken asynchronously. It offers fast determination of the recent consistent global checkpoint (maximum consistent state) of a distributed system after the system recovers from a failure. The main feature of the proposed recovery algorithm is that it avoids to a good extent unnecessary comparisons of checkpoints while testing for their mutual consistency. The algorithm is executed simultaneously by all participating processes, which ensures its fast execution. Moreover, we have presented an enhancement of the proposed recovery idea to put a limit on the dynamically growing lengths of the data structures used. It further reduces the number of comparisons necessary to determine a recent consistent state and thereby reducing further the time of completion of the recovery algorithm. Finally, it is shown that the proposed algorithm offers better performance compared to some related existing works that use asynchronous checkpointing.

Povzetek: Opisan je izboljšan postopek okrevanja v porazdeljenih sistemih.

1 Introduction

Checkpointing and rollback-recovery are well-known techniques for providing fault-tolerance in distributed systems [1]-[5]. The failures are basically transient in nature such as hardware error [1]. Typically, in distributed systems, all the sites save their local states, known as local checkpoints. All the local checkpoints, one from each site, collectively form a global checkpoint. A global checkpoint is consistent if no message is sent after a checkpoint of the set and received before another checkpoint of the set [2]-[4], that is, each message recorded as received in a checkpoint should also be recorded as sent in another checkpoint. In this context, it may be mentioned that a message is called an orphan message if it is recorded as received in a checkpoint, but not recorded as sent in another checkpoint. The local checkpoints belonging to a consistent global checkpoint will be termed in the present work as globally consistent checkpoints (GCCs). After recovery from a failure processes in a distributed computation restart their computation from a consistent global checkpoint /state (CGS) of the system, i.e. from their respective GCCs. It may be noted that a consistent global checkpoint of a system is termed as a recent or a maximum one if, after the system recovers from a failure, the number of events (states) rolled back at each processor is a minimum [6].

To determine consistent global checkpoints, two fundamental approaches have been reported in the literature [1]-[9]. These are synchronous and asynchronous approaches. In the synchronous approach, processes involved coordinate their local checkpoint actions such that the set of all recent checkpoints in the system is guaranteed to be consistent. Although it simplifies recovery it has the following disadvantages: (1) additional messages need to be exchanged by the checkpointing algorithm when it takes each checkpoint; (2) synchronization delay is introduced during normal operation [5]. In the asynchronous approach, processes take checkpoints independently without any synchronization among them. Therefore, it is the simplest form of taking checkpoints. However, because of the absence of synchronization there is no guarantee that a set of local checkpoints taken will be a consistent set of checkpoints. That is, there may exist orphan messages between the local checkpoints. In order to get rid of the orphan messages while determining the GCCs, processes have to rollback. In such a situation, rolling back one process causes one or more other processes to roll back. This effect is known as the domino effect [5]. This is the main drawback of the asynchronous approach. So, a recovery algorithm has to search for the most recent consistent set of checkpoints before the system restarts its normal operation. Therefore, the recovery process is

quite complex while the checkpointing scheme is much simpler compared to the same in synchronous approach.

2 Related Works

In this work, we have considered asynchronous checkpointing approach because of its simplicity in taking checkpoints. So, in this section we state briefly the contributions of some noted related works. When processes take checkpoints independently, some or all of the checkpoints taken may be useless for the purpose of constructing consistent global checkpoints. A set of checkpoints can belong to the same consistent global snapshot if no zigzag path (Z-path) exists from a checkpoint to any other checkpoint [15]. In other words, absence of a Z-path means absence of any orphan message. A theoretical framework for characterizing quasi-synchronous algorithms has been presented in [12]. Quasi-synchronous checkpointing algorithms reduce the number of useless checkpoints by preventing the formation of noncausal Z-paths between checkpoints and advance recovery line. "Advancement of recovery line" is interpreted as follows: the more the recovery line is advanced, the less is the amount of computation to be redone by processes after the system of processes restart their normal operation; meaning thereby the reduction in the amount of rollback per process after the system recovers from failure. Depending on the degree to which the non causal Z-paths are prevented, quasi-synchronous checkpointing algorithms are classified into three classes namely [12], Strictly Z-Path Free (SZPF), Z-Path Free (ZPF), and Z-Cycle Free (ZCF).

Manivannan and Singhal [13] have presented a quasi-synchronous checkpointing algorithm which allows the processes to take checkpoints asynchronously and reduces the number of useless checkpoints by forcing processes to take additional checkpoints. In this checkpointing algorithm, each process maintains a counter which is periodically incremented and the time period is same in all the processes. When a process takes a checkpoint, it assigns the current value of its counter as the sequence number for the checkpoint. Each message is equipped (i.e. piggybacked) with the sequence number of the current checkpoint. If the sequence number accompanying the message is greater than the sequence number of the current checkpoint of the process receiving the message, then the receiving process takes a checkpoint and assigns the sequence number received in the message as the sequence number to the new checkpoint and then processes the message. Quasi-synchronous checkpointing algorithm makes sure that none of the checkpoints taken lies on a Z-cycle in order to make all checkpoints useful. Asynchronous recovery algorithms are also presented in this paper based on the checkpointing algorithm. A failed process needs to roll back to its latest checkpoint and requests other processes to rollback to their consistent (latest) checkpoints. The work claims to be free from any domino effect. However, arguably this work is more of a synchronous approach than an asynchronous approach; partly because all processes have identical time periods to take checkpoints, and checkpoint sequence numbers are used so that all the i^{th} checkpoints of all processes are taken at the same time (i.e., logically at same time). Hence, we argue that there is no question of domino effect as this work is not at all an asynchronous approach.

Gupta et al. [11] have proposed a hybrid roll forward checkpointing/recovery approach. Processes take checkpoints periodically and these time periods are different for different processes. Periodically, in absence of any failure, an initiator process invokes the algorithm to advance the recovery line; the duration of this period is assumed to be much larger than the time period of any individual process. Therefore, the domino effect is limited by this time period. The main advantages of this work are that each process may need to keep at most two checkpoints at any time, processes participate in the algorithm simultaneously ensuring re-execution time after a failure is limited by the period of execution of the algorithm, and finally, recovery is as simple as in the synchronous checkpointing/recovery approach.

Ohara et al. [14] proposed an uncoordinated checkpointing algorithm for finding a recovery line where a given checkpoint is the earliest. In this algorithm, each process maintains a set of all local checkpoints on that process in a local vector. All local checkpoints which are just behind a given checkpoint are initially assumed to form a consistent global checkpoint. The algorithm checks happened-before relation for any coupled local checkpoints belonging to an ordered global checkpoint set. If there exists any happened-before relation, it replaces a local checkpoint with a successive local checkpoint of the same process. The algorithm may end by either finding a recovery line or running out of local checkpoints to be replaced.

Venkatesan and Juang [16] presented an asynchronous checkpointing algorithm where each process take checkpoints independently and keeps track of the number of messages it has sent to other processes as well as the number of messages it has received from other processes. The algorithm is initiated by the process which fails and is recovered from thereafter or when it learns about process failure. During its each iteration, a process needs to compare the number of messages received by it and the actual number of messages sent by the other process, at each of its checkpoint starting from the most recent one. The received vectors corresponding to all the checkpoints including the current one and the one where next iteration starts, need to be fetched from the storage in order to decide the checkpoint for the next iteration to start with.

3 System Model

The distributed system has the following characteristics [1], [6], [10]:

1. Processes do not share memory and they communicate via messages sent through channels.
2. Channels are made virtually lossless and order of the messages is preserved by some end-to-end transmission protocol.

3. When a process fails, all other processes are notified of the failure in finite time. We also assume that no further processor (process) failures occur during the execution of the algorithm. In fact, the algorithm must be restarted if there are further failures.

Below we state the problem considered in this work.

Problem Formulation: In this work, we have considered asynchronous checkpointing approach because of its simplicity in taking checkpoints. That is, processes take checkpoints periodically and each process determines independently its time period of taking its checkpoints. So, different processes may have different time periods for taking their checkpoints. After the system recovers from a failure, processes start from the recent consistent state of the system. However, the main drawback of this approach is that determining a consistent global checkpoint may involve a very large number of pairwise comparisons of checkpoints belonging to different processes because of the presence of a possible domino effect. In absence of any hybrid approach [11], in the worst case, all checkpoints of all processes may have to be compared. However, asynchronous checkpointing approach is suitable for highly reliable systems where failures occur very seldom.

In this work, our objective is to design an efficient recovery algorithm that will reduce considerably the number of unnecessary pairwise comparisons of checkpoints while determining a consistent global checkpoint. In other words, our objective is to identify a priori the checkpoints that can not be the GCCs so that we can exclude these checkpoints from comparison resulting in a fast determination of a recent consistent global checkpoint (state) of the system. Note that an initial version of this work has appeared in [17].

4 Data Structures

Let us assume that the distributed system under consideration consists of n processes. Each process P_i maintains a vectors V_i of length n. The V_i vector records the number of messages process P_i has sent to every other process with the exception that the element $v_{i,i}$ (=V_i (i)), i.e. the number of messages process P_i has sent to itself will be always zero. The V_i vector is described below:

$$V_i = [v_{i,0}, v_{i,1}, \ldots . , v_{i,i}, \ldots . , v_{i,n-1}]$$

where $v_{i,j} = V_i$ (j) and represents the number of messages sent by process P_i to process P_j, and $v_{i,i}$ is always zero.

All entries in V_i are initialized to zero. Each time process P_i decides to send a message m to process P_j, then V_i(j) is incremented by one. This facilitates process P_i to know how many messages it has sent to process P_j. In this work, $C_{j,r}$ represents the r^{th} checkpoint taken by process P_j. Sometimes when mentioning the checkpoint number is irrelevant, we simply use C_j to denote a checkpoint taken by P_j. Each process P_i also maintains a linear list R_i of dynamically growing length. At any given time t, the length of the list R_i (i.e. the number of the entries in the list) is equal to the number of checkpoints taken by P_i till time t. For example, the length of the list is 3 at the 3^{rd} checkpoint of process P_i where as its length will be 4 at its 4^{th} checkpoint and so on. The list R_i is described as $R_i = [r_{i,1}, \ldots . , r_{i,r}, \ldots\ldots]$, where $r_{i,r} = R_i$ (r) and represents the number of messages received by process P_i from all other processes till its r^{th} checkpoint. Each such list is initially empty.

Each process stores its vectors and the lists together with the corresponding checkpoints in stable storage. Also copies of the lists and the vectors are stored in the respective local memories of the processors running the processes. It offers their faster access than to access them from stable storage whenever possible. In addition, each process maintains a Boolean flag. This flag is used to convey some specific information (described later).

5 Observations

Consider the system of three processes P_1, P_2, and P_3 as shown in Fig. 1. The vectors V_1, V_2, and V_3 initially have all their entries set to zero. The lists R_1, R_2, and R_3 are initially all empty. By the time process P_1 takes its first checkpoint $C_{1,1}$, it did not send any message to P_2 or P_3. So its V_1 vector is [000]. Also, process P_1 received one message before it took its first checkpoint; so now the list R_1 has one entry, i.e. $R_1 = [1]$. By the time process P_1 takes its second checkpoint $C_{1,2}$, it has already sent one message to P_2. So it increments $V_1(2)$ by 1 and the vector V_1 is now = [010]. Also, process P_1 has not received any messages (from P_2 or from P_3) before it takes its second checkpoint. So the list R_1 at $C_{1,2}$ is [1,1]. In the same way, the vector and the list are updated at each checkpoint of each process. This example will be used later in this paper to illustrate the working principle of our proposed algorithm.

We assume that a process P_i after recovery from its failure acts as the initiator process, i.e., P_i is responsible for invoking the recovery algorithm. To start with P_i sends a message requesting all P_j, $0 \le j \le n-1$, $j \ne i$, to send to it their respective V_j vectors corresponding to their latest checkpoints. Upon receiving the request, every process P_j sends its V_j to P_i. After receiving the vector V_j from all processes the initiator process P_i forms a two dimensional array V_N. It is written below.

$$V_N = \begin{vmatrix} 0 & v_{0,1} & \cdots & v_{0,n-1} \\ v_{1,0} & 0 & \cdots & v_{1,n-1} \\ . & . & \cdots & . \\ v_{j,0} & v_{j,1} & \cdots & v_{j,n-1} \\ . & . & \cdots & . \\ v_{n-1,0} & v_{n-1,1} & \cdots & 0 \end{vmatrix}$$

where the j^{th} row represents V_j, $0 \le j \le n-1$. The initiator process then computes the column sums to create the following vector:

$$V_C = [v_c^0, v_c^1, \ldots, v_c^j, \ldots, v_c^{n-1}]$$

where v_c^j = column sum of the entries of the j^{th} column of V_N and is given as

$$v_c^j = V_C(j) = \sum V_N(i, j), \text{ for } i = 1 \text{ to } n.$$

Therefore, v_c^j represents the total number of messages sent to process P_j by all other processes as recorded in each sending process' latest checkpoint. The initiator process P_i then unicasts v_c^j (= $V_C(j)$) to process P_j. After receiving v_c^j from P_i, each process P_j computes $D_j = R_j(r) - v_c^j$, assuming that the last checkpoint of process P_j is the r^{th} checkpoint ($C_{j,r}$). The difference D_j (if >0) gives the exact number of orphan messages received by a process P_j till its checkpoint $C_{j,r}$, from all other processes in the system. Initiator process P_i also does similar computation to determine the exact number of orphan messages (if any) it has received till its latest checkpoint $C_{i,r}$. Proof of this statement is given later.

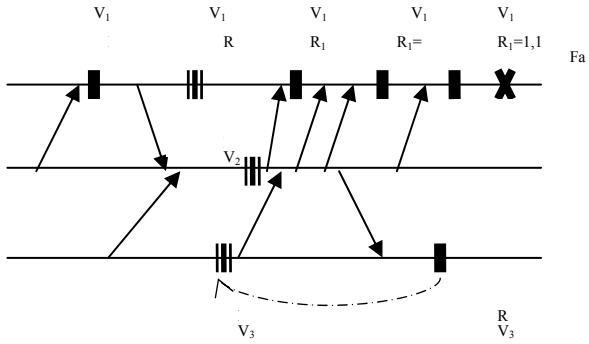


Figure 1: Vectors (V_i) and lists (R_i) for i = 1, 2, and 3

Observe that for every process P_j, v_c^j and $R_j(r)$ may not be identical, because some of the sent messages (recorded already by the sending processes at their respective latest checkpoints) may not have arrived yet at P_j (i.e. $v_c^j \geq R_j(r)$), or some of the received messages (by P_j) may not have been recorded at the latest checkpoints of some sending processes because these messages may have been sent after their latest checkpoints (i.e. $v_c^j \leq R_j(r)$).

Assume that the last checkpoint of process P_j is the r^{th} checkpoint ($C_{j,r}$) and D_j is greater than zero (D_j >0). Search in the list R_j is performed backwards, starting with its last component. Thus, we search the proceeding entries of the list R_j from $R_j(r)$ till the first $R_j(m)$ so that $R_j(r) - R_j(m) \geq D_j$, (m < r). Then, the checkpoints $C_{j,r}$, …, $C_{j,\,m+1}$ are excluded from the consideration of GCC composition, i.e. these checkpoints will be skipped. So, now we start from the checkpoint $C_{j,m}$ of process P_j. The vector V_j at checkpoint $C_{j,m}$ along with the Boolean flag "1" are sent to the initiator process P_i for the computation of the next iteration.

In the next iteration, if D_j is smaller than or equal to zero ($D_j \leq 0$), which means that process P_j has not received any orphan message till the checkpoint $C_{j,r}$. process P_j will send the flag "0" to the initiator process P_i. The initiator process P_i will use the vector V_j at $C_{j,r}$ for the computation of the next iteration. Initiator process P_i is also involved in similar computation like any other process P_j to determine its appropriate vector V_i needed for the computation of the next iteration. This will be repeated until all processes send "0" flags to the initiator process P_i and P_i's own flag is also 0 . Then the initiator process P_i will notify all processes to rollback to their respective latest checkpoints at which their corresponding flags have the value 0 each. Thus, this set of checkpoints is a globally consistent checkpoint (proof is given later).

The following observations are necessary for designing the recovery algorithm.

Lemma 1: Let $C_{j,r}$ be the latest checkpoint of process P_j at time t. If $D_j > 0$, then process P_j has received a total D_j number of orphan messages from other processes.

Proof: $R_j(r)$ represents the total number of messages process P_j has received so far from all other processes till time t. Also v_c^j represents the total number of messages sent by all other processes to P_j as recorded in their latest checkpoints. Therefore $D_j > 0$ means that at least some process P_i ($i \neq j$) has sent some message(s) to P_j after taking its latest checkpoints. It also means that the sending processes have not yet been able to record these D_j messages. Since all such D_j messages have been received and recorded in P_j's latest checkpoint, but remain unrecorded by the sending processes, therefore P_j has received D_j number of orphan messages from the rest of the processes with respect to the checkpoint $C_{j,r}$. ■

Lemma 2: If $D_j \leq 0$, process P_j has not received any orphan message.

Proof: $D_j = 0$ means that the number of messages received by P_j is equal to the number of messages sent to P_j and these sent (also received) messages have already been recorded by the sending processes in their latest checkpoints. Therefore the received messages can not be orphan.

Also, $D_j < 0$ means that the number of the messages received by P_j is less than the number of messages sent to it. Now v_c^j is the total number of messages sent by all other processes to P_j as recorded in the latest checkpoints of the sending processes. It means that all messages received by P_j have already been recorded by the senders. Hence none of such received messages can be an orphan. Hence the proof follows. ■

Lemma 3: Let $D_j > 0$ at the checkpoint $C_{j,r}$ of process P_j and let m denote the largest integer that satisfies $R_j(r) - R_j(m) \geq D_j$ (m < r). Then none of the checkpoints $C_{j,r}$, $C_{j,r-1}$, …, $C_{j,\,m+1}$ belongs to the set of the globally consistent checkpoints.

Proof: Because m is the largest integer that satisfies $R_j(r) - R_j(m) \geq D_j$ (m < r), the relation $R_j(r) - R_j(i) < D_j$ is established for any i ($m+1 \leq i \leq r$). Moreover, according to Lemma 1, P_j has received exactly D_j number of orphan messages from all other processes. So there must be at least one orphan message received by process P_j before $C_{j,r}$, and the same also is true before every checkpoint between $C_{j,r}$ and $C_{j,m}$. Hence, none of the checkpoints $C_{j,r}$, $C_{j,r-1}$, …, $C_{j,\,m+1}$ can belong to the set of the globally consistent checkpoints. ■

Theorem 1: Given a set $S^* = \{C_{j,r}\}$ of n checkpoints, one from each P_j, $0 \le j \le n-1$, if for every checkpoint $C_{j,r}$, its corresponding $D_j \le 0$, then S^* is the set of the globally consistent checkpoints.

Proof: Since $D_j \le 0$, for each process P_j, ($0 \le j \le n-1$) at its checkpoint $C_{j,r}$ ε S^*, therefore, all received messages by any such process P_j have already been recorded as sent by the sending processes in their corresponding checkpoints. Hence, according to Lemma 2 none of the messages received by process P_j is an orphan message. This is true for all processes. Therefore, the system of n processes does not have any orphan messages with respect to the checkpoints of the set S^*. Hence the set S^* is the set of globally consistent checkpoints. ■

Before we present the algorithm formally, we give an illustration of its working principle using the example of Fig. 1.

An illustration: Suppose a failure 'f' occurs on the processor running the process P_1. The process P_1 that became faulty, acts as the initiator after recovery from failure. After the system recovers from the failure, to start with, initiator process P_1 broadcasts a request asking the other two processes P_2 and P_3 to send their respective vectors V_2 and V_3 corresponding to their latest checkpoints $C_{2,1}$, and $C_{3,2}$. In this example, the three latest checkpoints of processes P_1, P_2, and P_3 before the failure occurs are $C_{1,5}$, $C_{2,1}$, and $C_{3,2}$. The respective vectors V_1, V_2, and V_3 at the three latest checkpoints are [010], [100] and [020]. After receiving all these vectors, P_1 (it becomes the initiator after recovery from failure) forms a two dimensional array V_N. It is written below:

$$V_N = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{vmatrix}$$

P_1 creates the vector V_C = [130] and unicasts v_c^j to each process P_j, for j = 1, 2, and 3. After receiving v_c^j from P_i each process P_j computes D_j ($= R_j(r) - v_c^j$) (assuming the last checkpoint of P_j is the r^{th} checkpoint) to determine the total number of orphan messages (if any) it has received with respect to its latest checkpoint and also P_i does the same. The lists R_1, R_2, and R_3 at the latest checkpoints ($C_{1,5}$, $C_{2,1}$, and $C_{3,2}$) of processes P_1, P_2 and P_3 are [1,1,2,4,5], [2] and [0,1] respectively. P_1 finds that $D_1 = (5-1) = 4$; so it has received 4 orphan messages. It calculates the difference between $R_1(5)$ and $R_1(2)$ and finds that $R_1(5) - R_1(2) = 4 = D_1$; so process P_1 now considers the vector V_1 (= [010]) at $C_{1,2}$ along with a flag "1" for the computation of the next iteration. P_2 finds that it has not received any orphan message because $D_2 = (2-3) < 0$. So it sends the same vector [100] and a flag "0" to P_1. Process P_3 finds that $D_3 = (1-0) = 1$; so it has received an orphan message. It calculates the difference between $R_3(2)$ and $R_3(1)$ and finds that $R_3(2) - R_3(1) = 1 = D_3$; so process P_3 now sends the vector V_3 (= [010]) at $C_{3,1}$ along with a flag "1" to P_1 for the computation of the next iteration. In the second iteration, P_1 forms the following two dimensional array.

$$V_N = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}$$

P_1 creates the vector V_C = [120] and unicasts v_c^j to process P_j, for j = 1, 2, and 3. P_1 finds that it has not received any orphan message because at $C_{1,2}$, its $D_1 = 1 - 1 = 0$. So, it sets its flag to 0. P_2 also finds that it has not received any orphan message because at $C_{2,1}$, its $D_2 = 2 - 2 = 0$; and it sends the flag "0" to P_1. Similarly, P_3 finds that it has not received any orphan message because at $C_{3,1}$, its D_3 ($= R_3(1) - v_c^3$) $= 0 - 0 = 0$, and it sends a flag "0" to P_1. Thus, P_1 receives flag 0 from each process including its own flag set to 0. It then notifies each process to rollback to the current checkpoints corresponding to these flags (= 0). At this time, none of the processes needs to roll back further and hence P_1 terminates the algorithm. Thus the algorithm terminates after two iterations. Therefore the GCCs belonging to the maximum consistent state are $C_{1,2}$, $C_{2,1}$, and $C_{3,1}$.

It may be noted that in each iteration we need to fetch only the latest R_j for each process P_j and some V_j vectors (not all) to determine the GCCs. In each iteration, the checkpoints that can not be the GCCs are identified and their vectors V_j are not fetched at all. That is, the presented approach will not repeat its operation unnecessarily for these vectors corresponding to these non-GCCs. It definitely makes the approach fast and efficient. Observe what happens if we do not consider the above idea to determine the GCCs. It is stated below.

First, $C_{1,5}$, $C_{2,1}$, and $C_{3,2}$ are considered and compared pairwise to determine if they are globally consistent. Since $C_{1,5}$ and $C_{3,2}$ are not, so in the next iteration $C_{1,4}$, $C_{2,1}$, and $C_{3,1}$ are considered pairwise. But $C_{1,4}$ cannot be a GCC. Therefore $C_{1,3}$, $C_{2,1}$, and $C_{3,1}$ are now considered. But since $C_{1,3}$ can not be a GCC, therefore $C_{1,2}$, $C_{2,1}$, and $C_{3,1}$ are now considered. This time it is found that these three checkpoints are globally consistent. Therefore four iterations for pairwise comparisons of three checkpoints, one from each process, are needed to determine the GCCs as opposed to only two when the approach presented in this work is followed. It also means that the number of trips to the stable storage for fetching checkpoints can also be reduced to a good extent in the proposed approach. It definitely makes our algorithm fast. Moreover when processes take large number of checkpoints before a failure occurs, our approach may offer even much better performance from the viewpoint of a possible large reduction in the number of iterations (i.e. the number of trips to stable storage as well) to determine the GCCs. As a result, the recovery scheme also will be faster. Besides, it is clear from the example that each process P_j simultaneously identifies the checkpoints that cannot be globally consistent and therefore these checkpoints should be skipped. This parallelism of the algorithm further enhances the speed of execution of the recovery approach.

6 Algorithm to Determine Globally Consistent Checkpoints

In the following algorithm we assume that process P_i was faulty. So, it becomes the initiator of the recovery algorithm after it recovers from the failure.

6.1 Algorithm Recovery

Input: Given the latest n checkpoints, one for each process P_j, $0 \leq j \leq n-1$, for an n process system and the corresponding vectors V_j and lists R_j at these n checkpoints.

Output: A set of globally consistent checkpoints (maximum consistent state of the system).

The responsibilities of each participating process P_j and the initiator process P_i are stated in Fig. 2.

Proof of Correctness: Each process P_j repeats its steps 1, 2, 3, and 4 to arrive at a checkpoint that has not recorded the receipt of any orphan message from the other processes (using the observations of Lemmas 1, 2, and 3). In other words, it identifies the checkpoints that can not belong to the set of the globally consistent checkpoints and skips them. This decision is taken by identifying a checkpoint $C_{j,m}$ such that m is the largest integer that satisfies $R_j(r) - R_j(m) \geq D_j$ ($m < r$). None of the checkpoints $C_{j,r}$, $C_{j,r-1}$, …, $C_{j,m+1}$ can belong to the set of the globally consistent checkpoints and they are skipped. However, the initiator process P_i decides when to terminate the algorithm, i.e., when the checkpoints can become globally consistent. Process P_i checks to see if all processes send flags of 0, i.e. $D_j \leq 0$ for each process P_j. If so, the algorithm terminates according to Theorem 1. Note that the condition $D_j \leq 0$ must always occur during the execution of the algorithm. It may be observed that in the worst case, because of some typical communication pattern, the domino effect may force each process to restart from its initial state where for each process P_j we always have $D_j = 0$. Besides, since the algorithm starts with the latest checkpoints, the number of events (states) rolled back at each processor is a minimum. This is true because, in its Step 4 each process P_j skips only the checkpoints that are non GCCs. Thus the algorithm determines the maximum consistent state of the system as well. ■

6.2 Advantages of the proposed approach

The presented algorithm offers the following advantages. During its each iteration, each process P_j determines the checkpoints that can not be the GCCs. Therefore, the algorithm is able to avoid any unnecessary computations of V_C corresponding to these non GCCs. The presented algorithm skips checkpoints that do not belong to the set of the globally consistent checkpoints; thus it avoids many unnecessary pairwise comparisons. It also means that the number of trips to the stable storage for fetching checkpoints can also be reduced to a good extent in the proposed approach. It definitely makes the algorithm fast and efficient. The simultaneous execution of the algorithm by all participating processes also contributes to the speed of execution of the algorithm. Besides, the algorithm can find the maximum number of checkpoints to be skipped by determining the largest integer m, which satisfies $R_j(r) - R_j(m) \geq D_j$. This guarantees significant reduction in the iterations of computation.

6.3 Performance

Message complexity**:** Suppose the termination of the algorithm requires the construction of the vector V_C by the initiator process P_i to occur k times (i.e. k number of iterations). During each such time every process in the n-process system exchanges a couple of messages with the initiator process P_i. Thus, *O(n)* messages are sufficient for each time. Thus, considering k times, the message complexity of the algorithm is *O(kn)*.

Besides message complexity, another factor that must be considered as a performance measure is the number of pairwise comparisons of the checkpoints among the processes that is needed to be performed by any asynchronous checkpointing/recovery approach. This is done in order to determine a consistent global state of the system. Obviously larger the number of such comparisons, larger is the execution time of the recovery algorithm. This has been discussed in the previous subsection.

It may be noted that the number of such pairwise comparisons is also related to the number of times checkpoints are fetched from stable storage, i.e. the number of trips to the storage. The time spent on such trips may be substantial enough to affect to a good extent the speed of execution of any recovery algorithm. One possible solution may be to fetch a large number of checkpoints at a time. However, it may not be a good idea at all in many situations; for example, a process may end up in fetching too many when that many are not needed, or too little when more are needed. So, it becomes quite arbitrary about how many checkpoints should be fetched at a time. Therefore, it is wise to consider that a process will fetch one checkpoint at a time and in fact, this is true for all existing asynchronous checkpointing / recovery algorithms. In the following analysis we consider the fact that larger the number of pairwise comparisons of checkpoints, larger is the number of trips to stable storage, and therefore, larger is the execution time as a result.

In our analysis we will not consider complexity due to message size, as most related works including ours use control messages of reasonably small size and all these works differ mainly in terms of the number of comparisons, number of iterations, and the number of control message needed to determine a consistent global state. It may be noted that computing this number of comparisons is not very straightforward because it depends solely on the nature of the distributed computations. However, we give an approximate analysis which may not be very accurate; still it will offer a clear understanding of the advantages of our algorithm

---

Initiator process P_i:

Step 1: It asks every process P_j to send its V_j corresponding to its latest checkpoint $C_{j,r}$;
Step 2: It receives all V_j for $0 \leq j \leq n-1$;
Step 3: It computes $V_C = v_c^0\ v_c^1 \ldots v_c^j \ldots v_c^{n-1}$;
Step 4: It unicasts v_c^j to each P_j;
Step 5: It computes D_i by calculating $(R_i(r) - v_c^i)$;

if $D_i > 0$
It searches the list R_i till it finds the largest integer m (< r) that satisfies $R_i(r) - R_i(m) \geq D_i$. Then it sets its flag to 1 and considers V_i corresponding to its checkpoint $C_{i,m}$ (i.e. $C_{i,r}$ is replaced by $C_{i,m}$) for the next iteration;
/ Checkpoints $C_{i,r}$, $C_{i,r-1}$, ..., $C_{i,m+1}$ are skipped */*
else
It sets its flag to 0 and considers V_i at $C_{i,r}$ for the next iteration;

Step 6: It receives the flag and V_j from each process P_j;

if flag = 0 for each process P_j , $0 \leq j \leq n-1$
P_i asks each process P_j to restart the application program from its last checkpoint corresponding to which D_j ;

P_i resets its vector V_i to zero and list R_i to an empty list corresponding to its restarting checkpoint at which $D_i \leq 0$;
It restarts computation; */* its responsibility associated with the algorithm is finished */*

/ Globally consistent checkpoints belonging to the maximum consistent state are determined*/*

else
Control flows to Step 3;

Process P_j:

Step 1: P_j receives request from P_i;

if P_i has requested to restart
P_j resets its vector V_j to zero and list R_j to an empty list corresponding to its restarting checkpoint at which $D_j \leq 0$;
It restarts computation;
else
It sends V_j corresponding to its latest checkpoint $C_{j,r}$ to the initiator process P_i;

Step 2: It receives v_c^j from P_i;
Step 3: It computes D_j by calculating $(R_j(r) - v_c^j)$;
Step 4: if $D_j > 0$
It searches the list R_j till it finds the largest integer m (< r) that satisfies $R_j(r) - R_j(m) \geq D_j$. Then it sends a flag of 1 and V_j to P_i corresponding to its checkpoint $C_{j,m}$ (i.e. $C_{j,r}$ is replaced by $C_{j,m}$);
/ Checkpoints $C_{j,r}$, $C_{j,r-1}$, ..., $C_{j,m+1}$ are skipped */*
else
It sends a flag of 0 and V_j at $C_{j,r}$ to the initiator process P_i;

Figure 2: The responsibilities of each participating process P_j and the initiator process P_i

over some other noted asynchronous checkpointing / recovery approaches [14], [16]. It is stated below.

Let the system consist of n processes. For simplicity we assume that after a failure occurs and the system recovers from it, each process will skip on an average its latest (r-1) checkpoints to restart its computation. Thus a process P_j will skip its latest (r-1) checkpoints $C_{j,m+2}$, …. , $C_{j,r+m}$. We also assume that the set $\{C_{0,m+1}, C_{1,m+1}, \ldots, C_{n-1,m+1}\}$ represents the globally consistent checkpoint (maximum consistent state) of the system and our algorithm will determine it in k number of iterations. In this simple model, we consider a recovery approach associated with asynchronous checkpointing scheme in which the pairwise comparisons to determine checkpoints' consistency involves first the checkpoints of the set $\{C_{0,m+r}, \ldots. , C_{n-1,m+r}\}$, followed by the set $\{C_{0,m+r-1}, \ldots. , C_{n-1,m+r-1}\}$, … and so on, and finally the set $\{C_{0,m+1}, \ldots., C_{n-1,m+1}\}$which is the globally consistent state. Therefore, the total number of comparisons is given by [r x{n(n-1)}/2]. Note that this may not be the exact way to perform the comparisons in a particular case; still it offers a clear view of how complex it can be. In general, a checkpoint(s) in one set may also have to be compared with a checkpoint(s) in another set. On the other hand, not necessarily all checkpoints in a set may be needed to be pairwise compared. It depends on the nature of the distributed computations. So the actual number of comparisons may be larger or smaller than the number [r× {n(n-1)}/2]. Anyway, it is clear that this number is much larger than the total number of comparisons k×n, offered by our approach, where n is the number of parallel comparisons to test if $D_j > 0$ in each iteration and $1 \leq k \leq r$. Observe that in the worst case, the number of comparisons of the proposed approach may become [r× {n(n-1)}/2]. Below we have compared the performance of our approach with the approaches in [14], [16].

6.3.1 Comparison with Ohara et. al. [14]

Ohara et al. [14] have proposed an asynchronous approach for finding a recovery line where a given checkpoint is the earliest. All the local checkpoints which are just behind a given checkpoint are initially assumed to form a consistent global checkpoint. In this algorithm, happened-before relations are checked for every coupled local checkpoints belonging to an ordered global checkpoint set. If there exists any happened-before relation, it replaces a local checkpoint with a successive local checkpoint of the same process. The algorithm may end by either finding a recovery line or running out of local checkpoints to be replaced. This leads to exhaustive comparisons of happened before relations for every coupled local checkpoints. The number of such comparisons is approximately [r× {n(n-1)}/2] as calculated earlier. In our algorithm, it skips the checkpoints that do not belong to the set of the globally consistent checkpoints. Thus, our algorithm reduces to a good extent unnecessary pairwise comparisons of the checkpoints to determine global consistent checkpoint of the system. Performance comparison of the above mentioned approach [14] and our approach is shown in Fig. 4.

Fig. 3 illustrates how the number of comparisons is affected with the increase in the average number of checkpoints per process (r) in the asynchronous approach [14] and in our approach. Fig. 4 shows the variation of the number of comparisons with the increase in the number of processes (n). Both figures highlight the advantages offered by our approach, i.e. considerable amount of reduction in the number of comparisons in our approach. It helps the processes to restart their computation related to the distributed application much faster after the system recovers from a failure.

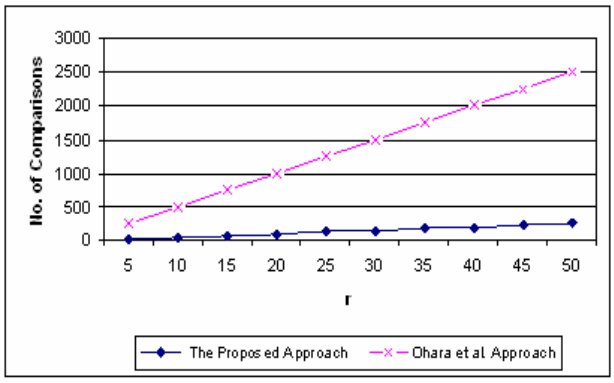


Figure 3: Number of comparisons vs. the average number of checkpoints per process (r).

6.3.2 Comparison with Venkatesan et. al. [16]

Venkatesan and Juang [16] presented an asynchronous checkpointing algorithm where each process takes checkpoints independently and keeps track of the number of messages it has sent to other processes as well as the number of messages it has received from other processes. The existence of orphan messages is discovered by comparing the number of messages sent and received. The algorithm is initiated by the process when a failure occurs or when it learns about process failure.

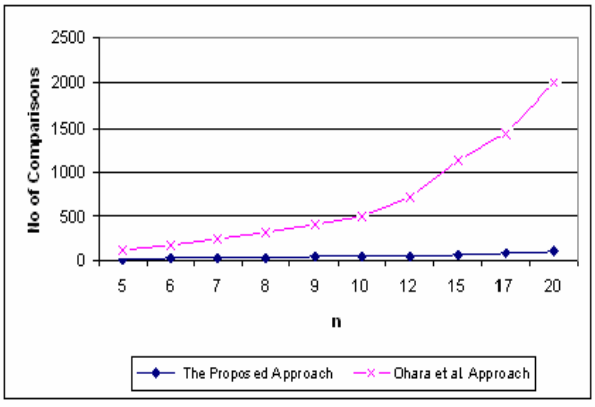


Figure 4: Number of comparisons vs. the number of processes (n).

During its each iteration, a process needs to compare the number of messages received by it and the actual number of messages sent by the other process, at each of its checkpoints starting from the recent one. The received vectors corresponding to all the checkpoints including the current one and the one where next iteration should start, need to be fetched from the storage in order to decide the checkpoint for the next iteration to start with. It means that the number of trips to the storage for fetching the information related to the received message (for the purpose of comparison) will be equal to the number of checkpoints starting from the current checkpoint all the way to the checkpoint where the next iteration should start.

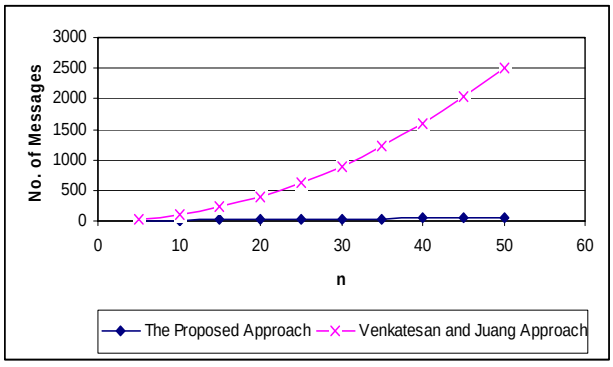


Figure 5: Number of control messages vs. the number of processes (n).

In our algorithm, the decision about the checkpoint at which the next iteration should start is based on the R-vector at the recent checkpoint only. This algorithm skips checkpoints that do not belong to the set of the globally consistent checkpoints by examining this R-vector only. Therefore, in order to determine the checkpoint for the next iteration to start with, the number of trips to the storage is only one per iteration. This means that the total number of trips to complete the execution of our algorithm is reduced to a good extent compared to that in [16]. We now compare the two algorithms based on the number of control messages needed to execute the respective algorithms.

In [16], in each iteration, for an n-process system n(n-1) messages are exchanged among the processes. Thus, $O(n^2)$ messages are exchanged in each iteration. In our algorithm, 3(n-1) messages are exchanged in each iteration. Thus, $O(n)$ messages are sufficient in each iteration in our algorithm where n is the number of processes in the system. Fig. 5 shows the message complexity comparison of the two algorithms with the increase in the number of processes. This figure clearly shows the advantage offered by our algorithm over the one in [16].

7 Further Enhancement

We have seen that the linear list R_j maintained by a process P_j increases dynamically. If the application program has large execution time and there is seldom any failure during its execution, the length of the lists may become too large; thereby it may consume considerable amount of memory. To solve this problem, i.e. to keep the list from growing too much we will propose a simple solution in this section. The following operation is needed in the implementation of the idea.

We define the subtraction operation on two vectors V_j of process P_j at its two checkpoints $C_{j,m}$ and $C_{j,s}$ with (s > m) as follows:

V_j at $C_{j,s} - V_j$ at $C_{j,m}$ = [($v_{j,0}$ at $C_{j,s} - v_{j,0}$ at $C_{j,m}$), … , ($v_{j,n-1}$ at $C_{j,s} - v_{j,n-1}$ at $C_{j,m}$)] = [($v_{j,p}$ at $C_{j,s} - v_{j,p}$ at $C_{j,m}$)]
for $0 \le p \le n-1$

We now state the basic idea to keep the growing lengths of the lists in control. This idea has been used in designing the enhanced recovery algorithm stated later in this section. It may be noted that the recovery algorithm stated earlier does not consider the use of this idea.

In absence of any failure an algorithm runs periodically (say the time period is T which should be much larger than the time period of any individual process) to put a limit on the length of the R-vector. The lengths of the lists (R-vectors) may then be limited by the number of checkpoints taken by the processes during the time interval (T) between two successive executions of the algorithm. Besides in doing so, this also advances the recovery line in the event that a recent recovery line exists other than the one found during the previous execution of the algorithm. In effect, the number of comparisons of the checkpoints to determine a recent consistent state may also drastically reduce since there is a possibility that the algorithm will consider in a particular run only the checkpoints which the processes take during the interval T. Therefore, this enhanced algorithm, in general, may take much less time to complete its execution compared to Algorithm Recovery. Also note that at the completion of the l^{th} execution of the algorithm a process P_j will have in stable storage only its recent globally consistent checkpoint, say $C_{j,m}$ and any other checkpoint (s) it has taken thereafter and prior to the start of the l^{th} periodic execution of the algorithm.

In describing the following two rules for updating the lists R_j and the vectors V_j of a process P_j we have assumed that the latest globally consistent checkpoint of process P_j is $C_{j,m}$ as determined by the l^{th} execution of the algorithm and it has taken (k-m) more checkpoints thereafter and prior to the start of the l^{th} periodic execution of the algorithm.

Rule 1: Updated R_j at $C_{j,m}$ = {} and updated V_j at $C_{j,m}$ = [00…0]

Rule 2: Updated R_j at $C_{j,s}$ for ($m+1 \le s \le k$) = [($R_j(m+1) - R_j(m)$), … , ($R_j(s) - R_j(m)$)], and
Updated V_j at each $C_{j,s}$ = [($v_{j,p}$ at $C_{j,s} - v_{j,p}$ at $C_{j,m}$)]
for $0 \le p \le n-1$

When we implement the above idea of reducing the lengths of the lists, either of the following two approaches can be adopted:

Approach 1: When a failure occurs and the system recovers from the failure, the algorithm is run again in

spite of its periodic execution, with the hope that a recent (maximum) consistent state may be found which is not identical to the one determined by its last periodical execution. In such a situation the time to complete the application will be less because of the advancement of the recovery line.

On the other hand, if such a situation as mentioned above does not exist, the algorithm will output the same consistent state as determined in its last periodic execution. In this case, however, the application will take an additional amount of time equal to the execution time of the algorithm for its completion.

Approach 2: After the system recovers from a failure all processes restart from their respective globally consistent checkpoints which have already been determined by the algorithm's last periodic execution prior to the occurrence of the failure. The recovery becomes as simple as that in a synchronous approach. However, since this approach does not look for the possible existence of a recent consistent state other than the already existing one, therefore the time to complete the application may increase.

Observe that irrespective of which approach is followed, the next periodic execution of the algorithm will occur T time units after the system restarts. About when to apply a specific rule, Rules 1 and 2 will be implemented when the algorithm runs periodically in absence of any failure. Rule 1 is also implemented when determination of a consistent global state of the system is needed after the system recovers from a failure (Approach 1). In the following algorithm we have considered a combination of the two approaches.

For the selection of an initiator process for running the algorithm periodically, we consider that each process P_i maintains a local CLK_i variable which is incremented at periodic time interval T. It also maintains a local counter denoted as $counter_i$, initially set to 0 and is incremented by process P_i during its turn to initiate the recovery algorithm. Thus, a process on its own determines if it is its turn to initiate the execution of the algorithm. In this context, observe that the set of GCCs is unique and is independent of the initiator process. We state below how a process P_i does it before we formally state the algorithm:

Selection of an initiator process:
At each process P_i ($0 \leq i \leq n-1$):
If $CLK_i = (i+(counteri*n))*T$
counteri= counteri+1;
/*When its turn to initiate the recovery algorithm, i.e., P_i becomes the initiator*/

Algorithm Recovery – Enhanced:
Input: Given the latest n checkpoints, one for each process P_j , $0 \leq j \leq n-1$, for an n process system and the corresponding vectors V_j and lists R_j at these n checkpoints.
Output: A set of globally consistent checkpoints (maximum consistent state of the system).

The responsibilities of the initiator process P_i and each participating process P_j are stated in Fig. 6.

An example: Consider the system as shown in Fig. 7. Ignore the presence of the failure 'f' for the time being. Suppose that the periodic execution of algorithm starts immediately after processes P_1 and P_3 take their respective checkpoints $C_{1,5}$ and $C_{3,2}$. The algorithm determines the latest consistent global checkpoint of the system. It is $\{C_{1,2}, C_{2,1}, C_{3,1}\}$.

The two rules are applied to update the lists R_1, R_2, and R_3, and the vectors V_1, V_2, and V_3 at the checkpoints of processes P_1, P_2, and P_3 starting from their respective latest globally consistent checkpoints, which are namely $C_{1,2}$, $C_{2,1}$, $C_{3,1}$. The system with the updated lists and vectors is shown in Fig. 8. The checkpoints shown in Fig. 8 are the only ones saved in stable storage.

Now assume that a failure 'f' has occurred. Therefore the algorithm determines the consistent global checkpoint of the system, which is $\{C_{1,2}, C_{2,1}, C_{3,1}\}$ and applies only Rule 1 to reset the vectors to zero and to make the lists empty at the respective GCCs of the three processes.

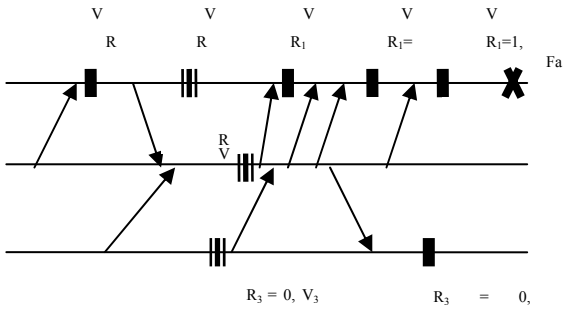


Figure 7: Before the execution of the algorithm

The system in this situation is shown in Fig. 9. The three respective consistent checkpoints are the only ones saved in the stable storage at this time.

Note that the consistent global state remains the same (see Figs. 8 and 9). This is the situation when time to complete the application program increases by an amount equal to the time to execute the recovery algorithm. This has been pointed out earlier in the description of Approach 1. However, this will not happen if only Approach 2 is followed for recovery.

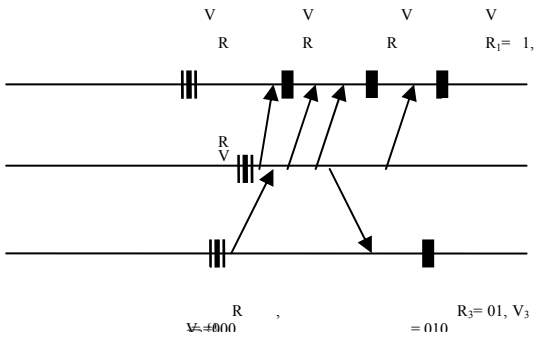


Figure 8: After the execution of the algorithm in absence of any failure

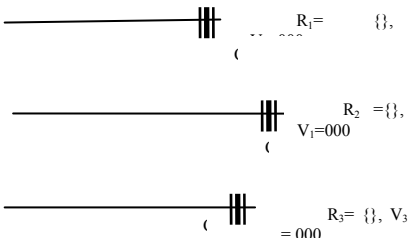


Figure 9: The system restarts from its consistent global state $\{C_{1,2},C_{2,1},C_{3,1}\}$ after recovery.

7.1 Comparison with [11] and [13]

Gupta et al. [11] have proposed a roll-forward hybrid checkpointing / recovery scheme using basic checkpoints. The direct dependency concept used in the communication-induced checkpointing scheme has been applied to basic checkpoints to design a simple algorithm to find a consistent global checkpoint. They have used the concept of forced checkpoints that ensures a small re-execution time after recovery from a failure. This scheme has the advantages of simple recovery as in synchronous approach and simple way to create checkpoints like in asynchronous approach.

Our proposed approach (enhanced version) is not a hybrid approach. It runs periodically only to put a limit on the size of the R-vectors. This is the primary objective of the enhanced approach. In doing so it may come out with a recent recovery line that is different from the one found during the last execution of the algorithm. Thus, effectively as mentioned earlier, even though the proposed algorithm is not a hybrid one, still as in [11] it may reduce drastically the number of comparisons needed to identify a recovery line, as well as it may limit the domino effect by the time period T, based on the message communication pattern among the processes.

Our proposed approach is quite different from the work in [13] in that in our approach processes take checkpoints completely independently based on their individual time periods that are different for different processes. In [13], processes take checkpoints with the same time periods and they make sure that there is no orphan message between any two i^{th} checkpoints of two processes. Therefore, it is more of a synchronous approach than an asynchronous approach, where as our approach is purely an asynchronous approach.

8 Conclusions

In this paper we have presented an efficient recovery algorithm for distributed systems. Asynchronous checkpointing scheme has been considered because of its simplicity in taking checkpoints. The main feature of the recovery algorithm is that to determine a maximum consistent state, the algorithm in its each iteration does not need to compare all the vectors at all the checkpoints of the processes. In its each iteration the algorithm identifies and skips those checkpoints that can not belong to the set of the globally consistent checkpoints. It not only reduces the computational overhead to a good extent, but also the number of trips to the stable storage for fetching checkpoints is reduced compared to the works in [14] and [16], and as a result its execution becomes even faster. In this context, it may be noted that in any algorithm that uses asynchronous checkpointing, there is always some computational time wasted to create process checkpoints that later do not belong to CGS and this problem can not be avoided. This is true for our proposed algorithms as well. Besides, it is executed simultaneously by all participating processes while determining a maximum consistent state. It further contributes to its speed of execution. We have also proposed a simple enhanced asynchronous recovery scheme to control the dynamically growing length of the lists. In effect, the number of comparisons of the checkpoints to determine a recent consistent state may also drastically reduce and based on the communication pattern among the processes it may limit the domino effect by the time period T. Even though we do not apply any hybrid checkpointing scheme [11], still this approach offers the option to achieve a recovery scheme which is as simple as the approach proposed in [11]. In this context, it may be noted that if the system model changes such that order of the messages sent through the channel cannot be preserved, it will adversely affect the processing time, because a process must wait to receive message m_1 before processing its already received message m_2. Here, we have assumed that the proper order is m_1 followed by m_2.

Our future work is directed at the new challenging area of designing recovery schemes for cluster federation computing environment in which different clusters may adopt different ways for checkpointing, for example, some may apply coordinated approach, where as other may apply asynchronous approach [18], [19].

```
Initiator process Pi:
Step 1:     if the algorithm is executed on failure
                recovery_on_failure = 1;
            else  recovery_on_failure = 0;
            It asks every process Pj (j ≠ i) to send its Vj corresponding to its latest checkpoint Cj,r;

Step 2:     It receives all Vj for 0 ≤ j ≤ n-1;
Step 3:     It computes Vc = vc^0 vc^1 … vc^j … vc^n-1;
Step 4:     It unicasts vc^j to each Pj, for j ≠ i;
Step 5:     It computes Di by calculating (Ri(r) - vc^i);
            if Di> 0
                It searches the list Ri till it finds the largest integer m (< r) that satisfies Ri(r) – Ri(m) ≥ Di. Then it
                sets its flag to 1 and considers Vi corresponding to its checkpoint Ci, m (i.e. Ci,r is replaced by Ci,m ) for
                the next iteration;

                                                /* Checkpoints Ci,r, Ci,r-1, …., Ci, m+1 are skipped */
            else
                It sets its flag to 0 and considers Vi at Ci, r for the next iteration;
Step 6:     It receives the flag and Vj from each process Pj;
            if flag = 0 for each process Pj , 0 ≤ j ≤ n-1

            /*Globally consistent checkpoints belonging to the maximum consistent state are determined */

                if recovery_on_failure = 1
                   Pi asks each process Pj to restart the application program from its last checkpoint
                   corresponding to which Dj ;
                   Pi implements Rule 1 corresponding to its restarting checkpoint at which Di ≤ 0;
                   It restarts computation from the restarting checkpoint (its GCC);
                else
                   Pi asks each process Pj to continue its normal computation from its latest checkpoint;
                   Pi implements Rule 2 followed by Rule 1;  /* Periodic determination of GCCs is done */
                   It continues its normal computation from its latest checkpoint;
            else
                Control flows to Step 3;
-------------------------------------------------------------------------------------------------------------------------
Process Pj:
Step 1:     Pj receives request from Pi;

                if Pi  has requested to restart
                                                /* The system restarts after recovery from a failure*/

                Pj implements Rule 1 corresponding to its restarting checkpoint at which Dj ≤ 0;
                It restarts computation from the restarting checkpoint (its GCC);
            else if
                Pi has requested to continue with the application program
                Pj implements Rule 2 followed by Rule 1;
                                                /* Periodic determination of GCC is done */
                It continues its normal computation from its latest checkpoint;
            else
                It sends Vj corresponding to its latest checkpoint Cj,r to the initiator process Pi;
Step 2:     It receives vc^j from Pi;
Step 3:     It computes Dj by calculating (Rj(r) - vc^j);
Step 4:     if Dj > 0
                It searches the list Rj till it finds the largest integer m (< r) that satisfies Rj(r) - Rj(m) ≥Dj. Then it
                sends a flag of 1 and Vj to Pi corresponding to its checkpoint Cj, m (i.e. Cj,r is  replaced by Cj,m );

                                                /* Checkpoints Cj,r, Cj,r-1, …., Cj, m+1 are skipped */
            else
                It sends a flag of 0 and Vj at Cj,r to the initiator process Pi;
```

Figure 6: The responsibilities of the initiator process P_i and each participating process P_j for the enhanced algorithm.

9 References

[1] R. Koo and S. Toueg, "Checkpointing and Rollback-Recovery for Distributed Systems", IEEE trans. Software Engineering, vol. SE-13, no. 1, pp. 23-31, Jan 1987.

[2] Y. M. Wang, A. Lowry, and W. K. Fuchs, "Consistent Global Checkpoints Based on Direct Dependency Tracking", Information Processing Letters, vol. 50, no. 4, pp. 223-230, May 1994.

[3] K. M. Chandy and L. Lamport, "Distributed Snapshots: Determining Global States of Distributed Systems", ACM Trans. Computing Systems, vol.3, no. 1, pp. 63-75, Feb. 1985.

[4] Y. Wang, "Consistent Global Checkpoints that Contain a Given Set of Local Checkpoints", IEEE Trans. Computers, vol. 46, no. 4, pp. 456-468, April 1997.

[5] M. Singhal and N. G. Shivaratri, Advanced Concepts in Operating Systems, McGraw-Hill, 1994.

[6] S. Venkatesan, T. Juang, and S. Alagar, "Optimistic Crash Recovery Without Changing Application Messages", IEEE Trans. Parallel and Distributed Systems, vol. 8, no.3, pp. 263-271, March 1997.

[7] R. Baldoni, F. Quaglai, and P. Fornara, "An Index-Based Checkpointing Algorithm for Autonomous Distributed Systems", IEEE Trans. Parallel and Distributed System, vol. 10, no.2, pp.181-192, Feb. 1999.

[8] J. Tsai, S. -Y. Kuo, and Y. -M. Wang, "Theoretical Analysis for Communication-Induced Checkpointing Protocols with Rollback Dependency Trackability", IEEE Trans. Parallel and Distributed Systems, vol. 9, no. 10, pp. 963-971, Oct. 1998.

[9] G. Cao and M. Singhal, " On Coordinated Checkpointing in Distributed Systems, IEEE Trans. Parallel and Distributed Systems ", vol. 9, no.12, pp. 1213-1225, Dec.1998.

[10] P. Jalote, Fault Tolerance in Distributed Systems, PTR Prentice Hall, (1994), Addison-Wesley, (1998).

[11] B. Gupta, S.K. Banerjee and B. Liu, "Design of new roll-forward recovery approach for distributed systems", IEE Proc. Computers and Digital Techniques, Volume 149, Issue 3, pp. 105-112, May 2002.

[12] D. Manivannan and M. Singhal, "Quasi-synchronous checkpointing: Models, characterization, and classification", Parallel and Distributed Systems, IEEE Transactions on Volume 10, Issue 7, pp. 703– 713, July 1999.

[13] D. Manivannan, and M. Singhal, "Asynchronous recovery without using vector timestamps", Journal of Parallel and Distributed Computing, Volume 62, Issue 62 pp. 1695-1728, Dec 2002.

[14] M. Ohara., M. Arai., S. Fukumoto., and K. Iwasaki.,"Finding a Recovery Line in Uncoordinated Checkpointing", Proceedings 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04), pp. 628 – 633, 2004.

[15] R. H. B. Netzer and J. Xu, "Necessary and Sufficient Conditions for Consistent Global Snapshots", IEEE Trans. Parallel and Distributed Systems, vol. 6, no. 2, pp. 165-169, Feb. 1995.

[16] T. Juang and S. Venkatesan, "Crash Recovery with Little Overhead", Proc. 11th International Conference on Distributed Computing Systems, pp. 454-461, May 1991.

[17] B. Gupta, Y. Yang, S. Rahimi, and A. Vemuri, "A High-Performance Recovery Algorithm for Distributed Systems", Proc. 21st International Conference on Computers and Their Applications, pp. 283-288, Seattle, March 2006.

[18] S. Monnet, C. Morin, R. Badrinath, "Hybrid Checkpointing for Parallel Applications in cluster Federations", Proc. 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, IL, USA, pp. 773-782, April 2004.

[19] J. Cao, Y. Chen, K. Zhang and Y. He, "Checkpointing in Hybrid Distributed Systems", Proc.7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04), pp. 136-141, May 2004.

Discovering Hidden Knowledge from Biomedical Literature

Ingrid Petrič[1], Tanja Urbančič[1,2] and Bojan Cestnik[2,3]

[1]University of Nova Gorica
Vipavska 13, 5000 Nova Gorica, Slovenia

[2]Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia

[3]Temida, d.o.o.
Dunajska 51, 1000 Ljubljana, Slovenia
E-mail: ingrid.petric@p-ng.si, tanja.urbancic@p-ng.si, bojan.cestnik@temida.si

Keywords: text mining, ontology construction, autism

Received: June 3, 2006

In this paper we investigate the potential of text mining for discovering implicit knowledge in biomedical literature. Based on Swanson's suggestion for hypotheses generation we tried to identify potential contributions to a better understanding of autism focusing on articles from database PubMed Central. First, we used them for ontology construction in order to obtain an improved insight into the domain structure. Next, we extracted a few rare terms that could potentially lead to new knowledge discovery for the explanation of the autism phenomena. We present a concrete example of such constructed knowledge about a substance calcineurin and its potential relations with other already published indications of autism.

Povzetek: Prispevek opisuje uporabo metod rudarjenja besedil na medicinskih člankih s področja avtizma.

1 Introduction

The practice of biomedicine is, as well as other activities of our society, inherently an information-management task (Shortliffe, 1993). Internet, the very common and increasingly used information source, provides massive heterogeneous collections of data. Huge bibliographic databases thus often contain interesting information that may be inexplicit or even hidden. One of such databases is MEDLINE, the primary component of PubMed, which is the United States National Library of Medicine's bibliographic database. It covers over 4.800 journals published in more than 70 countries worldwide and thus contains over 14 million citations from 1966 to the present (PubMed, 2006). The daily increasing number of biomedical articles provides a huge potential source of new data. In MEDLINE database there are between 1.500-3.500 complete references added since 2002 each day from Tuesday to Saturday (PubMed, 2006).

There is an urgent need to assist researchers in extracting knowledge from the rapidly growing volumes of databases in order to improve the usefulness of these vast amounts of data. For such reasons, the ability to extract the right information of interest remains the subject of the growing field of knowledge discovery in databases. Knowledge discovery is the process of discovering useful knowledge from data, which includes data mining as the application of specific algorithms for extracting patterns from data (Fayyad et al., 1996). In fact, important information hidden in huge databases could be discovered by data mining and knowledge discovery techniques. More specifically, those databases that contain bibliographic semi-structured data can be approached by text mining as specific kind of data mining.

Although the technology for data and text mining is well advanced, its potential still seems to lack sufficient recognition. Healthcare in general is one of the slowest sectors in utilizing information and communication technologies to their full benefit; however, the need for computer literacy has already been recognised and acknowledged by professionals in this sector (Štepankova, Engova, 2006). Therefore, one of the major challenges of biomedical text mining over the next 5 to 10 years is to make these techniques better understood and more useful to biomedical researchers (Cohen, Hersh, 2005). At the same time, the continued cooperation with professional communities such as the biomedical research community is required to ensure that their needs are properly addressed. Such collaboration is particularly crucial in complex scientific areas, as for example in autism field of biomedical research. The specific requirements in autism research, as presented by Zerhouni (2004), actually emphasise the need for

increasing the efficiency of communication of research findings to the related science community.

Autism belongs to a group of pervasive developmental disorders that in most cases have unclear origin. The main characteristic components of abnormal functioning in autism are the early delay and abnormal development of cognitive, communication and social interaction skills of affected individuals. In the fourth, revised edition of Diagnostic and Statistical Manual of Mental Disorders, a category of pervasive developmental disorders refers to a group of symptoms of neurological development, connected with early brain mechanisms that in large extent condition the social abilities already in the childhood (American Psychiatric Association, 2000). Such heterogeneous features of autistic developmental disturbance and its different degrees of affecting children have led to contemporary naming of autism conditions with the term: *Autism spectrum disorders*, to which suits the abbreviation *ASD*. The lack of studies, evidenced by Zerhouni (2004), that would increase the knowledge about risk factors and early development of autism, and that would better define characterization of autism spectrum disorders, has led us to choose the autism as an application domain of our research in knowledge technologies.

In this article we focus on the areas and methods where text mining potentially enriches biomedical science and thus interdisciplinary connects information technologies with biomedical expert knowledge. First we describe several text mining approaches in real biomedical settings towards extracting knowledge from data. Then we present our approach towards integration of real problem analysis and extraction of potentially useful information from data. Our main aim was to extract some implicit and previously unknown interesting information from professional articles about autism. Some of our text mining results are finally described with example pairs of implicit connections that we managed to identify from biomedical articles.

2 Text mining in biomedicine

There are several biomedical examples, where data mining has been successfully applied, as described in a review by Van Someren and Urbančič (2006). Examples include diagnosis, where data mining relates symptoms and other attributes of patients to their disease, subgroups of patients that are at risk for certain disease, and gene expression, with a growing number of applications, where predictions and identifications of disease markers are made, based on features of genes.

While data mining usually operates with collections of well structured data, researchers often have to deal with semi-structured text collections, too. Such datasets require the use of text mining techniques. Extracting important information from the increasingly available biomedical knowledge represented in digital text forms, has been proved as an important opportunity for biomedical discoveries and hypothesis generation. Having access and ability to work with the newest information, indeed means great potential for experts, who can benefit from the advantages of information systems and technologies. Biomedical informatics thus presents an essential element of biomedical research process. Methods that have been recently used for biomedical text mining tasks include the following items (Cohen, Hersh, 2005):

- *Named entity recognition* in order to identify all of the instances of a name for specific type of domain, within a collection of text;
  Examples of recent areas of biomedical research:
  - drug names within published journal articles,
  - gene names and their symbols within a collection of MEDLINE abstracts.

  Text mining approaches: lexicon-based, rules-based, statistically based, combined.
- *Text classification* with the goal to automatically determine whether a document or a part of it has particular attributes of interest;
  Examples of recent areas of biomedical research:
  - documents discussing a given topic,
  - texts containing a certain type of information.

  Text mining approaches: classification rule induction.
- *Synonym and abbreviation extraction* with the attempt to speed up literature search with automatic collections of synonyms and abbreviations for entities;
  Examples of recent areas of biomedical research:
  - gene name synonyms,
  - biomedical term abbreviations.

  Text mining approaches: combination of named entity recognition system, with statistical, support vector machine classifier-based, and automatic or manual pattern-based matching rules algorithms.
- *Relationship extraction* with the goal to recognize occurrences of a pre-specified type of relationship between a pair of entities of specific types;
  Examples of recent areas of biomedical research:
  - relationships between genes and proteins,
  - text-based gene clustering.

  Text mining approaches: neighbour divergence analysis, vector space approach and k-medoids clustering algorithm, fuzzy set theory on co-occurring dataset records, type and part-of-speech tagging.
- *Integration frameworks* with intention to address many different user needs;
  Examples of recent areas of biomedical research:
  - comparison of gene names and functional terms,
  - gene based text clusters.

  Text mining approaches: template-based, text profiling and clustering based.
- *Hypothesis generation* that focuses on the uncovering of implicit relationships, worthy of further investigation, that are inferred by the presence of other more explicit information;
  Examples of recent areas of biomedical research:

- connection between patient benefit and food substances,
- potential new uses and therapeutic effects of drugs.

Text mining approaches: Swanson's ABC model-based.

In the continuation we concentrate on hypotheses generation as a central point of our research interest.

3 Related work

The machine learning process is characterized by the search space, which reflects the expression of the hypothesis language, as a target knowledge (Botta et al., 2003). Idea of the text mining approach towards hypothesis generation, known as Swanson's ABC model, consists of discovering complementary structures in disjoint journal articles. This model assumes that when one literature reports that agent A causes phenomenon B, and second literature reports that B influences C, we could propose that agent A might influence phenomenon C (Swanson, 1990). To find some published evidence leading to undiscovered knowledge, the A and C literatures should have few or no published articles in common. In such way, Swanson discovered, among other, several relationships that connected migraine and decreased levels of magnesium (Swanson, 1990).

To facilitate the discovery of hypotheses by linking findings across literature, Swanson and his colleagues designed a set of interactive software that is available on a web-based system called Arrowsmith (Smalheiser, Swanson, 1998). Pratt and Yetisgen-Yildiz (2003) designed LitLinker that uses data mining techniques to identify correlations among concepts and then uses those correlations for discovery of potential causal links between biomedical terms. Sehgal et al. presented a system that may be used to explore topics and their relationships using text collections such as MEDLINE (Sehgal et al., 2003). Weeber et al. experimented with Swanson's idea of searching the literature for generating new potential therapeutic uses of the drug thalidomide with the use of a concept-based discovery support system DAD on the scientific literature (Weeber et al., 2003). Another example of discovering new relations from bibliographic database according to Swanson's model is identification of disease candidate genes by an interactive discovery support system for biomedicine Bitola (Hristovski et al., 2005). Transitive text mining was explored also by Grohmann and Stegmann (2005), who developed a web-based tool, C-MLink.

For successful data mining a wide background knowledge concerning the problem domain presents a substantial advantage. In fact, hypothesis generation from text mining results relies on background knowledge, experience, and intuition (Srinivasan, 2004). With this consideration we started our examination of autism phenomena with the identification of its main concepts and the review of what is already known about autism. We identified such information by ontologies construction, which we found a very fast and effective way of exploring large datasets. Ontologies in general with their capability to share a common understanding of domains support researches with ability to reason over and to analyse the information at issue (Joshi, Undercoffer, 2004). Many tools that help constructing ontologies from texts were developed and successfully used in practice (Brank et al., 2005). Among them, OntoGen (Fortuna et al., 2006), the interactive tool for semi-automatic construction of ontologies, received a remarkable attention.

4 Identification of domain structure

An important goal in our recognition of autism phenomena was to uncover the fundamental concepts that provide the common knowledge about autism. To identify some useful pieces of knowledge from the large amount of digital articles one approach would be to read and manually analyse all available data. Since this is evidently a time consuming task, we instead chose to guide our attention only on the most relevant information about the domain of interest. We performed our research with the computational support of OntoGen.

4.1 Target dataset

We decided to analyse the professional literature about autism that is publicly accessible on the World Wide Web in the database of biomedical publications, PubMed. In the PubMed database we found 10.821 documents (till August 21, 2006) that contain derived forms of autis*, the expression root for autism. There were 354 articles with their entire text published in the PubMed Central database. Other relevant publications were either restricted to abstracts of documents or their entire texts were published in sources outside PubMed. From the listed 354 articles we further restricted the target set of articles on documents to those that have been published in the last ten years. As a result, we got 214 articles from 1997 forward, which we decomposed to titles, abstracts and texts for the purpose of further analysis.

4.2 Text mining support system

One of the most frequently used text representations in text mining is word-vector representation, where the word-vector contains some weight for each word of text, proportional to the number of its occurrences in the text (Mladenić, 2006). Such representations are used also by OntoGen, which enables interactive construction of ontologies in a chosen domain. We used it to construct several autism ontologies. The input for the tool is a collection of text documents. With machine learning techniques OntoGen supports important phases of ontology construction by suggesting concepts and their names, by defining relations between them, and by automatic assignment of documents to the concepts (Fortuna, 2005).

4.3 Ontology of autism domain

Our aim was first to review the autism literature and to identify the most frequent topics researched in this domain. With this intention we built the autism domain ontology with OntoGen on 214 articles from PubMed Central database that treat problems of autism. OntoGen displayed sub-concepts of autism domain as suggested by its clustering algorithm, and described them with their main keywords extracted from text documents. The keywords that we used for concepts description were calculated both according to the concept centroid vector, and by the Support Vector Machine based linear model. The system also displayed the current coverage of each concept by the number of documents that it positively classified into the concept and the inner-cluster similarity measures. Ontologies built with OntoGen, as an example shown in Figure 1, actually helped us to substantially speed up the process of reviewing and understanding the complex and heterogeneous spectrum of scientific articles about autism.

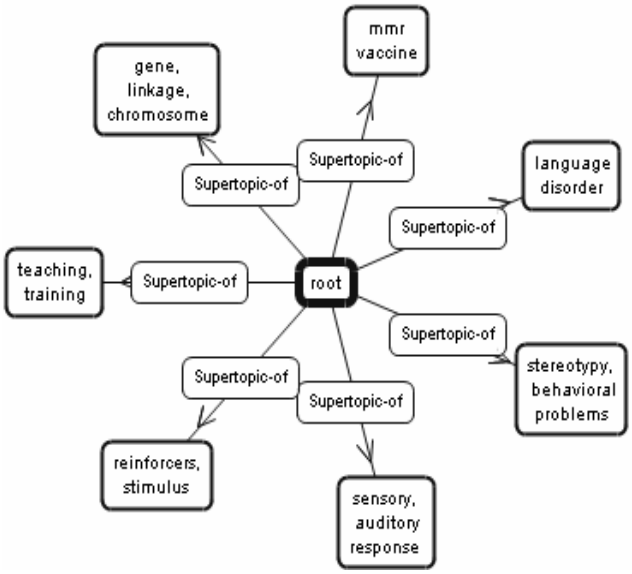


Figure 1: Concepts of autism ontology with 7 subgroups, built on 214 abstracts from the PubMed Central database.

The main concepts of autism phenomena as they result from the first level of our ontology model (first level subgroups of autism domain) are: genetics; teaching and training; reinforcers and stimulus; sensory and auditory response; stereotypy and behavioural problems; language disorders, and MMR (Measles, Mumps, and Rubella) vaccine. Important confirmation of the resulted ontology construction is the recent state of autism research as described by Zerhouni (2004) that summarizes the main scientific activities of autism research in the major areas of epidemiology, genetics, neurobiology, environmental factors and specific treatments of autism.

5 Extraction of implicit relationships from autism data

Besides constructing an ontology on the input file of texts, OntoGen creates also a *.txt.stat file with statistical incidence of terms as they appear in documents collected in the input dataset. We utilized this OntoGen's by-product as the basis for our approach toward the identification of rare relations between autism data. As our goal was to discover undocumented knowledge about autism phenomena, we assumed that starting our search on rare connections between data rather than on frequent ones, we would have better chances to discover implicit relations that are still unknown and might, however, be useful for the autism researchers.

5.1 The related concepts

Our approach towards discovering knowledge about autism concentrated on identifying interesting concepts within autism sub-areas of interest. Therefore, we considered the subdivision of autism domain on research fields; moreover, we particularly guided our attention on neurobiological basis of autistic abnormalities.

To find some related concepts, which would lead us to potential discoveries of new knowledge, we took the *.txt.stat file created by OntoGen while constructing ontologies. We first focused our attention on those terms listed in this text file that appeared only in one article from the input dataset. Taking into account also background knowledge about autism, we chose words that could be useful for autism discovery. Three of the chosen terms, presented also in the intersection area in Figure 2, are: lactoylglutathione, synaptophysin and calcium_channels. There are three major reasons for these choices. First, we found that an increase in polarity of glyoxalase I in autism brains was reported and that glyoxalase system involves also lactoylglutathione. Second, as the altered synaptic function was also discussed in autism articles, we took in consideration synaptophysin, a protein localized to synaptic vesicles. And third, abnormal calcium signalling was found in some autistic children, thus we chose also term calcium_channels for further discovery. After selecting these three terms of interest, we searched the article database to find what all these terms have in common.

One of the goals of text mining is to automatically discover interesting hypotheses from a potentially useful text collection (Srinivasan, 2004). By text mining on PubMed articles that treat these selected terms domains, we constructed their ontologies and from the OntoGen's *.txt.stat files we retrieved the words they all have in common (the words that appeared in the three *.txt.stat files). One of such terms, listed also in Figure 2, that could be interesting for the hypothesis generation and forward research on autism phenomena, is calcineurin. Calcineurin is calcium- and calmodulin-dependent serine/threonine protein phosphatase, which is widely present in mammalian tissues, with the highest levels found in brain (Rusnak, Mertz, 2000). Our literature mining in disjoint journal articles showed that it could be

related to autistic disorders, however to the present no direct evidence of calcineurin role in autism has been reported yet on the internet.

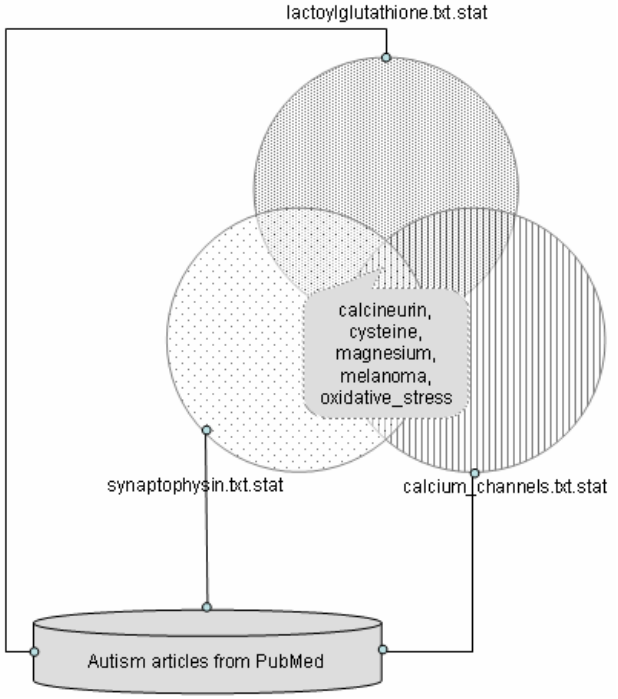


Figure 2: Results of our approach to literature mining on autism domain.

5.2 The explored conjectures

In order to justify the role of calcineurin in autistic problem domain we decided to search and explore possible reasoning paths that relate the selected substance to some known expressions of autism. Since the direct relation was not yet noted in the literature, our goal was to find a few plausible interconnecting terms that relate the two notions (Swanson, 1990). Having this in mind, we explored the union of PubMed articles about autism and articles about calcineurin. By building ontologies on such input dataset of combined articles the goal was to discover documents having as much as possible words in common. For this purpose we searched for the highest similarity measures inside the clusters of ontologies. Interestingly, by this search we identified several pairs of instances of PubMed articles that are connecting the two categories of biomedical literature, autism and calcineurin, respectively. This way we were able to find eleven pairs of articles, which, when put together, could be seen as arguments for new hypotheses of autism and calcineurin relationship, such as the three listed in Table 1.

When showing the presented results to the expert of autistic spectrum and related disorders, she not only confirmed strong interest in the method and in the discovered relations, but was also able to guide our further work very efficiently by turning our attention on discovering the relationship between autism and fragile X chromosome.

Autism literature	Calcineurin literature
Fatemi et al. (2001) reported a reduction of Bcl-2 (a regulatory protein for control of programmed brain cell death) levels in autistic cerebellum.	Erin et al. (2003) observed that calcineurin occurred as a complex with Bcl-2 in various regions of rat and mouse brain.
Qiu et al. (2006) described the low-density lipoprotein receptors that regulate cholesterol transport, in neuropsychiatric disorders, such as autism.	Cofan et al. (2005) published their article about effect of calcineurin inhibitors on low-density lipoprotein oxidation.
Bear et al. (2004) reported about the loss of fragile X protein, an identified cause of autism that increased long-term depression in mouse hippocampus.	Zhabotinsky et al. (2006) described induction of long-term depression that depends on calcineurin.

Table 1: Hypotheses for calcineuring and autism relationship.

6 Conclusion

Our study confirms the potential of ontology construction by OntoGen on biomedical literature to systematically structure main concepts. The evaluation of the ontology constructed on autism showed important similarity to the reported state of autism research.

Considering OntoGen's statistical data can lead to discovery of potentially useful and previously unknown information related to the researched phenomena. In such way, OntoGen's functionality can be extended to retrieve new information from vast amounts of textual data that experts otherwise have to explore manually. As connecting sets of literature about synaptophysin, lactoylglutathione and calcium channels that were selected as three interesting rare terms from autism articles, we found calcineurin, cysteine, magnesium, melanoma, oxidative stress and many others. In the preliminary expert evaluation the approach proposed in this paper proved to be successful. However, further assessment of the possible role of calcineurin and other resulting candidates in autism is needed to justify our methodological approach and to see if it can contribute to the knowledge corpus of autism phenomena.

Acknowledgement

This work was partially supported by the Slovenian Research Agency programme Knowledge Technologies (2004-2008). We thank Nada Lavrač for her suggestion to use OntoGen and Blaž Fortuna for his discussions about OntoGen's performance. We also appreciate help and support we got from Marta Macedoni-Lukšič in our efforts to better understand autism.

References

[1] American Psychiatric Association (2000) Diagnostic and Statistical Manual of Mental Disorders, Fourth Edition, Text Revision. Washington, DC.

[2] Bear M.F., Huber K.M., Warren S.T. (2004) The mGluR theory of fragile X mental retardation, *Trends in Neurosciences*, 27(7), pp. 370-377.

[3] Botta M., Saitta L., Sebag M. (2003) Relational Learning as Search in a Critical Region, *Journal of Machine Learning Research*, 4, pp. 431-463.

[4] Brank J., Grobelnik M., Mladenić D. (2005) A survey of ontology evaluation techniques, *SIKDD 2005 at multiconference IS 2005*, Ljubljana, Slovenia.

[5] Cofan F., Cofan M., Campos B., Guerra R., Campistol J.M., Oppenheimer F. (2005) Effect of calcineurin inhibitors on low-density lipoprotein oxidation, *Transplantation Proceedings*, 37(9), pp. 3791-3793.

[6] Cohen A.M., Hersh W.R. (2005) A Survey of Current Work in Biomedical Text Mining, *Briefings in Bioinformatics*, 6(1), pp. 57-71.

[7] Erin N., Bronson S.K., Billingsley M.L. (2003) Calcium-dependent interaction of calcineurin with Bcl-2 in neuronal tissue, *Neuroscience*, 117(3), pp. 541-555.

[8] Fatemi S.H., Stary J.M., Halt A.R., Realmuto G.R. (2001) Dysregulation of Reelin and Bcl-2 proteins in autistic cerebellum, *Journal of Autism and Developmental Disorders*, 31(6), pp. 529-535.

[9] Fayyad U., Piatetsky-Shapiro G., Smyth P. (1996) Knowledge Discovery and Data Mining: Towards a Unifying Framework. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon.

[10] Fortuna B. (2006) [http://ontogen.ijs.si/index.html], OntoGen: Description.

[11] Fortuna B., Grobelnik M., Mladenić D. (2006) System for semi-automatic ontology construction. Demo at ESWC 2006, Budva, Montenegro.

[12] Grohmann G., Stegmann J., C-MLink: a web-based tool for transitive text mining, *Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics*, Sweden, Stockholm, pp. 658-659

[13] Hristovski D., Peterlin B., Mitchell J.A., Humphrey S.M. (2005) Using literature-based discovery to identify disease candidate genes, *International Journal of Medical Informatics*, 74, pp. 289-298.

[14] Joshi A., Undercoffer J.L. (2004) On Data Mining, Semantics, and Intrusion Detection. What to Dig for and Where to Find It, *Data mining. Next Generation Challenges and Future Directions*, Menlo Park, California, pp. 437-460.

[15] Mladenić D. (2006) Text Mining: Machine Learning on Documents, *Encyclopedia of Data Warehousing and Mining*, Hershey: Idea Group Reference, pp. 1109-1112.

[16] Pratt W., Yetisgen-Yildiz M. (2003) LitLinker: Capturing Connections across the Biomedical Literature, *Proceedings of the International Conference on Knowledge Capture (K-Cap'03), Florida*, pp. 105–112.

[17] PubMed (2006) [http://www.ncbi.nlm.nih.gov/], Overview.

[18] Qiu S., Korwek K.M., Weeber E.J. (2006) A fresh look at an ancient receptor family: emerging roles for density lipoprotein receptors in synaptic plasticity and memory formation, *Neurobiology of Learning and Memory*, 85(1), pp. 16-29.

[19] Rusnak F., Mertz P. (2000) Calcineurin: Form and Function, *Physiological Reviews*, 80(4), pp. 1483-1521.

[20] Sehgal A., Qiu X.Y., Srinivasan P. (2003) Mining MEDLINE Metadata to Explore Genes and their Connections, *Proceedings of the SIGIR 2003 Workshop on Text Analysis and Search for Bioinformatics*.

[21] Shortliffe E.H. (1993) The Adolescence of AI in Medicine: Will the Field Come of Age in the '90s? *Artificial Intelligence in Medicine*, 5(2), pp. 93-106.

[22] Smalheiser N.R., Swanson D.R. (1998) Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses, *Computer Methods and Programs in Biomedicine*, 57, pp. 149-153.

[23] Srinivasan P. (2004) Text mining: Generating hypotheses from MEDLINE, *Journal of the American Society for Information Science and Technology*, 55(5), pp. 396-413.

[24] Swanson D.R. (1990) Medical literature as a potential source of new knowledge, *Bulletin of the Medical Library Association*, 78(1), pp. 29-37.

[25] Štepankova O., Engova D. (2006) Professional Competence and Computer Literacy in e-age, Focus on Healthcare, *Methods of Information in Medicine*; 45, pp. 300-305.

[26] Van Someren M., Urbančič T. (2006) Applications of machine learning: matching problems to tasks and methods, *The Knowledge Engineering Review*, 20(4), pp. 363-402.

[27] Weeber M., Vos R., Klein H., De Jong-van den Berg L.T., Aronson A.R., Molema G. (2003) Generating Hypotheses by Discovering Implicit Associations in the Literature: A case Report of a Search for New Potential Therapeutic Uses for Thalidomide, *Journal of the American Medical Informatics Association*, 10(3), pp. 252-259.

[28] Zerhouni E.A. for National Institutes of Health and National Institute of Mental Health (2004) Congressional Appropriations Committee Report on the State of Autism Research. Department of Health and Human Service, Bethesda, Maryland.

[29] Zhabotinsky A.M., Camp R.N., Epstein I.R., Lisman J.E. (2006) Role of the neurogranin concentrated in spines in the induction of long-term potentiation, *Journal of Neuroscience*, 26(28), pp. 7337-7347.

Approximate Representation of Textual Documents in the Concept Space

Jasminka Dobša
University of Zagreb, Faculty of Organization and Informatics
Pavlinska 2, 42 000 Varaždin, Croatia
jasminka.dobsa@foi.hr

Bojana Dalbelo Bašić
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10 000 Zagreb, Croatia
Bojana.Dalbelo@fer.hr

Keywords: dimensionality reduction, concept decomposition, information retrieval

Received: November 17, 2006

In this paper we deal with the problem of addition of new documents in collection when documents are represented in lower dimensional space by concept indexing. Concept indexing (CI) is a method of feature construction that is relying on concept decomposition of term-document matrix. By using CI original representations of documents are projected on the space spread by centroids of clusters, which are called concept vectors. This problem is especially interesting for application on World Wide Web. Proposed methods are tested for the task of information retrieval.
Vectors on which the projection is done in the process of dimension reduction are constructed on the basis of representations of all documents in the collection, and computation of the new representations in the space of reduced dimension demands recomputation of concept decomposition. The solution to this problem is the development of methods which will give approximate representation of newly added documents in the space of reduced dimension.
In the paper are introduced two methods for addition of new documents in the space of reduced dimension. In the first method there no addition of new index terms and added documents are represented by existing list of index terms, while in the second method list of index terms is extended and representations of documents and concept vectors are extended in dimensions of newly added terms. It is shown that representation of documents by extended list of index terms does not improve performance of information retrieval significantly.

Povzetek: Predstavljeni sta dve metodi konceptualnega indeksiranja dokumentov.

1 Introduction

In this paper we deal with the problem of addition of new documents in collection when documents are represented in lower dimensional space by concept indexing. This problem is especially interesting for application on World Wide Web. Proposed methods are tested for the task of information retrieval [1].

There are lots of motives for dimension reduction in the vector space model: decrease of memory space needed for representation of documents, faster performance of information retrieval or automatic classification of documents, reduction of noise and redundancy present in the representation of documents. Methods for dimension reduction in the vector space model based on extraction of new parameters for representation of documents (feature construction) tend to overcome the problem of synonyms and polysemies which are two major obstacles in information retrieval. Disadvantage of feature construction may be uninterpretability of newly obtained parameters or features.

Our investigation is based on the method of feature construction called *concept indexing* which was introduced in 2001 by Dhillon and Modha [7]. This method uses centroids of clusters created by the spherical k-means algorithm or so-called *concept decomposition* (CD) for lowering the rank of the term-document matrix. By using CI original representations of documents are projected on the space spread by centroids of clusters, which we call here *concept vectors*.

Representation of new document in the vector space model is trivial. The problem appears when we want to add new documents in the space of reduced dimension. Namely, vectors on which the projection is done in the process of dimension reduction are constructed on the basis of representations of all documents in the collection, and computation of the new representations in the space of reduced dimension demands recomputation of the concept decomposition. The solution to this problem is the development of methods which will give approximate representation of newly added documents in

the space of reduced dimension. Application of such a methods will delay a process of recomputation of concept decomposition.

Methods for addition of representations of new documents in the space of reduced dimension are already developed for LSI method [3,9]. The method of LSI was introduced in 1990 [4] and improved in 1995 [3]. Since then LSI is a benchmark in the field of dimension reduction. Although the LSI method has empirical success, it suffers from the lack of interpretation of newly obtained features which causes the lack of control for accomplishing specific tasks in information retrieval. Kolda and O'Leary [8] developed a method for addition of representations of new documents for LSI method that uses semi-discrete decomposition which saves memory space.

When the collection of documents is extended it seems natural to extend also the list of index terms with terms present in added documents, which were not present in starting collection of documents, or were present very rarely and they were not included in the list of the index terms. In the paper are introduced two methods for addition of new documents in the space spread by concept vectors, which is called *concept space.* In the first method there no addition of new index terms and added documents are represented by existing list of index terms, while in the second method list of index terms is extended and representations of documents and concept vectors are extended in dimensions of newly added terms.

This paper is organized as follows. Section 2 provides a description of technique of dimensionality reduction by concept decomposition. In Section 3 novel algorithms for approximate addition of documents in concept space are proposed. Section 4 provides an example, while Section 5 describes experiment where proposed algorithms are tested. Last section gives conclusions and directions for further work.

2 Dimensionality reduction by the concept decomposition

Let the $m \times n$ matrix $\mathbf{A} = [a_{ij}]$ be the term-document matrix. Then a_{ij} is the weight of the i-th term in the j-th document. A query has the same form as a document; it is a vector whose i-th component is the weight of the i-th term in the query. A common measure of similarity between the query and the document is the cosine of the angle between them.

Techniques of feature construction enable mapping documents' representations, which are similar in their content, or contain many index terms in common, to the new representations in the space of reduced dimension, which are closer than their representations in original vector space. That enables retrieving of documents which are relevant for the query, but do not contain index terms contained in the vector representation of query.

In this section we will describe the algorithm for computation of concept decomposition by the fuzzy k-mans algorithm [5].

2.1 Fuzzy k-means algorithm

The fuzzy k-means algorithm (FKM) [10] generalizes the hard k-means algorithm. The goal of the k-means algorithm is to cluster n objects (here documents) in k clusters and find k mean vectors or centroids for clusters. Here we will call these mean vectors *concept vectors*, because that is what they present. As opposed to the hard k-means algorithm, which allows a document to belong only to one cluster, FKM allows a document to partially belong to multiple clusters. FKM seeks a minimum of a heuristic global cost function $J_{fuzz} = \sum_{i=1}^{k}\sum_{j=1}^{n} \mu_{ij}^{b} \left\| \mathbf{a}_j - \mathbf{c}_i \right\|$ where $\mathbf{a}_j, j = 1,\ldots,n$ are vectors of documents, $\mathbf{c}_i, i = 1,\ldots,k$ are concept vectors, μ_{ij} is the fuzzy membership degree of document $\mathbf{a}_j$ in the cluster whose concept is $\mathbf{c}_i$ and b is a weight exponent of the fuzzy membership.

In general, the J_{fuzz} criterion is minimized when concept $\mathbf{c}_i$ is close to those documents that have a high fuzzy membership degree for cluster i , $i = 1,\ldots,k$. By solving a system of equations $\frac{\partial J_{fuzz}}{\partial \mathbf{c}_i}$ and $\frac{\partial J_{fuzz}}{\partial \mu_{ij}}$, we obtain a stationary point for which fuzzy membership degrees are given by

$$\mu_{ij} = \frac{1}{\sum_{r=1}^{k} \left(\frac{\left\| \mathbf{a}_j - \mathbf{c}_i \right\|^2}{\left\| \mathbf{a}_j - \mathbf{c}_r \right\|^2} \right)^{\frac{1}{b-1}}} \qquad (1)$$

for $i = 1,\ldots,k$ and $j = 1,\ldots,n$, while centroids or concept vectors are given by

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{b} \mathbf{a}_j}{\sum_{j=1}^{n} \mu_{ij}^{b}} \qquad (2)$$

for $i = 1,\ldots,k$. For such a stationary point the cost function reaches a local minimum. We will obtain concept vectors by starting with arbitrary initial concept vectors $\mathbf{c}_i^{(0)}, i = 1,\ldots,k$ and by computing fuzzy membership degrees $\mu_{ij}^{(t)}$, cost function $J_{fuzz}^{(t)}$ and new

concept vectors $\mathbf{c}_i^{(t+1)}$ iterative, where t is the index of iteration, until $\left|J_{fuzz}^{(t+1)} - J_{fuzz}^{(t)}\right| < \varepsilon$ for some threshold ε.

2.2 Concept decomposition

Our target is to approximate each document vector by a linear combination of concept vectors. The *concept matrix* is an $m \times k$ matrix whose j-th column is the concept vector $\mathbf{c}_j$, that is $\mathbf{C}_k = [\mathbf{c}_1, \ldots, \mathbf{c}_k]$. If we assume linear independence of the concept vectors, then it follows that the concept matrix has rank k. Now we define the *concept decomposition* $\mathbf{P}_k$ of the term-document matrix $\mathbf{A}$ as the least-squares approximation of $\mathbf{A}$ on the column space of the concept matrix $\mathbf{C}_k$. Concept decomposition is an $m \times n$ matrix $\mathbf{P}_k = \mathbf{C}_k \mathbf{Z}^*$ where $\mathbf{Z}^*$ is the solution of the least-squares problem, ie. $\mathbf{Z}^* = \left(\mathbf{C}_k^T \mathbf{C}_k\right)^{-1} \mathbf{C}_k^T \mathbf{A}$.

$\mathbf{Z}^*$ is a matrix of the type $k \times n$ and its columns are representations of documents in the concept space. Similarly, representation of query $\mathbf{q}$ in the reduced dimension space is given by $\left(\mathbf{C}_k^T \mathbf{C}_k\right)^{-1} \mathbf{C}_k^T \mathbf{q}$ and similarity between document and the query is given by the cosine of the angle between them. Concept indexing is a technique of indexing text documents by using concept decomposition.

3 Addition of representations of new documents in the concept space

In this section novel algorithms for addition text documents' representations in the concept space are proposed. The goal is to add new documents in a collection represented in the reduced dimension space, and this goal is achieved with and without an extension of the list of the index terms.

Let us introduce matrix notation that will be used in the section. Matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \tag{3}$$

will be an extended term-document matrix, where $\mathbf{A}_1$ a is matrix of starting documents in the space of starting terms, $\mathbf{A}_3$ is a matrix of starting documents in the space of added terms, $\mathbf{A}_2$ is a matrix of added documents in the space of starting terms and $\mathbf{A}_4$ is a matrix of added documents in the space of added terms. Further, let m_1 be number of starting terms, m_2 number of added terms, n_1 number of starting documents and n_2 number of added documents.

Here we will introduce two methods of approximate addition of new documents in the concept space:

(a) projection of new documents on existing concept vectors (Method A) and,
(b) projection of new documents on existing concept vectors extended in dimensions of newly added terms (Method B).

Assume that documents of a starting matrix $\mathbf{A}_1$ are clustered by fuzzy k-means algorithm and centroids of clusters are computed. Let $\mathbf{C}_1$ be the concept matrix the columns of which are concept vectors and let $\mathbf{C}_2$ be a matrix consisting of extensions of concept vectors in dimensions of added terms. Concept vectors of the matrix $\mathbf{C}_1$ are calculated by the formula (2) using columns of matrix $\mathbf{A}_1$ as document representations, while extensions of concept vectors are calculated by the same formula using respective columns of matrix $\mathbf{A}_3$ as representations of starting documents in the space of added terms. Let extensions of concept vectors form extension of the concept matrix denoted by $\mathbf{C}_2$. Then

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \tag{1}$$

is the concept matrix the columns of which are concept vectors extended in dimensions of newly added terms. Representations of documents in the concept space of extended term-document matrix will be given by expression

$$\begin{aligned}
&\left(\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}\right)^{-1} \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&= \left(\mathbf{C}_1^T \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{C}_2\right)^{-1} \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}^T \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&\approx \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \begin{bmatrix} \mathbf{C}_1^T & \mathbf{C}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&= \begin{bmatrix} \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_1^T & \vdots & \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} \\
&= [\left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_1^T \mathbf{A}_1 + \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_2^T \mathbf{A}_3 \\
&\vdots \quad \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_1^T \mathbf{A}_2 + \left(\mathbf{C}_1^T \mathbf{C}_1\right)^{-1} \mathbf{C}_2^T \mathbf{A}_4] \\
&= [(5)+(6) \quad \vdots \quad (7)+(8)]
\end{aligned} \tag{4}$$

In the third line of the expression (4) it is assumed approximation $\left(\mathbf{C}_1^T \mathbf{C}_1 + \mathbf{C}_2^T \mathbf{C}_2\right) \approx \mathbf{C}_1^T \mathbf{C}_1$. Such an approximation is justified by the fact that extensions of concept vectors are sparser than concept vectors formed from starting documents, because the coordinates of extended concept vectors are weights of added terms which were not included in list of the index terms before addition of new documents. It was established, by experiment, that $\left\|\mathbf{C}_2^T \mathbf{C}_2\right\|_2 << \left\|\mathbf{C}_1^T \mathbf{C}_1\right\|_2$. The number

of operations is significantly reduced by this approximation, because inverse $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}$ is already computed during the computation of starting documents projection.

This approximation is not necessary for the application of Method A, because this method does not use extensions of concept vectors. Representations of starting documents are given by expression (5), while representations of added documents are given by expression (7). Pre-processing of extended term-document matrix includes normalization of columns of matrices $\mathbf{A}_1$ (starting documents) and $\mathbf{A}_2$ (added documents) to the unit length. Let us now calculate number of operations needed for application of Method A. Representations of starting documents are already known, and so is matrix $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_1^T$. That is why the number of operations is equivalent to the number of operations needed for multiplication of matrices $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_1^T$ and $\mathbf{A}_2$, which is $2m_1kn_2$.

By the Method B added documents are projected on the space of extended concept vectors. Vector representations of starting documents are already known, and they are given by the (5), while representations of added documents are computed by the formula

$$\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_1^T\mathbf{A}_2 + \alpha\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}\mathbf{C}_2^T\mathbf{A}_4, \quad (9)$$

where coefficient $\alpha > 1$ has a role of stressing the importance of added terms and documents. Pre-processing of extended term-document matrix includes normalization of its columns to the unit length. Performance of Method B demands computation of concept vectors' extensions and computation of added documents projections. Computation of the first summand in formula (9) demands $2m_1kn_2$ operations, while computation of the second summand demands $(2k^2m_2+2m_2n_2k)$ operations, because inverse $\left(\mathbf{C}_1^T\mathbf{C}_1\right)^{-1}$ is already calculated. Addition of matrix elements of the first and second summand and multiplication by scalar in the formula (10) demands $2n_2k$ operations. Further, computation of concept vectors extensions by application of formula (2) demands $(2n_1km_2+2n_1k)$ operations. Normalization of columns of extended term-document matrix and concept matrix is not included in calculation of number of operations, because it is a standard operation of pre-processing included in every algorithm. That means that application of Method B demands

$$\begin{aligned} N_B &= 2m_1kn_2 + 2k^2m_1 + 2m_2n_2k + 2n_1km_2 + 2n_1k + 2n_2k \\ &= 2k(m_1n_2 + km_2 + m_2n_2 + n_1m_2 + n_1 + n_2) \end{aligned}$$

operations.

4 An example

By this example [6] it will be shown, in an illustrative way, how documents are projected by CI method into the two-dimensional concept space. The collection of 19 documents (titles of books) will be used where 15 documents will form collection of starting documents and 4 documents will form the collection of added documents. The documents are categorized in three categories: documents from the field of data mining (DM documents), documents from the field of linear algebra (LA documents) and documents which combine these two fields (application of linear algebra on data mining). The documents with their categorization are listed in Table 1. A list of terms is formed from words contained in at least two documents of starting collection, after which words on the stop list are ejected and variations of words are mapped on the same characteristic form (e.g. the terms *matrix* and *matrices* are mapped on the term *matrix*, or *applications* and *applied* are mapped on *application*). As a result, a list of 16 terms is obtained which we have divided in three parts: 8 terms from the field of data mining (*text, mining, clustering, classification, retrieval, information, document, data***)**, 5 terms from the field of linear algebra (*linear, algebra, matrix, vector, space***)** and 3 neutral terms (*analysis, application, algorithm***)**. Then we have created a term-document matrix from starting collection of documents and normalized the columns of it to be of the unit norm. This is a term-document matrix of starting documents in the space of starting terms $\mathbf{A}_1$. Then we have applied CD (k=2) to that matrix. In CD $\mathbf{C}_2\mathbf{Z}^*$ rows of concept matrix $\mathbf{C}_2$ are representations of terms and columns of $\mathbf{Z}^*$ are representations of documents of starting collection.

We have also created two queries (underlined words are from the list of terms):

1) Q1: Data mining
2) Q2: Using linear algebra for data mining.

For Q1 all data mining documents are relevant, while for Q2 documents D6, D18 and D19 are relevant. Most of the DM documents do not contain words *data* and *mining***.** Such documents will not be recognized by the simple term-matching vector space method as relevant. Documents D6 and D19, which are relevant for Q2, does not contain any of terms from the list contained in the query. The representation of the query $\mathbf{q}$ by concept indexing will be $\tilde{\mathbf{q}} = (\mathbf{C}_k^T\mathbf{C}_k)^{-1}\mathbf{C}_k^T\mathbf{q}$ and in the same way will be computed representations of added documents' collection (application of Method A).

Number	Status (Starting/Added)	Categorization	Document
D1	Starting	DM	Survey of text mining: clustering, classification, and retrieval
D2	Starting	DM	Automatic text processing: the transformation analysis and retrieval of information by computer
D3	Starting	LA	Elementary linear algebra: A matrix approach
D4	Starting	LA	Matrix algebra and its applications in statistics and econometrics
D5	Starting	DM	Effective databases for text and document management
D6	Starting	Combination	Matrices, vector spaces, and information retrieval
D7	Starting	LA	Matrix analysis and applied linear algebra
D8	Starting	LA	Topological vector spaces and algebras
D9	Starting	DM	Information retrieval: data structures and algorithms
D10	Starting	LA	Vector spaces and algebras for chemistry and physics
D11	Starting	DM	Classification, clustering and data analysis
D12	Starting	DM	Clustering of large data sets
D13	Starting	DM	Clustering algorithms
D14	Starting	DM	Document warehousing and text mining: techniques for improving business operations, marketing and sales
D15	Starting	DM	Data mining and knowledge discovery
D16	Added	DM	Concept decomposition of large sparse text data using clustering
D17	Added	LA	A rank-one reduction formula and its applications to matrix factorizations
D18	Added	Combination	Analysis of data matrices
D19	Added	Combination	A semi-discrete matrix decomposition for latent semantic indexing in information retrieval

Table 1: Documents and their categorization (DM – data mining documents, LA – linear algebra documents). Documents D6, D18 and D19 are combination of these two categories. Words from the list of terms are underlined.

In Figure 1 are shown images of representations of documents and queries in the concept space. It can be seen that LA documents of starting collection are grouped (and located near *x* axes); DM documents of starting collection are somewhat more dispersed, but generally also grouped around *y* axes, while D6 document (combination) is in the group of LA documents. It appears that way because during the clustering by fuzzy k-means algorithm D6 document was clustered to group of LA documents. Namely, fuzzy k-means algorithm allows documents to belong to multiple clusters partially during the process of clustering, but the result of convergence are hard partitions, which means that at the end algorithm decide in which cluster document belong.

Shaded areas on Figure 1 represent the areas of relevant documents for queries in the cosine similarity sense (cosine of the angle between points in shaded areas and representation of the queries is greater than 0.9). The added documents are shown on the figure in the shape that correspond to the category they belong, but in lighter colour then documents of the starting collection. By usage of the Method A document D16 (DM document) is

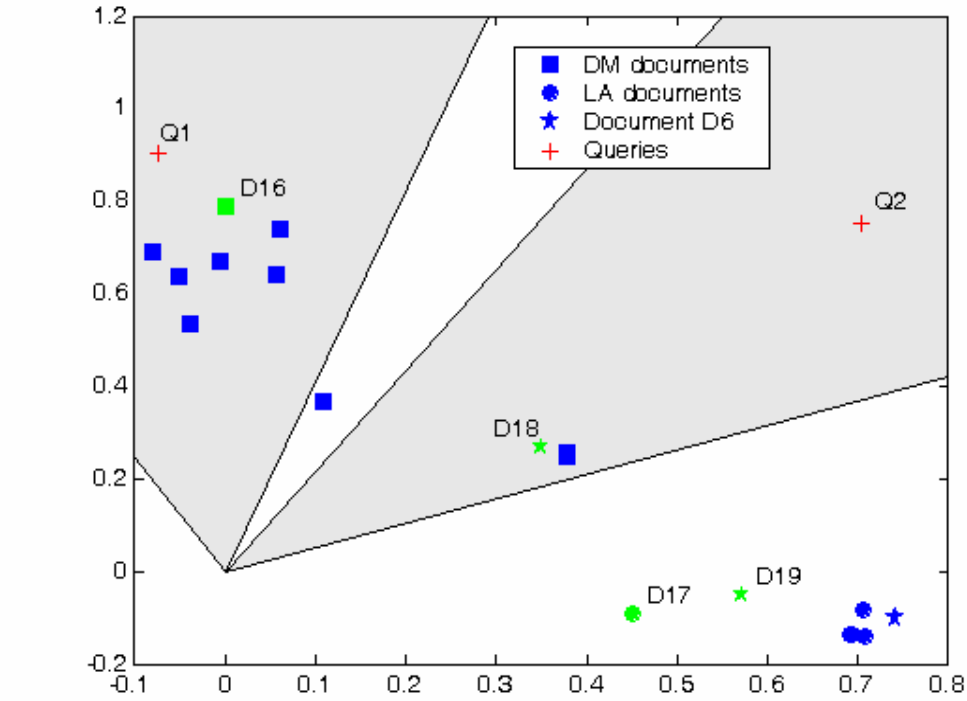


Figure 1: Representations of starting and added documents in the concept space. Representations of added document are shown in the shape that correspond to category of document, but in lighter colour then representations of starting documents. Shaded areas are areas of relevant documents for queries.

mapped in the group of DM documents and D17 document (LA document) is mapped near group of LA documents. Document D19 which combines fields of linear algebra and data mining is mapped near LA documents (because it is represented by index terms similarly as document D6) and document D18 which contains index term *data* also contained in the query Q2 is mapped in the area of relevant documents for Q2 query.

5 Experiment

Experiments are conducted on MEDLINE collection of documents. The collection contains 1033 documents (abstracts of medical scientific papers) and 35 queries. The documents of collection are split randomly into two parts: starting documents and added documents. The ratio of starting and added documents is varied: first added documents form 10% of the whole collection, then 20% of the whole collection, and so on. Starting list of index terms is formed on the basis of starting collection of documents. In the list are included all words contained in at least two documents of starting collection, which are not on the list of stop words. Further, the list of index terms is formed for the whole collection of documents in an analogous way. The obtained list of index terms for the whole collection contains 5940 index terms.

We have used measure of mean average precision (MAP) [1] for evaluation of the experimental results. Concept decomposition is conducted under starting collection of documents and added documents are represented in the concept space by using one of the described methods for approximate addition of documents. After that, an evaluation of information retrieval performance is conducted under the whole collection of documents. Dimension of the space of reduced dimension is fixed to $k=75$.

In the first row of Table 2, there is MAP of information retrieval in the case that procedure of concept decomposition is conducted under whole collection of documents (percentage of added documents is 0%). This value presents MAP in the case of recomputation of concept decomposition when new documents are added in the collection. All other values of MAP in the cases when the collection is divided into collection of starting and added documents in the different ratios, could be compared to this value. The second column of Table 2 presents number of added documents, while the third column presents number of added terms. Let us note that number of added terms grows linearly, and that the collection with only 20% of starting documents is indexed with a much smaller set of index terms then the whole collection. The fourth row presents MAP for approximate addition of documents by Method A .

Percentage of added documents	**Number of added documents**	**Number of added terms**	**MAP** Method A	**MAP** Method B α=1.0	**MAP** Method B α=1.5	**MAP** Method B α=2.0
0	0	0	54.99	54.99	54.99	54.99
10	104	456	51.98	52.20	52.33	**52.37**
20	208	753	54.96	55.10	55.09	**55.23**
30	311	1264	51.90	51.78	51.97	**52.03**
40	414	1673	50.84	50.60	51.09	**51.64**
50	517	2089	**48.64**	47.99	48.29	**48.64**
60	620	2696	44.26	44.08	45.04	**45.49**
70	723	3282	**43.59**	41.86	42.32	42.70
80	826	4024	39.87	40.56	42.56	**43.74**

Table 2: Mean average precision of information retrieval for approximate addition of new documents by Method A (without addition of new index terms) and Method B (with addition of new index terms) compared for different splits of document collection. Parameter α (used in Method B) has a role of additional stressing the importance of added terms and documents. The best results for every split of document collection are shown bolded. Generally, the best results are achieved for Method B, α=2.0, but these results are not significantly better in comparison to results obtained by Method A.

The rest columns of Table 2 present MAP of information retrieval for approximate addition of new documents by Method B for different values of parameter α.

The best results for every split of documents are show bolded. From the results we can conclude that an addition of new index terms does not improve results of MAP significantly. Namely, results obtained by Method B are better then results achieved by Method A and additional stressing of added terms and documents (for α>1) has positive effect on results. Nevertheless, results obtained by Method B, α=2.0 are not significantly better in comparison to results obtained by Method A according to pared t-test (α=0.05).

6 Conclusions and future work

Values of MAP for approximate methods are acceptable in comparison to repeated computation on concept decomposition when the number of added documents is the same or smaller than the number of starting documents. There is a drop of MAP when the number of added documents exceeds the number of starting documents. Results of MAP are not significantly improved by the methods that use extended list of index terms obtained as a result of addition of documents. It is interesting to notice that this statement is valid even in the cases when the list of index terms is significantly enlarged, which is when larger proportion of documents is added. This results show a great redundancy present in the textual documents.

In the future we plan to develop new methods of approximate addition of documents that will correct existing concept vectors by using the representations of added documents.

References

[1] R. Baeza-Yates, B.Ribeiro-Neto. *Modern Information Retrieval*, Addison-Wesley, ACM Press, New York, 1999.

[2] M. W. Berry, Z. Drmač, E. R. Jessup. Matrices, Vector Spaces, and Information Retrieval, *SIAM Review*, Vol. 41. No. 2, 1999, pp. 335-362.

[3] M. W. Berry, S. T. Dumais, G. W. O'Brien. Using linear algebra for intelligent information retrieval, *SIAM Rewiew,* Vol. 37. 1995, pp. 573-595.

[4] S. Deerwester, S. Dumas. G. Furnas. T. Landauer, R. Harsman. Indexing by latent semantic analysis, *J. American Society for Information Science,* Vol. 41. 1990, pp. 391-407.

[5] J. Dobša, B. Dalbelo-Bašić. Concept decomposition by fuzzy k-means algorithm, *Proceedings of the IEEE/WIC International Conference on Web Intelligence, WI 2003,* 2003, pp. 684-688.

[6] J. Dobša, B. Dalbelo-Bašić, Comparison of information retrieval techniques: latent semantic indexing and concept indexing, *Journal of Inf. and Organizational Sciences*, Vol.28 , No. 1-2, 2004, pp.1-17

[7] I. S. Dhillon, D. S. Modha, Concept Decomposition for Large Sparse Text Data using Clustering, *Machine Learning* , Vol. 42. No. 1, 2001, pp. 143-175.

[8] T. Kolda, D. O'Leary. A semi-discrete matrix decomposition for latent semantic indexing in information retrieval, *ACM Trans. Inform. Systems,* Vol. 16, 1998, pp. 322-346.

[9] G.W. O'Brien. *In formation Management Tools for Updating an SVD-Encoded Indexing Scheme,* Master s thesis, The University of Knoxville, Tennessee, 1994.

[10] J. Yen. R. Langari. *Fuzzy Logic: Intelligence, Control and Information,* Prantice Hall, New Jersey, 1999.

A General Brokering Architecture Layer and its Application to Video on-Demand over the Internet

Franco Cicirelli and Libero Nigro
Laboratorio di Ingegneria del Software
Dipartimento di Elettronica Informatica e Sistemistica
Universitá della Calabria, I-87036 Rende (CS) - Italy
E-mail: f.cicirelli@deis.unical.it, l.nigro@unical.it

Keywords: service oriented computing, application framework, middleware, peer-to-peer, Internet, video on-demand, Java, Jini, Java Media Framework, RTP/RTCP protocols

Received: February 7, 2006

GOAL -General brOkering Architecture Layer- is a service architecture which allows the development and the management of highly flexible, scalable and self-configurable distributed and service-oriented applications over the Internet. GOAL centres on a design pattern which decouples the design of service functionalities from the distribution concerns. A service wrapper is specifically responsible of the distribution aspects. The wrapper is weaved at runtime to its corresponding service by a dynamic proxy object. GOAL makes it possible to augment, in a transparent way, the behaviour of a software object in order to permit it to be remotely accessible. Current implementation of GOAL depends on Sun Microsystems' Jini as the brokering/middleware layer. The paper describes GOAL and demonstrates its practical use through the design and implementation of a Video on-Demand system. Java Media Framework is used for pumping multimedia data at the transmitter side and for rendering purposes at the receiver side, RTP/RTCP protocols are used for multimedia streaming.

Povzetek: Predstavljen je GOAL – arhitektura za napredne spletne aplikacije, npr. video na zahtevo.

1 Introduction

Service Oriented Computing [1] emerged in the last decade as a computing paradigm centred on the concept of *service* [2, 3] as basic building block. Services are suitable for developing and organising applications for large-scale open-environments. They are effective in improving software productivity and quality, as well as fostering system evolution and maintenance. A service is a coarse-grained software component virtualizing a hardware or software resource which is made exploitable for on-demand use. Binding to the service/resource typically occurs just at the time the component is needed. After usage the binding may be discarded. Applications, tailored according to user requirements, are constructed through a combination and composition [4] of independent, outsourced service components exposing a well-defined interface. Main features of the service paradigm include dynamism and transparency. The former refers to services which can appear or disappear in a community without centralized control, possibly notifying about their presence/absence. This behaviour depends on the use of the so called discovery protocols [5, 6, 7]. The latter feature means that services can be used without knowledge about service provider platforms and service provider locations. Service dynamism and interaction model strongly relate service architectures to peer-to-peer architectures [8]. Service computing relies on the high-level abstraction entities defined by Service Oriented Architecture (SOA) [9, 10] in order to (i) characterize and organize service-based applications and (ii) capture the relationships existing among these entities. Basic entities in a SOA are the *service provider*, the *service client*, and the *service registry* which acts as a broker among clients and providers. Each service, offered by a provider, preliminarily requires to be advertised in a registry in order for it to become subsequently discoverable and utilizable by a client (e.g. a human user or another service).

GOAL, the General brOkering Architecture Layer proposed in this paper, is a novel service architecture allowing the development of highly flexible, dynamic and self-configurable service-based applications. GOAL aims at simplifying the management of service lifecycle by reducing the burden of designing, developing and deploying software objects suitable to work in a distributed context. Different distribution aspects like data consistency, fault tolerance, security and remote communications are treated as cross-cutting aspects. In particular, the latter two concerns are directly addressed by the system and are the responsibility of *proxy* objects. GOAL offers a minimal *framework* [11], easy to understand and use, and a few *meta-services*. A *service design pattern*, enforcing common guidelines for service development, is provided. Designing a new service does not introduce dependencies from a particular API or system components. Weaving customer-

objects and system-objects occurs only during service operation. Meta-services are system entities which allow one to publish, search and use customized services. A service, once advertised, becomes available within a GOAL community. Matching criteria can be specified during a searching phase. In addition, when a new service appears/leaves the community, interested clients can be automatically notified. The notification mechanism ensures system flexibility and scalability and permits the development of highly dynamic and self-adapting software. A service can leave the community due to an explicit removing operation or due to a crash. In the latter case, self-healing properties are carried out using a fail silent model [12] based on a leasing mechanism.

Current implementation of GOAL depends on Jini [7, 13, 14] as the underlying service infrastructure, borrowing advantages of dynamic registration, service lookup, notification of remote events, distributed object access and platform-independence enabled by Java. However, the brokering layer, i.e. Jini, is fully abstracted and can possibly be replaced. Communication among services is based on the exchange of Java objects and fully exploits benefits of *runtime code mobility* [15].

GOAL can be used as the starting point for building further abstraction layers targeted to specific application domains. As a significant example, a Management Architecture for Distributed meAsureMent Services -MADAMS- [16] was developed directly on top of GOAL mechanisms. MADAMS is tuned to the requirements of distributed measurement systems [17, 18, 19]. MADAMS rests on the concept of *measurement service* as the basic abstraction entity modelling a (physical or virtual) measurement instrument, and the concept of *connector* which provides inter-instrument communications. MADAMS also supports recursive service composition. MADAMS was successfully employed for demand monitoring and control [16] and for remote calibration of distributed sensors [20].

This paper describes GOAL features and the general design guidelines for achieving distributed services. As an application, GOAL capabilities are demonstrated through the development of a distributed Video on-Demand (VoD) system [21, 22, 23]. The VoD system depends on Java Media Framework (JMF) [24] which is responsible both for pumping multimedia data into a network connection at a sender side and for presenting multimedia data at a receiver side. Data streaming depends on RTP/RTCP protocols [25].

The structure of the paper is the following. Section 2 summarizes some related work. Section 3 presents the proposed service architecture along with its programming model. In particular, a description about the service design pattern, system security and the catalogue of meta-services is provided. Section 4 illustrates design and implementation and service catalogue concerning the prototyped VoD system. Finally, an indication of directions which deserve further work is furnished in the conclusions.

2 Related Work

As with other technologies suited to the development of distributed systems, ensuring transparency and hiding management of distribution concerns allow developers to focus only on domain-specific problems. For these purposes, different infrastructures and middleware layers have been proposed centred on the service metaphor. Sun Microsystems' Jini [13, 7, 26] permits the construction of service-based applications in terms of fundamental mechanisms of service publication/discovery, leasing management, remote event notification and transaction support.

In [27] an approach based on *tuple-space* [28] for building service frameworks is proposed. Concepts like *actor*, which execute client requests, *virtual resource* and *virtual service* are introduced. Virtual entities enable abstraction layers to be achieved on top of either physical resources or services thus ensuring a common and uniform way for accessing them. Spaces are used to manage (e.g. create, destroy, search) agents, services and resources.

Other solutions are targeted to abstracting and hiding details of the adopted middleware/brokering layer in order to favour its interchangeability. By providing a well-defined set of components (i.e. interfaces and objects) and through code generation mechanisms, Iceni [29, 30] allows an automatic service management in different computing contexts such as Open Grid Service Infrastructure or Jini service community. In [31] a framework is proposed which hides behaviour of underlying transport layer and separates *coordination patterns*, i.e. request/response interactions, from *computational logic*, i.e. service functionalities.

Colombo platform [32] introduces the concept of *servicelet* as the unit of development and deployment. A servicelet is a stateless object corresponding to a single service or to a collection of them. Context information are managed by specific *Servicelet Context* entities which are handled by the runtime system. Management of explicit metadata in the form of machine-readable service descriptions, including functional and non-functional QoS characteristics, is an important characteristic of Colombo. The goal is to avoid generating a gap between the internal representation of service capabilities and the external, interoperable service view which is defined by the service contract.

Sirena framework [33] defines an architecture to seamlessly connect heterogeneous (resource constrained) devices and services furnished by such devices. Sirena comprises an incoherent set of tools having the responsibility of generating service stubs and skeletons, managing service lifecycle, supporting visual composition for service orchestration and so forth.

A different goal is pursued in Arcademis [34] which is a Java-based framework enabling the implementation of modular and highly customizable middleware architectures for specific application domains. A distributed system built on top of Arcademis is structured according to *three abstraction levels*. The first level is constituted by basic components like invokers, which are responsible for emitting

remote calls, or schedulers which are used to possibly order remote calls. Although these are abstract classes and interfaces, Arcademis also provides concrete components that can be used without further extensions. The second level is represented by the concrete middleware platform obtained from Arcademis basic components. The framework defers to this level decisions about serialization strategy, communication and lookup protocols that will be adopted. Finally, the third abstraction level is made up by components which make services available to end users.

In the context of the above mentioned proposals, the original contribution of GOAL is twofold: (i) to allow development of new services without introducing, at design time, bindings to specific framework components (e.g. abstract classes or interfaces), (ii) to transparently handle distribution concerns as cross-cutting aspects. All of this fosters low coupling among entities, system evolution and maintenance in a natural way.

3 GOAL Service Architecture

GOAL addresses all the activities involved in the lifecycle of services by exploiting a specific service design pattern and by using a set of well-defined system components and interfaces having specific roles. A main concern rests in encapsulating and hiding implementation details of core components by using stable interfaces so that if changes occur, e.g. in the middleware layer which is replaced or in the communication protocol stack, no consequence is induced in the implemented and working applications. GOAL components and features are discussed in the following.

3.1 Service Design Pattern

The development of a generic service follows the service design pattern depicted in Fig. 1. Each remote service, i.e.

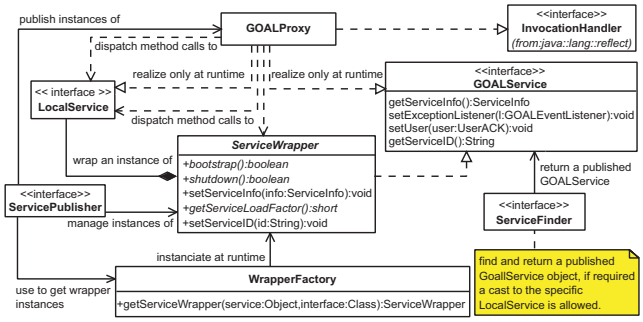


Figure 1: Components of service design pattern.

a new software object made available within a GOAL community, is first developed as a local object. This permits design efforts to concentrate only on the effective service functionalities. In this design phase, the only constraint to fulfil is in defining functional aspects of the new service by means of an interface. One such interface is shown in Fig. 1 as the *LocalService* interface. Interfaces allow a what-how decomposition [35] which ensure service client code immutability with respect to service implementation changes. Any object can be a candidate for a remote service because no restrictions are introduced in the definition of its functional behaviour except for the serializability of the parameters appearing in method signatures. Remote concerns are managed by means of a service wrapper. This kind of object extends local service behaviour with distribution aspects like transaction support and so forth. As a common guide line, the wrapper may enfold the local service and execute distributed tasks by interleaving them with method calls on the wrapped service. A service wrapper can require to be bootstrapped, for instance by initializing its internal state with information retrieved by contacting other services. A shutdown operation allows a wrapper to tear down, i.e. becoming out of work or unpublished. All of this is reflected in Fig. 1 by the *ServiceWrapper* abstract class. Other common functionalities allow: (a) setting the service identifier, (b) setting service info (e.g. a description of service behaviour and functionalities) and (c) managing an estimated load factor value of the service provider. By default, the above concerns are system addressed. When no special requirements have to be met, a *DefaultWrapper* can be transparently used. Would new functionalities be added to the local service, e.g. in order to cope with data consistency and integrity among multiple system nodes, an extension of the default wrapper may be supplied. At compile time, the local service interface and the relevant wrapper may be completely unrelated. A wrapper has to override only the local service operations whose capabilities require to be extended. Only at runtime, the wrapper and the local service behaviour will be weaved according to an aspect-oriented programming style [36]. Two problems arise when making a local service remotely accessible: (i) the service has to be advertised into a community, (ii) a representative object for the service, i.e. a *GOAL proxy*, is required to be downloaded on service client in order to support remote communications. Service advertisement is the responsibility of the *ServicePublisher* meta-service (see Fig. 1). Service finding and proxy download activities are in charge of the *ServiceFinder* meta-service (see Fig. 1). The proposed publisher/finder mechanisms help in hiding details about the actual brokering layer. Would the brokering layer be replaced, e.g. CORBA used instead of Jini, only the publisher/finder objects have to be correspondingly modified. While publishing a local service, behind the scene the service publisher (i) asks to a *WrapperFactory* for a wrapper instance, (ii) correlates service and wrapper with the proxy and (iii) makes the latter one object available to a GOAL community using functionalities of the actual brokering layer. The *GOALProxy* (see Fig. 1) is a remotely accessible object which, moving to a service client host, transparently acts as a dispatcher of request/response messages between remote user and local service provider. In the case the communication protocol changes, e.g. XMLRPC is preferred to Java RMI, only

the proxy requires to be changed. By preferring the execution of overridden methods, the proxy realizes the interweaving among local service and the corresponding wrapper. At runtime, by exploiting Java dynamic proxy mechanisms [37], a GOALProxy is able to implement a list of specified interfaces without requiring code generation. Implementing a specific interface of a local service ensures that a generic client would not be able to perceive any difference between direct service usage with respect to proxy mediate usage. The sequence diagram in Fig. 2 summarizes the effects of the service design pattern on service publication and utilization. Figure 3, instead, depicts communication details characterizing interactions among service client and the associated service provider. A GOAL-

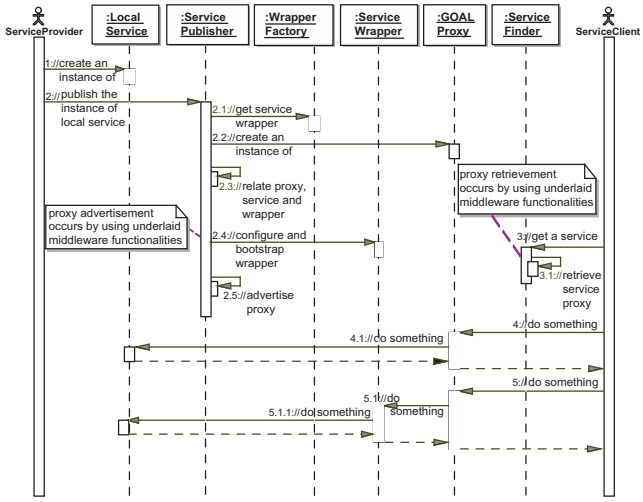


Figure 2: Sequence diagram capturing service utilization.

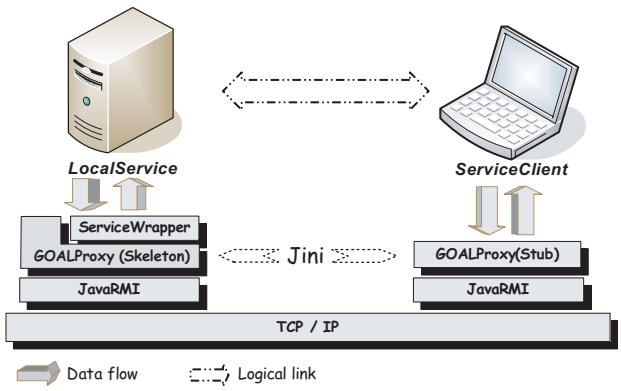


Figure 3: GOAL service usage scenario: communication details.

Proxy may enforce the download of the entire service code on a user node. This is useful when the service must be executed on the client side, e.g. for accessing to hardware or software resources hosted on a particular computing node [20]. Proxy behaviour is defined during publication simply by setting some of the so called *GOALServiceProperty(s)*. If no constraints appear in the object serializability or persistence, a service may be used according to *remote* or *downloadable* mode. A GOALProxy may be specialized in order to augment the capabilities of the overall system. For instance, to deal with fault-tolerance concerns, a proxy can be designed to abstract communications between a single client and a group of equivalent service providers, so as if one of the provider becomes unavailable or crashes, the client remains able, in a transparent way, to continue its work [38]. The class diagram of the service design pattern (see Fig. 1) makes also clear that, once published, a local service becomes a *GOALService*. GOALService interface defines a set of functionalities which are common to all services in a GOAL system. In particular, the setExceptionListener method is used to set a listener whose aim is to handle exceptions not thrown by local service methods but raised during remote operation. The setUser method is used to set a *UserACK* object especially devoted to cope with security concerns (see section 3.2). Remaining operations should be self explanatory. Other common issues of the service design pattern are user *friendliness* and *load balancing* support. Each service may possess a graphical user interface (GUI) obtained through a *GUIfactory* object previously published by using the service publisher (see also Fig. 6). Service finder is then used by clients in order to search and retrieve the factory. Advantages of this approach are: (i) a service is developed independently from its graphical interface, (ii) the GUI is instantiated only on the client side thus avoiding serialization of graphical objects, (iii) the GUI allows use of the service without any previous knowledge about it, (iv) multiple graphical user interfaces, possibly tied to different user node capabilities, can be supported. Load balancing is carried out by using the so-called *remote service attributes*. Every service has one of such an attribute that expresses its load factor, i.e. NORMAL for a low or normal load factor, and WARNING or BUSY for a high/very high load factor. Although services are usually supposed to remain context-free, remote attributes can provide a kind of context information [39, 40] exploitable during the finding phase. For instance, the service finder always tries to return services in the NORMAL state, if there are any, otherwise the first one matching searching criteria is returned. The service publisher keeps up to date remote attributes by periodically querying service wrappers state or (possibly) by monitoring the CPU usage on provider nodes.

3.2 Security Concerns

Handling security is an essential issue in a distributed and multi-user scenario. The service code downloaded from a remote site requires to be trusted along with the remote site itself. User credentials must be verified, usage of system resources granted and resource accesses controlled. Authentication and authorization mechanisms of GOAL are im-

plemented through the UserACK object (see Fig. 1) which permits user identification and acknowledgment of its roles (e.g. administrator or normal user), privileges and grants. The concept of *user groups* is introduced and users may become members of one or multiple groups. Each group, e.g. admin group, owns some *GOAL permissions* and the UserACK holds the union of all permissions relevant to the user joined groups. Information stored in a permission follow the same hierarchical schema adopted in Java package specifications. Creating a permission with a service package info and service name enables access to the corresponding service. By providing only package information, a grant is given to all services belonging to the package. A finer authorization control is achieved by specifying service method/function name(s) in the permission. The use of UserACK makes it possible, in a decentralized context, to accept or discard a user request. User grants are checked directly by GOAL proxies. Therefore, authentication and authorization concerns are transparently managed with respect to service functionalities and service implementation. During publication, a specific GOALServiceProperty can be used to state if the proxy has to enable or disable the management of security concerns, i.e. to state if the service has to be considered *secure* or *public*. In the case security aspects are to be explicitly managed by a service, the UserACK object must be transmitted as a parameter when invoking its methods.

Users can freely propose new groups and create their own UserACK. However, only *signed* UserACKs and *accepted* groups can be effectively used. A system administrator signs a new UserACK and establishes its expiration time. Signed UserACKs cannot be modified by users: the system is able to recognize when a UserACK is corrupted, modified or just invalid (e.g. it expired). UserACK and group management is responsibility of the core downloadable *Grant Management service* whose GUI is shown in Fig. 4. A UserACK submission form is offered and group membership is achieved by choosing items from a list of already accepted groups. Inspection of group properties is allowed. Likewise to UserACK, a group submission form is also available. Submitting a new group requires the group name and the list of group permissions to be provided. A reserved area, offering an overall vision of existing UserACKs and groups, is under the control of the system administrators (see Fig. 5). New UserACKs/groups can be signed/accepted and single permissions can be added/removed in a group as well as in a submitted UserACK.

The accesses to a service can be also allowed or denied depending on other criteria. Load balancing aspects, service availability or service exclusive use may often be considered during the UserACK acquisition phase. Confidentiality and privacy can be ensured by using the Secure Socket Layer for service communications, whereas trustness can be achieved by exploiting Java standard security mechanisms relevant to remote code management [41].

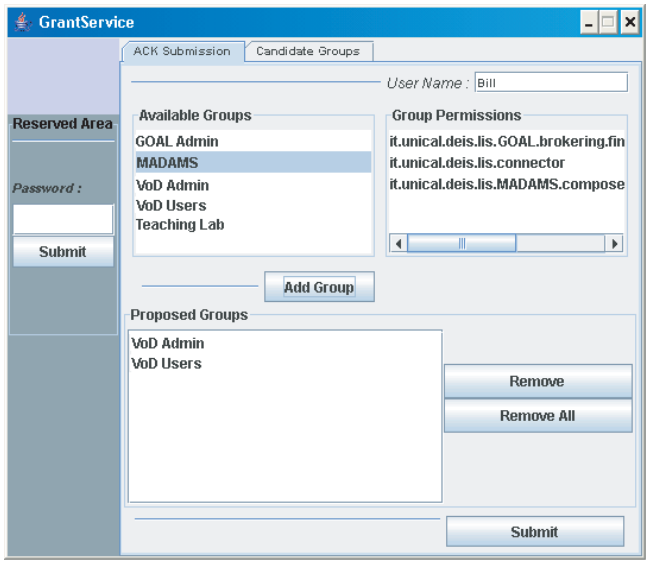


Figure 4: Grant Management service GUI.

3.3 Meta-Service Catalogue

GOAL meta-services are responsible for publishing, searching, retrieving or removing services from a community. Figure 6 portrays the UML class diagram of the publisher/finder services which depend only on interfaces. Actual objects are created by a singleton factory which ensures a coherent delivering according to the underlying middleware layer. To cope with security concerns, a valid UserACK is required when using meta-services. Only the method find(String):GOALService can be used without a UserACK. This provides a bootstrap mechanism exploitable by new users for contacting the service devoted to grant management in order to obtain the personal UserACK. The advertisement process is carried out by requiring the service to publish and a list of GOALServiceProperty. These properties allow to specify descriptive and behavioural attributes. Descriptive attributes may be used, for instance, to provide service description. Behavioural attributes must be used to specify the name of the interface through which the service will be retrieved by clients, the wrapper to use and so forth. Following a successful invocation, the publish method returns the unique service identifier. A service may be published using different interfaces thus allowing multiple views of the same local object. Among service properties it is also possible to specify a service working directory which will be used, by the system, for storing persistent information like the service identifier. Service properties may also be labelled as searchable. In this case, properties may be specified as matching attributes during the finding phase. The publishServiceUIFactory method is used to publish the UIFactory of a specified service, unpublish is used instead to remove a service from the community. Finding a service requires the service name, i.e. its interface name, and (possibly)

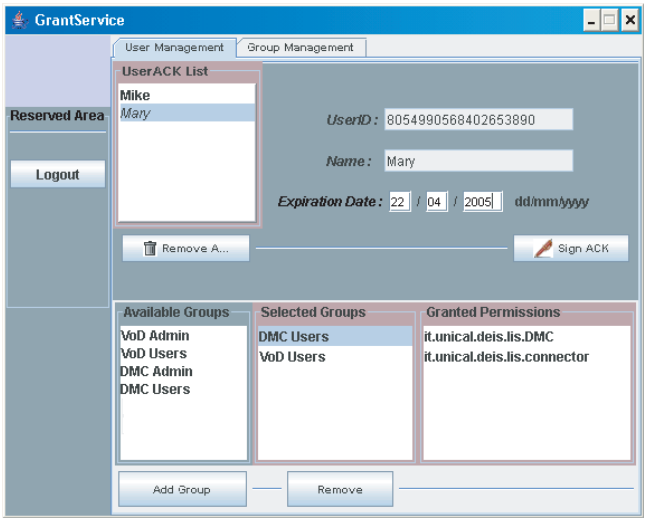


Figure 5: Administration panel of the Grant Management service.

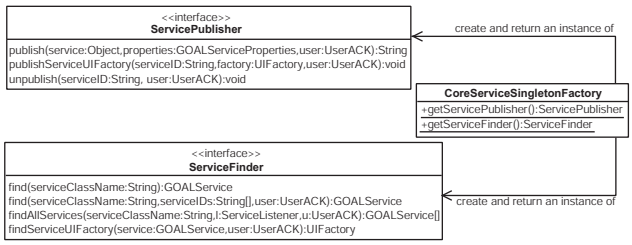


Figure 6: Publisher/finder service design.

service identifier information. Although the finding process is based on naming criteria, the matching can also occur when a published service implements any interface in a hierarchy. As a side-benefit, the use of textual name and the availability of service GUI enable usage of any published service without requiring specific Java code to be installed on the client node. The findAllService method allows service retrieval by bypassing the load balancing policy. If specified, a ServiceListener (see Fig. 6) notifies when a new searched service joins or leaves the community. Meta-services require their code to be pre installed on every GOAL node.

4 A GOAL-based VoD System

The following describes a VoD system developed on top of GOAL. The VoD system consists of a service federation which permits publishing, searching and retrieving as well as streaming and rendering of multimedia contents. Java Media Framework [24] is used for pumping (at provider side) and rendering (at client site) multimedia data. Streaming of multimedia contents relies on the RTP/RTCP protocols [42]. First the service architecture is described, then the list of developed services for the VoD system is provided.

4.1 System Architecture

The architecture of the achieved VoD is depicted in Fig. 7. It consists of five types of computing nodes having different roles in supporting VoD services. Nodes, and relevant services, can dynamically join or leave the system and when this occurs the other nodes are notified. Some

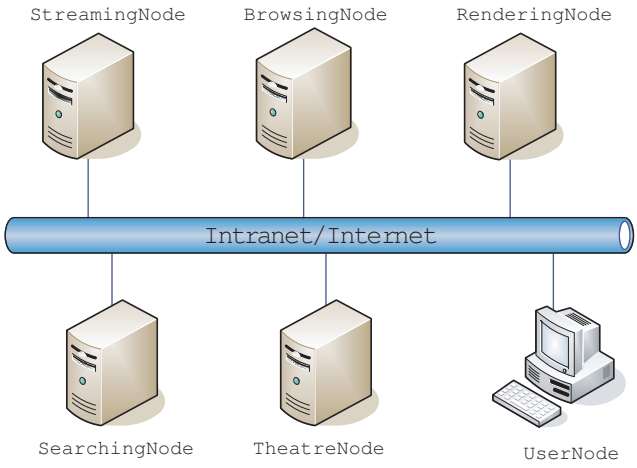


Figure 7: Architecture of the GOAL-based VoD system.

nodes may be duplicated: multimedia files and related descriptions are normally distributed across multiple *streaming nodes*. Other kind of nodes, instead, may be duplicated for fault-tolerance and load balancing issues. VoD services are ultimately requested and made it available to final users through *user nodes* (see Fig. 7). The architecture was designed so as to minimize code requirements on the user nodes. Here only some standard code like JMF and obviously GOAL meta-services code, is supposed to be statically available. All of this contributes to system evolution because, by exploiting the download of the service code, a user, on-demand, will always use the latest version of the various software components. A description of each node is provided in the following.

Streaming nodes are media servers. They contain multimedia data files and associated descriptions (e.g. title, authors etc.). Streaming nodes enable to: (i) access the descriptions of media data; (ii) add/remove multimedia files; (iii) create and manage multimedia sessions (for data streaming and control) on behalf of end users.

Browsing nodes respond to management functionalities. Relevant services offer a unified list of the multimedia content available on the various streaming nodes and allow users to organize multimedia data across them. The organization consists in adding/removing/modifying multimedia contents on different streaming nodes.

Searching nodes portray a whole vision of all the existing multimedia contents by providing: (i) the unified list of available media files distributed across streaming nodes; (ii) searching facilities, e.g. for selecting specific movies; (iii) user profiles in order to tailor media information on a per user basis or to send notifications when relevant new media data come into existence; (iv) trace facilities about media utilizations like reviews, user preferences and so forth.

Rendering nodes act as remote libraries from which user nodes can dynamically download the code required for rendering a video content, possibly by ensuring also receiver based QoS control, e.g. lip-sync [43].

Theatre nodes provide a service which is used as entry point for user interactions. In order to view a movie a user has to (i) searching it by using searching node functionalities, (ii) starting and managing multimedia sessions by using streaming node services, (iii) managing the rendering process on the user node by retrieving and using rendering libraries downloaded from a rendering node. All of this requires the utilization and the coordination of multiple VoD services which in turn are provided by different computing nodes. By using the service exported by a theatre node, a user obtains an holistic vision of the entire VoD system. In this way, issues concerning single service invocation and coordination are fully abstracted.

4.2 Service Catalogue

SessionController and VideoFileManager

Are specific of streaming nodes. SessionController negotiates and creates a *multimedia session* between a client node and a streaming server node, with distinct *control* and *streaming* bindings. The streaming binding is used for media data streaming, e.g. unicast, and relies on the RTP/RTCP protocols [25]. A negotiation phase is required for establishing port identifiers at both receiver and transmitter side. The control binding is TCP-based and is used for exchanging session control commands (e.g. play, rewind, pause and stop). SessionController does not require its functionalities to be extended for remote access. Therefore, the DefaultWrapper can be transparently used during the publication phase. SessionController service has a VCR-like GUI which is automatically made available at the end of the negotiation phase.

The VideoFileManager service is mainly devoted to adding/removing media files to/from a specific streaming node and managing media file information, e.g. title, director and language. Information about duration, file encoding and so forth are automatically detected and made available by the service. Media information are stored in XML format. VideoFileManager also notifies a set of listeners when, for instances, a new movie is added. A complete list of available movies is also provided. In order to enforce data consistency, listeners require to be notified under transactional support. Transaction management is responsibility of the VideoFileManagerWrapper (see Fig. 8) and relies on the Jini transaction mechanism. The class

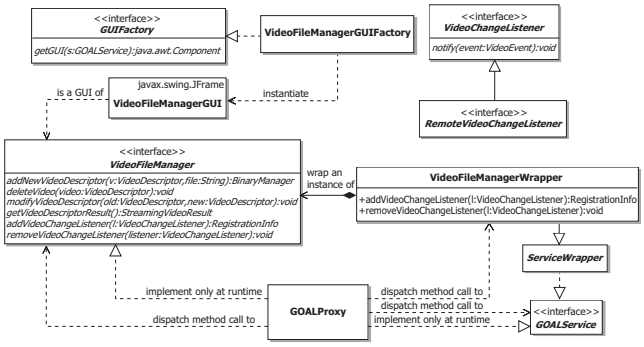


Figure 8: Class diagram of VideoFileManager and related entities.

diagram in Fig. 8 makes clear that the wrapper and the corresponding service are unrelated at compile time, i.e. they do not implement or extend any common entity. As discussed in section 3.1, the weaving between the two objects is only established at runtime by means of a GOALProxy. During the bootstrap phase (see section 3.1), the wrapper registers itself as a listener of the VideoFileManager. Subsequently, it will act as a dispatcher of notifications coming from the wrapped service and going toward remote listeners. By overriding the methods addVideoChangeListener and removeVideoChangeListener, the wrapper obtains to be the exclusive manager of the RemoteVideoChangeListener(s) which are handled under transaction. A *RemoteVideoChangeListener* is a listener whose functionalities are extended to support notification and management of remote events. In addition, the remote listener behaves as a transaction participant when notified by a transaction client [26], i.e. a VideoFileManagerWrapper. To enforce self-healing properties, the registration of a remote listener is regulated by a lease. As one can see, all the methods reported in Fig. 8 makes no use of UserACK objects, this is because security concerns are transparently handled by the GOALProxy. Figure 8 also shows the relationship existing between VideoFileManager and its GUI. Figure 9 depicts the GUI of a VideoFileManager service while an upload of a new movie occurs.

StreamSearcher

It is specific of searching nodes and provides a searching facility allowing a uniform access to all the media data available on existing streaming nodes. A StreamSearcher enriches media data with information about user activities and collects user reviews and profiles. It acts as a listener of events coming from VideoFileManager, e.g. informing that a new movie has been added to the video library, or coming from other StreamSearcher, e.g. informing that another review was added. This is reflected in the class diagram reported in Fig. 10 where a StreamSearcher interface extends the VideoChangeListener interface. As one can see,

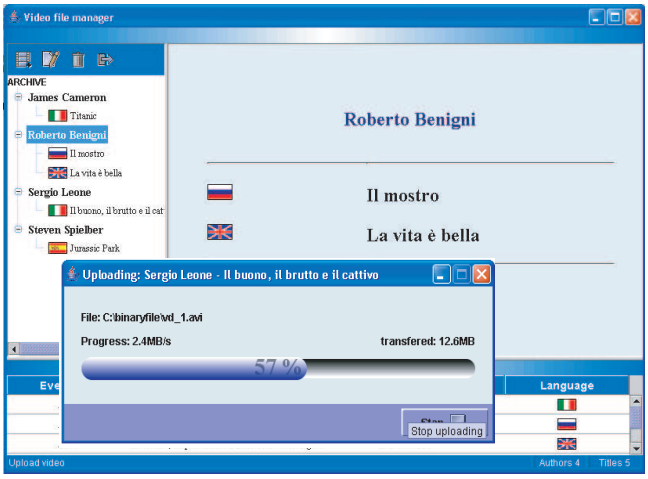


Figure 9: VideoFileManager GUI.

some methods of StreamSearcher require a UserACK object as parameter. Although security concerns are always managed by the proxy, one of such an object is required for tracing user's activities. Likewise to the VideoFileManager

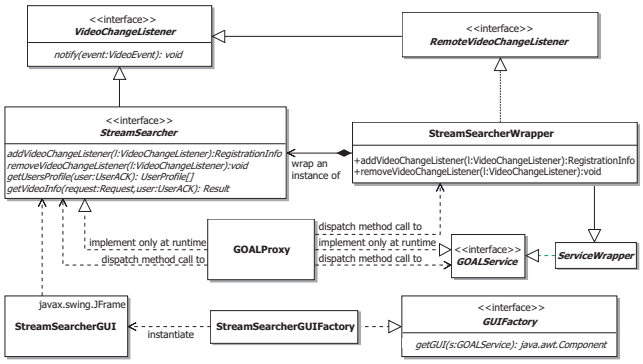


Figure 10: Class diagram of StreamSearcher and related entities.

service, a StreamSearcherWrapper (see Fig. 10) was introduced for guaranteeing consistency and integrity of the data exchanged with other services, i.e. VideoFileManager(s) and StreamSearcher(s). In this case, the wrapper extends the RemoteVideoChangeListener interface. In particular the wrapper registers itself as a listener of the enfolded service and as listener of the VideoFileManager and StreamSearcher working in the service community. At the bootstrap phase (see Figg. 1 and 2) the wrapper is in charge of initializing its own media data repository. If other searching nodes are available, a data mirroring is performed, otherwise it has to contact all the VideoFileManager(s) in order to retrieve info about movies. Media data and the so called enriched media data (i.e. title, authors, user reviews, and so forth) are represented in XML and stored in an XML DBMS such as eXist [44]. Figure 11 shows an interaction with the StreamSearcher service with a specification of searching criteria for finding a movie. The wrapper acts either as a transaction participant or a transaction client. Only at transaction commit, data received from other nodes are transmitted to the enfolded service.

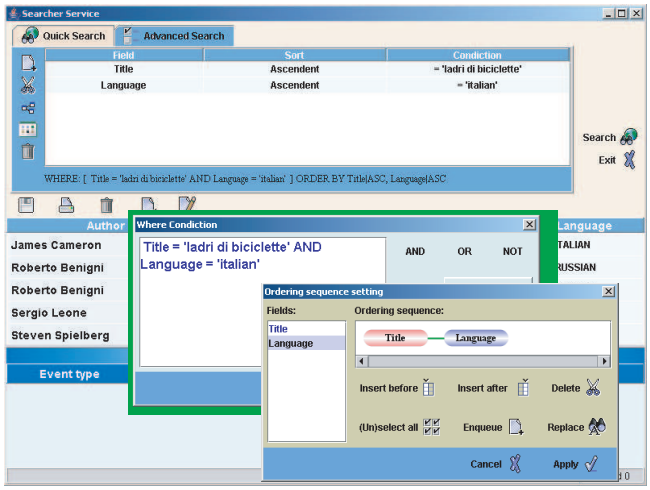


Figure 11: StreamSearcher GUI.

Renderer

It is specific of rendering nodes and, in the context of a multimedia session, it assists the audio/video rendering process on a client node. The rendering process can possibly be accompanied by a receiver-based QoS control filter [45]. Within an allowed end-to-end delay, one of such a filter separately buffers incoming audio/video packets, assembles media frames and synchronizes media frame presentation on the basis of their presentation time (this is achieved by elaborating either timestamps of RTP packets and report packets of the RTCP sender in order to periodically adjust the real-time clock of the receiver subsystem to the real-time clock of the sender). Too late arriving or corrupted packets are discarded. The filter is capable of controlling intra-medium jitter and inter-media skew directly affecting the lip-synch problem [43]. Rendering service allows management of volume and video zoom factor as well as shows reproduction time of the rendering movie. This service requires to be fully downloaded on client node and no remote functionalities are added.

Browser

Specific of Browsing nodes, this service allows the listing of the VideoFileManager available into the VoD system. This service is only for management purposes i.e. selecting a streaming node to administrate. Browser service requires to be fully downloaded on a client node and no remote functionality is added.

Theatre

Specific of Theatre nodes, this composed service provides added value to final user activities. Behind the scene a theatre asks for a searching service and for a rendering service as well as, once a movie is chosen, for the right session controller service in order to start and manage the incoming multimedia session. A negotiation phase is required between the SessionController and the Renderer for according IP addresses and port numbers. Theatre is a downloadable service which does not require remote functionalities to be added and the DefaultWrapper is used during its advertisement. The theatre service does not have an own graphical interface: it supports user interaction through the GUI(s) of the component services (see Fig. 12).

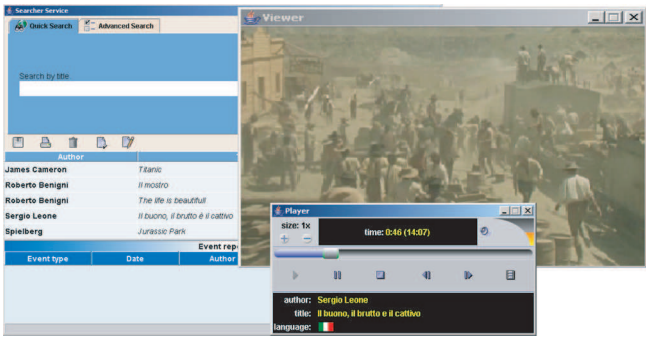


Figure 12: Theatre service vision.

5 Conclusions

The General brOkering Architecture Layer facilitates the development of general-purpose distributed service-based applications. GOAL is based on a service design pattern and makes an application development independent with respect to a specific middleware technology. By keeping a clear separation between local vs. remote concerns and by exploiting the service metaphor GOAL fosters software evolution and maintenance. Development time and design efforts as well as the initial background required for making an application remotely usable are very small. Management, though, of more complex distribution concerns like transaction support, requires a deeper knowledge about GOAL components and the underlying middleware layer. GOAL mechanisms have been successfully experimented in the realization of significant applications like distributed measurement systems [16, 20]. This paper reports about the achievement of a Video on-Demand system over the Internet. Current implementation of GOAL depends on Java/Jini technology. Directions of further work include the following:

- specializing service proxies with the purpose of allowing interoperability between GOAL services and Web Services [46]
- introducing design by contract [47] by foreseeing pre-conditions and post-condition to be transparently managed via service proxy when calling a service method
- making it available functionalities for supporting long term transaction [48] by offering a coordinator core-service accepting a list of methods to be managed under transaction
- adding management of non functional aspects [49], such as service availability, service response time and throughput, either in the service advertisement or within the finding process
- extending the VoD system in order to support multicast and cooperative multimedia sessions [23].

References

[1] M.P. Papazoglou and D. Georgakopoulos. Service oriented computing. *Communications of the ACM*, 46(10):24–28, 2003.

[2] K. Bennett, P. Layzell, D. Budgen, P. Brereton, L. Macaulay, and M. Munro. Service-based software: the future for flexible software. In *Proceedings of the Seventh Asia-Pacific Software Engineering Conference (APSEC'00)*, pages 214–221, Washington, DC, USA, 2000. IEEE Computer Society.

[3] R. Perrey and M. Lycett. Service-oriented architecture. In *Proceedings of the Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, pages 116–119. IEEE Computer Society, 2003.

[4] J. Yang and M. P. Papazoglou. Service components for managing the life-cycle of service compositions. *Information Systems*, 29(2):97–125, 2004.

[5] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service location protocol, version 2, rfc 2608. `http://www.ietf.org/rfc/rfc2608.txt`. Accessed on October 2005.

[6] UPnP. Universal plug and play device architecture. `http://www.upnp.org/download/UPnPDA10\_20000613.htm`. Accessed on October 2005.

[7] W.K. Edwards and W. Edwards. *Core Jini*. NJ: Prentice Hall, second edition, 2001.

[8] R. Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings of the First International Conference on Peer-to-Peer Computing (P2P '01)*, pages 101–102, Lingkoping, Sweden, August 2001. IEEE.

[9] M.P. Papazoglou. Service-oriented computing: Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering*, pages 3–12. IEEE Computer Society, December 2003.

[10] M. Shaw and D. Garlan. *Software architecture: perspective on an emerging discipline*. Prentice-Hall, 1996.

[11] M.E. Fayad and D.C. Schmidt. Object-oriented application framework. *Communications of the ACM*, 40(10):32–38, 1997.

[12] D. Cotroneo, C. Di Flora, and S. Russo. Improving dependability of service oriented architectures for pervasive computing. In *Proceeding of the Eighth International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS'03)*, pages 74–81. IEEE Computer Society, 2003.

[13] Sun Microsystems. Jini network technology - specifications (v2.1). `http://www.sun.com/software/jini/specs/index.xml`. Accessed on May 2006.

[14] Jini network technology. `http://www.sun.com/software/jini/`. Accessed on October 2005.

[15] A. Carzaniga, G.P. Picco, and G. Vigna. Designing distributed applications with mobile code paradigms. In *Proceedings of the 19th International Conference on Software Engineering*, pages 22–32. ACM Press, 1997.

[16] F. Cicirelli, D. Grimaldi, A. Furfaro, L. Nigro, and F. Pupo. MADAMS: a software architecture for the management of networked measurement services. *Computer Standards & Interfaces*, 28(4):396–411, 2006.

[17] D. Grimaldi, L. Nigro, and F. Pupo. Java based distributed measurement systems. *IEEE Transactions on Instrumentation and Measurement*, 47(1):100–103, 1998.

[18] A. Furfaro, D. Grimaldi, L. Nigro, and F. Pupo. A measurement laboratory over the internet based on Jini. In *Proceedings of the Twelfth IMEKO TC4*, pages 479–501, 2002.

[19] W. Winiecki and M. Karkowski. A new Java-based software environment for distributed measuring systems design. *IEEE Transactions on Instrumentation and Measurement*, 51(6):1340–1346, 2002.

[20] F. Cicirelli, A. Furfaro, D. Grimaldi, and L. Nigro. Remote sensor calibration through MADAMS services. In *Proceedings of the IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'05)*, Sofia, Bulgaria, 2005.

[21] L.A. Rowe, D.A. Berger, and J.E. Baldeschwieler. The Berkeley Distributed Video on-Demand System. In T. Ishiguro, editor, *Proceedings of the Sixth NEC Research Symposium*, pages 55–74. SIAM, 1995.

[22] K.C. Almeroth and M.H. Ammar. The Interactive Multimedia Jukebox (IMJ): a new paradigm for the on-demand delivery of audio/video. In *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, pages 431–441, 1998.

[23] G. Fortino and L. Nigro. ViCRO: an interactive and cooperative videorecording on-demand system over Internet MBone. *Informatica*, 24(1):97–105, 2000.

[24] Java Media Framework. `http://java.sun.com/products/java-media/jmf/index.jsp`. Accessed on November 2005.

[25] C. Crowcroft, M. Handley, and I. Wakeman. *Internetworking Multimedia*. UCL Press, London, 1999.

[26] R. Flenner. *Jini and JavaSpaces Application Development*. SAMS, first edition, 2001.

[27] J. Jang. An approach to designing reusable service frameworks via virtual service machine. In *Proceedings of the Symposium on Software reusability (SSR'01)*, pages 58–66. ACM Press, 2001.

[28] N. Carriero and D. Gelernter. *How to write parallel programs*. MIT Press, 1990.

[29] N. Furmento, W. Lee, A. Mayer, S. Newhouse, and J. Darlington. ICENI: an open grid service architecture implemented with Jini. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, pages 1–10. IEEE Computer Society Press, 2002.

[30] N. Furmento, J. Hau, W. Lee, S. Newhouse, and J. Darlington. Implementations of a Service-Oriented Architecture on top of Jini, JXTA and OGSI. In *Proceedings of the Second Across Grids Conference*, pages 90–99. Springer-Verlag, 2004.

[31] L. Fuentes and J.M. Troya. Towards an open multimedia service framework. *ACM Computing Surveys (CSUR)*, 32(1):24–29, 2000.

[32] F. Curbera, M. J. Duftler, R. Khalaf, W. A. Nagy, N. Mukhi, and S. Weerawarana. Colombo: Lightweight middleware for service-oriented computing. *IBM Systems Journal*, 44(4):799–820, 2005.

[33] H. Bohn, A. Bobek, and F. Golatowski. SIRENA - service infrastructure for real-time embedded networked devices: A service oriented framework for

Entropy-Driven Parameter Control for Evolutionary Algorithms

Shih-Hsi Liu
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294, USA
liush@cis.uab.edu, http://www.cis.uab.edu/liush

Marjan Mernik
Faculty of Electrical Engineering and Computer Science
University of Maribor
2000 Maribor, Slovenia
marjan.mernik@uni-mb.si, http://lpm.uni-mb.si/mernik

Barrett R. Bryant
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294, USA
bryant@cis.uab.edu, http://www.cis.uab.edu/bryant

Keywords: entropy, evolutionary algorithms, exploration, exploitation, PPC_{EA}

Received: November 3, 2006

Every evolutionary algorithm needs to address two important facets: exploration and exploitation of a search space. Evolutionary search must combine exploration of the new regions of the space with exploitation of the potential solutions already identified. The necessity of balancing exploration with exploitation needs to be intelligent. This paper introduces an entropy-driven parameter control approach for exploring and exploiting evolutionary algorithms. Entropy represents the amount of disorder of the population, where an increase in entropy represents an increase in diversity. Four kinds of entropy to express diversity and to control the entropy-driven approach are discussed. The experimental results of a unimodal, a multimodal with many local minima, and a multimodal with only a few local minima functions show that the entropy-driven approach achieves good and explicit balance between exploration and exploitation.

Povzetek: V članku je opisan adaptiven način krmiljenja raziskovanja in izkoriščanja v evolucijskih algoritmih, voden s pomočjo entropije.

1 Introduction

Evolutionary Algorithms (EAs) [2, 12] are a common term for solving problems with computers that uses models and mechanisms from biological evolution. Such nature inspired EAs simulate evolution and its mechanisms such as selection, crossover, and mutation. Most well known examples of EAs are Genetic Algorithms (GAs), Evolution Strategies (ESs), Evolutionary Programming (EP), and Genetic Programming (GP) [12]. They have been used successfully for planning, design, simulation and identification, controlling, classification, and for solving many other hard optimization problems. EAs are general purpose search methods with good yet implicit balance between exploration and exploitation. Exploration is a process of visiting entirely new regions of a search space and seeing if anything promising may be found in the regions. Exploitation is a process of using information gathered from the previously visited points in the search space to determine which regions might be profitable to be visited next. Additionally, exploitation techniques are good at finding local optima. However, how is the balance between exploration and exploitation achieved in EAs? More importantly, how can the balance be controlled?

In EAs, the selection process, operators (e.g., crossover and mutation), and population size establish a balance between the exploration and exploitation of the search space [6]. The selection process drives search towards the regions of the best individuals. Hence, exploitation is done by selection. However, Bäck [1] showed that the selection processes can control the level of exploration or exploitation by varying selection pressure. Higher selection pressure pushes the search towards more exploitation and lower selection pressure urges the search towards more exploration.

A mutation operator randomly modifies individuals, with a given probability, and thus increases the structural diversity of the population. From this point of view, the mutation operator is more an exploration operator. Such an operator helps to recover the genetic diversity lost during the selection phase and to explore new solutions avoiding premature convergence. Conversely, mutation can also be seen as an exploitation operator, because most of the genetic material is preserved. However, note that in some EAs (e.g., evolution strategies) mutation has a much bigger exploration role than in genetic algorithms. The crossover operator combines two or more parents to generate better offspring. Such a combination can be derived from the idea that the exchange of information between good individuals will generate even better offspring. From this point of view, the crossover operator is more an exploitation operator. However, a good crossover operator should also generate individuals in the exploration zone. Directing the evolutionary process towards exploration or exploitation is also possible by population resizing [9]. With bigger population size, the search space is more explored than with smaller population size. Therefore, good balance between exploration and exploitation in EAs is achieved by selection, good mutation and crossover operators and by determining parameters (e.g., p_m, p_c, tournament size, population size), which control mutation, crossover, and selection, respectively.

There have been a variety of studies on determining the best control parameter values [4, 5]. The main problem is to find a set of control parameters, which optimally balances exploration and exploitation: if crossover and mutation rates are very high, much of the space will be explored, but there is a high probability of losing good solutions and of failing to exploit existing schema. If crossover and mutation rates are low, the search space is not explored. The population diversity is therefore rapidly decreasing and ending up in a premature convergence to a non-optimal solution. Despite that, many researchers believed that EAs are effective because of a good ratio between exploration and exploitation. In EAs, however, this ratio is implicitly controlled. In some other search techniques such as reinforcement learning [18], one has explicit control over exploration and exploitation. In EAs, one no longer has explicit and respective control over exploitation and exploration, because it is difficult to delimit exploration from exploitation.

In this paper, an entropy-driven exploration and exploitation approach is presented. The exploration/exploitation of the search space is adapted on-line based on the current status of the evolutionary process. The on-line adaptation mechanism involves a decision process as to whether more exploitation or exploration is needed depending on the current progress of the algorithm and on the current estimated potential of discovering better solutions. This decision process is described in a metaprogramming fashion using a domain-specific language, PPCEA (Programmable Parameter Control for Evolutionary Algorithms) [10]. Because of space consideration, the paper only presents the experimental results using genetic algorithms. Experimenting the mutation role for balancing between exploration and exploitation in evolution strategies is our future work.

The paper is organized as follows. Section 2 describes the related work. In Section 3, four kinds of entropy are introduced to control exploration and exploitation. Section 4 shows the experimental results on the benchmark functions. Finally, Section 5 presents the conclusion.

2 Related Work

Optimal balance between exploration and exploitation has been mainly controlled by determining the best control parameter values. There are a variety of studies on this topic [5, 8, 10]. Recommendations on control parameters for a particular set of problems can be found in [4, 15]. In [5], an overview of this problem has been given, where the authors distinguish between parameter tuning and parameter control. Furthermore, methods for parameter control have been classified into deterministic, adaptive, and self-adaptive categories: the deterministic category adjusts parameters by deterministic rules; the adaptive category utilizes the feedback of the evolutionary process to control the direction and magnitude of parameters; and the self-adaptive category encodes parameters into individuals and undergoes mutation and recombination. An example of how to balance between exploration and exploitation by parameter control is described as follows. As soon as an algorithm approaches the optimum, the mutation step size must be decreased to balance the probability of generating a new successful point. A simple idea is to decrease the mutation step size s by a deterministic schedule such as $s_t = s_0/t$ or $s_t = \beta^t \cdot s_0$, where $\beta \in (0, 1)$.

One of the earliest researchers that investigated entropy in EAs was Rosca [14], whose experiments showed that populations appeared to be stuck in local optima when entropy did not change or decrease monotonically in successive generations. Rosca used fitness values in a population to define entropy and free energy measure. Our work extends Rosca's in trying to find different ways to compute entropy in EAs. Moreover, using entropy as a diversity measure and metaprogramming parameter control by PPCEA [10], we are able to control exploration and exploitation in an adaptable manner.

The Diversity-Guided Evolutionary Algorithm (DGEA) [17] uses a distance-to-average-point measure to alternate between phases of exploration and exploitation. It can be expressed easily as a PPCEA program. Moreover, DGEA does not use entropy as a measure for diversity.

In [11], entropy is introduced into EAs for determining the optimal number of clusters. However, in this case the fitness function is entropy-based.

3 Entropy in Evolutionary Algorithms

Entropy is a concept in thermodynamics, information theory, and statistical mechanics. The thermodynamic entropy S is a measure of the amount of energy in a physical system that cannot be used to do work. As such, it is also a measure of the disorder and randomness presented in a system. The entropy depends not only on the current state of the system, but also its history. Therefore, it is a state function of the parameters (e.g., pressure and temperature), which describe the observable macroscopic properties of the system. The macroscopic state of the system is defined by a distribution on the microstates that are accessible to a system in the course of its thermal fluctuations. Entropy S of the system is defined as:

$$S = -k_B \sum_i p_i \ln p_i \tag{1}$$

where k_B is a physical constant known as Boltzmann's constant, i is the energy of microstate, and p_i is the probability that it occurs during the system's fluctuations.

The basic concept of entropy in information theory has to do with how much randomness there is in a signal or random event. Shannon [16] defines entropy in terms of a discrete random event x, with possible states $1..n$ as:

$$H(x) = \sum_i^n p_i \log_2(\frac{1}{p_i}) = -\sum_i^n p_i \log_2 p_i \tag{2}$$

Statistical mechanics explains entropy as the amount of uncertainty which remains about a system, after its observable macroscopic properties have been taken into account. For a given set of macroscopic quantities, such as temperature and volume, entropy is a function of the probability that the system is in various quantum states. The more states available to the system with higher probability, the greater the disorder and thus, the greater the entropy. If the system has only one possible state, there is no uncertainty, and the entropy of the system is zero. If the system has n possible states which are equiprobable ($p_i = \frac{1}{n}$), the entropy is the highest:

$$H = -n\frac{1}{n}\log_2(\frac{1}{n}) = \log_2 n \tag{3}$$

As such, entropy represents also a succinct measure of diversity. Biological diversity refers to the differences between individuals in a population, which in nature imply structural (genotype) and behavioral (phenotype) differences. In EAs, identical genotypes produce the same fitness. Thus, a decrease in genotype diversity will necessarily cause a decrease in phenotype diversity. Hence, to measure entropy/diversity, one needs to define some structural measures. Such measures, however, might be computationally intensive in some instances of EAs (e.g., genetic programming) [3]. Fortunately, based on the described entropy/diversity relationship between genotype and phenotype, such measures at the phenotype level are sufficient.

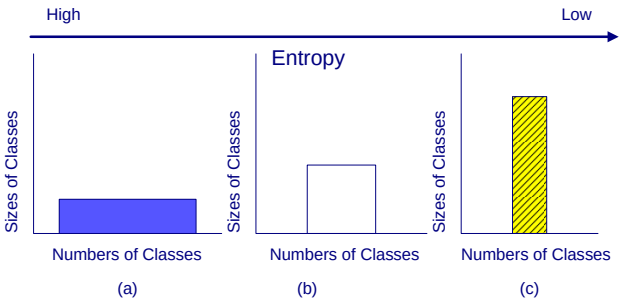


Figure 1: The relationship between entropy and the numbers and sizes of classes

Figure 1 shows how the numbers and sizes of classes of a population affect entropy. High entropy in EAs reveals the presence of many unique fitness values, where the population is evenly distributed over those values, as shown in Figure 1 (a). Figure 1 (c) represents low entropy computed from a population which contains fewer unique fitness values as many individuals have the same fitness.

Rosca [14] calculates entropy for a population by first placing fitness values into fitness classes p_i and counting the size of each fitness class. p_i is the proportion of the population occupied by the population partition i. Entropy is then defined as:

$$-\sum_i p_i \log_2 p_i \tag{4}$$

This paper extends [14] to experiment with entropy, using different flexible cases of fitness classes, to facilitate explicit balance between exploration and exploitation.

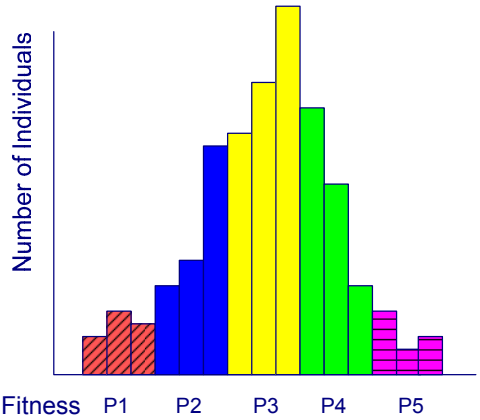


Figure 2: Fitness classes of linear entropy

Figures 2, 3, and 4 show three new cases for defining fitness classes:

– Linear: Assign a predefined yet changeable value to the number of fitness classes, n. For each generation, the interval between the best and worst fitness values is evenly partitioned into n sub-intervals as fitness

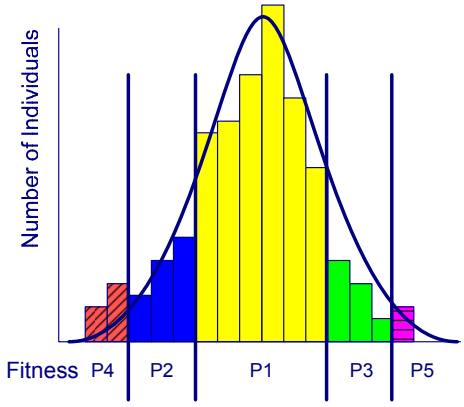


Figure 3: Fitness classes of Gaussian entropy

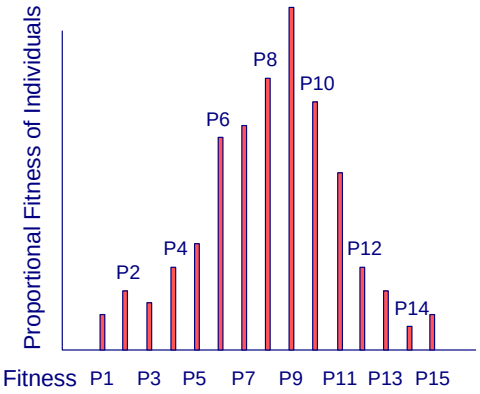


Figure 4: Fitness classes of fitness proportional entropy

classes (Figure 2). An individual whose fitness value is occupied in a specific sub-interval is classified into the corresponding fitness class. The concept of linear fitness classes is adapted from [14]. Changeable n and various upper and lower bounds of each generation (i.e., the best and worst fitness values) are the two key differences between our approach and Rosca's.

– Gaussian: The partition of fitness classes in this case is derived from Gaussian distribution, as shown in Figure 3. For each generation, fitness classes are "spread out" from the average fitness value ($average$) with the standard deviation (σ). For example, the upper/lower bound of the first fitness class (P1 in Figure 3) is computed as $average$ +/- σ. The boundaries of the successive classes (P2 - P5) can be generalized as $average$ +/- i*σ, where $i \in$ Z+ and $i \leq n/2$. For each generation, the lower bound of the leftmost fitness class is less than or equal to the smallest fitness value, and the upper bound of the rightmost fitness class is larger than or equal to the biggest fitness value.

– Fitness proportional: The fitness proportional approach is a variation of Rosca' approach [14]. Rosca's fitness classes are partitioned by individuals having the same phenotypes. p_i is the proportion of a population occupied in the i^{th} partition. In our approach, p_i is formalized as $f_i / \sum_i^{Popsize} f_i$, where f_i is the fitness value of an individual. p_i is the criterion for categorizing fitness classes. As all individuals of a population have different p_i (namely, different fitness values), the number of fitness classes n equals the population size ($Popsize$). If more than one individual has the same fitness value (i.e., $p_i = p_j$, where $i \neq j$), $p_j \cdot \log_2 p_j$ is eliminated in the Equation (1) and $n < Popsize$. It is because two identical fitness classes are not needed, and the elimination complies with the definition of diversity. Figure 4 shows 15 fitness classes sorted by p_i, each of which has one or more individuals occupied.

The next section exercises linear, Gaussian, fitness proportional and Rosca entropies for the entropy-driven approach and compares the experimental results with the Fogarty [7], Schaffer [15], and 1/5 success rule [12] approaches.

4 Experiments

Entropy-driven exploration and exploitation have been experimented with on the suite of test functions presented in [19]. Due to lack of space, only examples of the *Sphere Model* (f_1), *generalized Rastrigin's function* (f_9), and *Branin function* (f_{17}) are presented in this section. To illustrate all the experiments easily, Best fitness value (B), Average fitness value (A), Worst fitness value (W), Population Diversity (D), Standard Deviation (S), Linear Entropy (E), Gaussian Entropy (G), Fitness Proportional Entropy (P), and Rosca Entropy (R) with respect to a population from generations 0 to maximum generation (X-axis) are included in the following figures. Curves B, A, and W use the same definitions as all other EAs; curves E, G, and P are defined in Section 3; curve S is the standard deviation of the fitness values of all individuals; curve D is the Euclidean distance between all individuals; and curve R is the entropy defined in [14]. All but entropy curves (E, G, P, and R) use the left Y-axis as the coordinate. Table 4 shows the initial values setup (we used the same setting as in [19]) for the following experiments: f_1, f_9, and f_{17} have different maximum generation ($Maxgen$) settings; $Popsize$ is the population size; p_m and p_c are mutation and crossover rates; $Epoch$ is the stride of parameter adjustments during the evolutionary process; and $Round$ is the number of experiments for each example, and the experimental results in subsequent figures are the average values out of 50 rounds.

Sections 4.1, 4.2 and 4.3 respectively present f_1, f_9, and f_{17} with their experimental results of the Fogarty [7], Schaffer [15], 1/5 success rule [12], and entropy-driven approaches. Only two figures of each function are selected in the paper. All of the experimental results with the corresponding figures may be found at the PPCEA website [13].

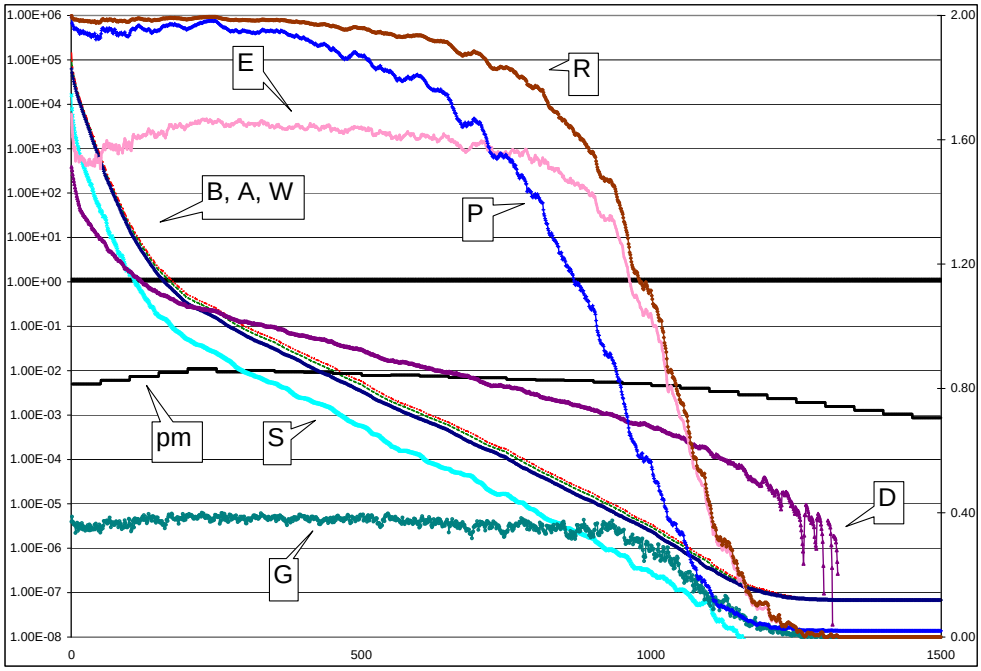


Figure 5: 1/5 success rule approach for f_1

Parameter	*Value*	*Parameter*	*Value*
Maxgen (f_1)	1500	Maxgen (f_9)	5000
Maxgen (f_{17})	100	Popsize	100
p_m	0.005	p_c	0.75
Epoch	50	Round	50

Table 1: Initial values of parameters in experiments on functions f_1, f_9, and f_{17}

4.1 The Sphere Model

The Sphere Model (f_1) is a unimodal function as shown in Equation (5).

$$f_1(x) = \sum_{i}^{d} x_i^2 \tag{5}$$

where $x_i \in [-100, 100]$, d (dimension) $= 30$, and min(f_1) $= f_1(0,...,0) = 0$.

The first presented experiment is the parameter tuning approach using the Schaffer parameter setting ($p_m = 0.005$ and $p_c = 0.75$). The mean best value and convergence rate[1] are $6.82 \cdot 10^{-8}$ and 830, respectively. The Fogarty approach is a deterministic one that initializes p_m = 0.11375 and adjusts the value using the Fogarty formula [7]. The mean best value is $2.13 \cdot 10^{-5}$ at generation 765. Figure 5 presents the results using the 1/5 success rule [12]. Such a rule determines p_m to be increased when the successful permutation rate (φ) is greater than 1/5, and to be decreased when φ is less than 1/5. In Figure 5, a good balance between exploration and exploitation (yet still more on exploration) can be found before generation 900: curves E and R are stable in the ranges between 1.4 and 1.65 and between 1.55 to 2.00, respectively; curves B, A, W, S, and D are smoothly decreased; and p_m is changed every 50 generations to adjust the mutation step. From generations 900 to 1220, curves E and R steeply decline, and curve G has downhill move. Such curves show that the evolutionary process is inclined from exploring to exploiting the current regions with relatively small mutation steps. From generations 1220 to 1320, all entropy curves are getting flat and curve D has a "saw-toothed" shape. Such curves imply that the searching process is in the exploitation phase and is not stuck in local optima. The best value found using the 1/5 success rule approach is $6.82 \cdot 10^{-8}$ at generation 1274.

Before examining the last chart, an entropy-driven approach written in PPCEA is shown in Figure 6. When entropy is greater than 0.5, p_m is decreased to facilitate the exploitation phase. Smaller mutation steps avoid the increase of population diversity. As entropy is smaller than 0.5, more exploration is required to avoid local optima. Therefore, p_m is increased to diversify the search regions. Such an example shows that balance between exploration and exploitation can be adjusted in synergy of entropy and p_m. Figure 7 shows the result using this source code.

In Figure 7, curves E, P, and R steeply decline between generations 0 and 450. Curves B, A, W, S, and D also diagonally traverse the plane. When curve E is between its midpoint (at generation 350) and upper bound (0.74 to 1.68), p_m is decreased (line 9 of the PPCEA code) to balance exploitation against exploration. As curve E is between its lower bound and midpoint (0 to 0.74), exploration outperforms exploitation by raising p_m. This phenomenon can be observed from curve D that declines more steeply and has a

[1]The point that curve Best becomes flat in the figure.

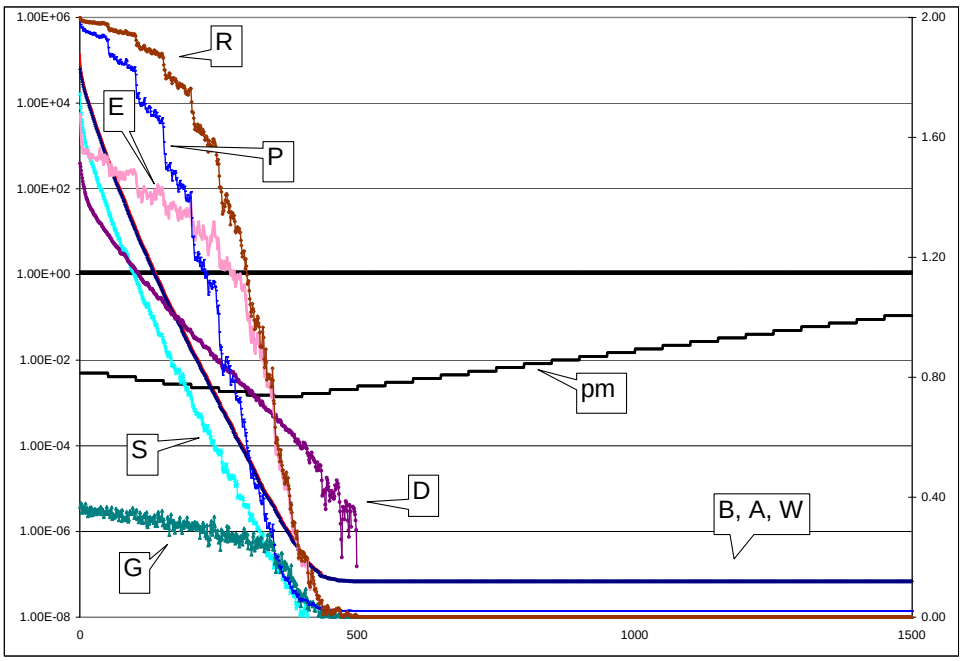


Figure 7: Entropy-driven approach for f_1

```
1 genetic
2    g := 0;
3    while ( g < Round ) do
4       t:=0;
5       init;
6       while ( t < Maxgen ) do
7          callGA;
8          if ( Entropy > 0.5 ) then
9              pm := pm * 0.82
10         fi;
11         if ( Entropy < 0.5 ) then
12             pm := pm * 1.22
13         fi;
14         t := t + Epoch
15      end;
16    g:= g + 1
17    end
18 end genetic
```

Figure 6: Entropy-driven parameter control written in PPC_{EA}

drastic "saw-toothed" shape from generations 400 to 500. Curve R is similar to curve E in terms of the shapes and the balance between exploration and exploitation. The best value found is the same as in the 1/5 success rule. However, please note that the convergence is much faster in the entropy-driven approach (at generation 467). Hence, many fitness evaluations after 467 generations can be skipped.

4.2 Generalized Rastrigin's Function

Generalized Rastrigin's Function (f_9) is a multimodal function with many local minima as shown in Equation (6).

$$f_9(x) = \sum_{i}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10] \qquad (6)$$

where $x_i \in [-5.12, 5.12]$, d (dimension) $= 30$, and min(f_9) $= f_9(0,...,0) = 0$.

For the Schaffer approach for f_9 (figure in [13]), exploration is still carried out energetically after generation 1000 in the search space comprising many local minima. Because of the late vivid exploration, the best optimal solution is still improved slightly (20.86) until the evolutionary process converges at generation 3988. Figure 8 (i.e., the experimental results of the Fogarty approach) is a good example to represent that the process is stuck at the local minima. The figure shows that p_m may decrease too fast to perform enough exploration. After generation 400, entropy curves (i.e., curves E, G, P, and R) and fitness curves (i.e., curves B, A, and W) are nearly static, yet diversity curves (i.e., curves D and S) exhibit extreme shakiness. This phenomenon implies that even though the exploration is still active, the relatively small p_m does not provide enough exploration power to assist the evolutionary process to jump out the local optima. Hence, the experimental results of the Fogarty approach are the worst among the four approaches (40.55 at generation 4079).

The characteristic of exploiting many local minima can be also examined in the results of the 1/5 success rule (figure in [13]). However, because of the inefficient exploration power determined by the small p_m value at the later stage, there is no exploration or exploitation activity

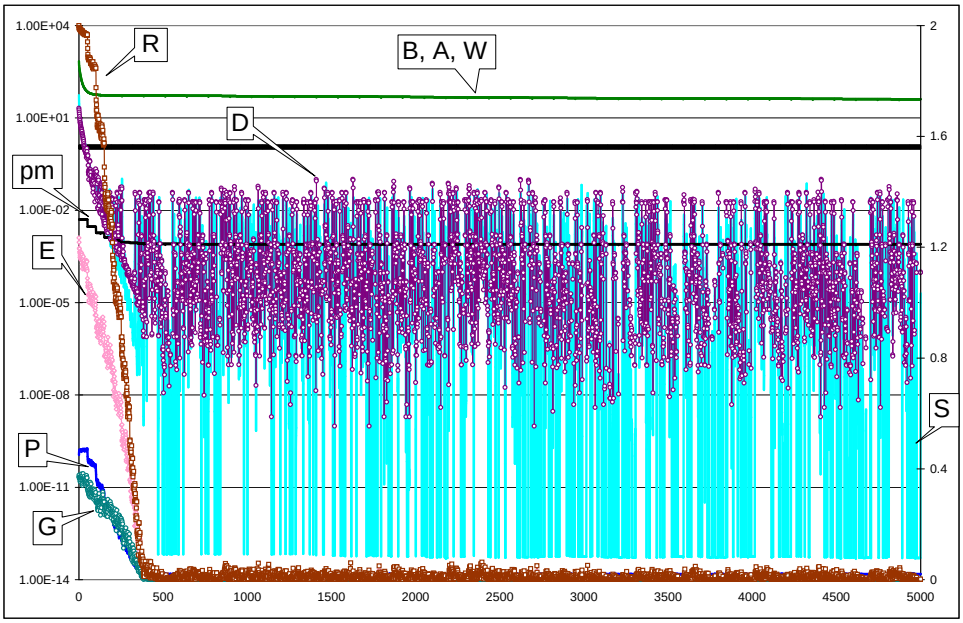


Figure 8: Fogarty approach for f_9

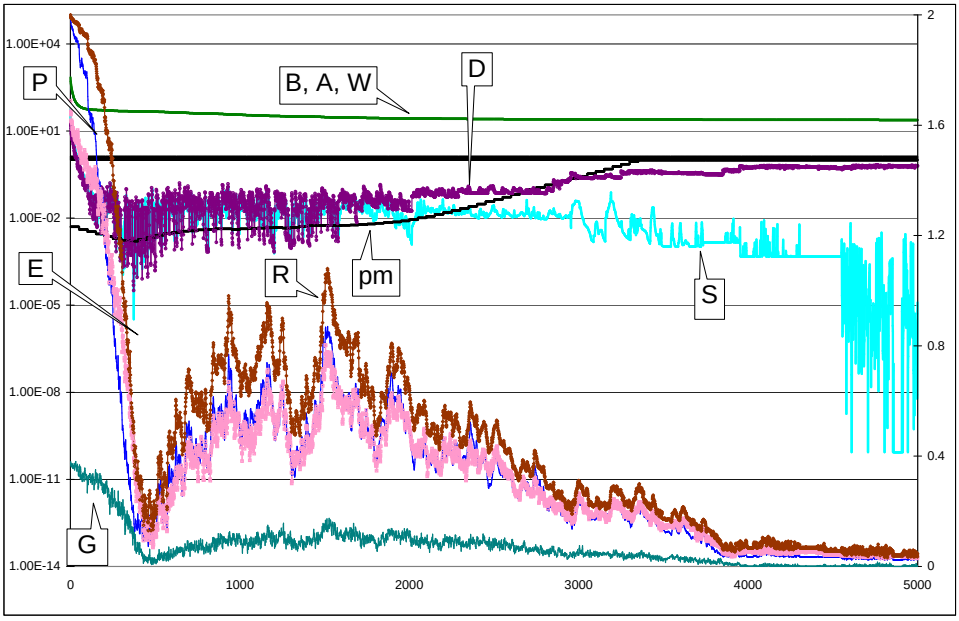


Figure 9: Entropy-driven approach for f_9

observed. The mean best value and convergence rate are 25.24 and 1265, respectively. Figure 9 shows a good case of balance between exploration and exploitation using the entropy-driven approach in the case of multimodal function. In the chart, the evolutionary process starts at inclining from more exploration toward more exploitation driven by declining p_m before generation 320. From generation 300 to 850, the rising p_m facilitates more exploration to discover many local minima. In this phase, the same or better values may be found and selected. Entropy curves and diversity curves are therefore updated drastically. Most importantly, because of the exploration on the search space of local minima, fitness values are still slightly improved (23.99) at the very late phase (generation 3023).

4.3 Branin Function

The Branin Function (f_{17}) is a multimodal function with only a few local minima as shown in the following equation.

$$f_{17}(x) = [x_2 - (5.1x_1^2)/(4\pi^2 + 5x_1/\pi - 6]^2 + 10[1 - 1/(8\pi)]cosx_1 + 10 \quad (7)$$

where $x_1 \in [-5, 10]$ and $x_2 \in [0, 15]$, d (dimension) $= 30$, and $\min(f_{17}) = f_{17}(0,...,0) = 0.398$.

Because there are only a few local minima in f_{17} given a small maximum generation number, the evolutionary process cannot be guaranteed to discover all of the local optima using the Schaffer approach (i.e., parameter tuning problem). In Figure 10, diversity curves appearing again after generations 30, 66, 76, 83 and 90 show that a few local optima are found in this phase. Fortunately, the evolutionary process still possesses enough exploration power to improve the value of mean best value (0.421 at generation 90). Similar to the Schaffer results, the Fogarty approach for f_{17} also generates small refinements for the mean best value (0.432) at the late stage (generation 80). However, the slightly different results between the two approaches may be derived from the early decreasing p_m in the Fogarty approach. Please refer to [13] for the enlarged Figure 10 and the numerical improvement of mean best value that may not be observed in Figure 10. For the 1/5 success rule, because the success mutation ratio is always below an ideal value, 0.2, the entire process inclines towards exploitation by reducing p_m. The mean best value (0.434) is close to the Fogarty approach. However, because of different formulae for adjusting p_m, the 1/5 success rule converges at generation 59, which is much earlier than the Fogarty approach.

Although f_{17} has a few local maxima, the entropy-driven approach still performs a good balance between exploration and exploitation as well as finding even better solutions at the end of the evolutionary process. Figure 11 presents similar characteristics (i.e., rising p_m, drastic changing entropy curves, and decreasing fitness value curves) as Figure 10. The mean best value is 0.398 at generation 100.

The experimental results on all benchmark functions indicate that the linear and Rosca approaches for defining fitness classes are superior to Gaussian and fitness proportional ones in terms of providing the information for balancing exploration and exploitation.

5 Conclusion and Future Work

The opinions on the research on exploration and exploitation are still widely divided [5]. In this paper, we introduce a novel entropy-driven exploration and exploitation approach. The balance between exploration and exploitation is fulfilled by the synergy of p_m, p_c and entropy online. The on-line adaptation mechanism involves PPCEA as to whether more exploitation or exploration is needed depending on the current progress of the algorithm and on the current estimated potential of discovering better solutions. The experimental results in all figures show that our approach can easily interpret the influence of exploration and exploitation using curve E and auxiliary curves.

Experiments with the entropy-driven exploration and exploitation approach for evolution strategies [12] are planned. Additionally, a more generic PPCEA that manipulates more similar related work (e.g., Harik's parameterless genetic algorithm [9]) will benefit the community of evolutionary computation.

References

[1] T. Bäck. *Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms*. Proc. 1^{st} IEEE Conf. on Evolutionary Computing, pages 57-62, 1994.

[2] T. Bäck, D.B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation*. University of Oxford Press, 1996.

[3] E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor. *Advanced Population Diversity Measures in Genetic Programming*. Parallel Problem Solving from Nature - PPSN VII, Springer-Verlag LNCS, No. 2439, pages 341-350, 2002.

[4] K. De Jong. *The Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Ph.D. thesis, Department of Computer Science, University of Michigan, Ann Arbor, Michigan, 1975.

[5] A. Eiben, R. Hinterding, and Z. Michalewicz. *Parameter Control in Evolutionary Algorithms*. IEEE Trans. on Evolutionary Computation, Vol. 3, No. 2, pages 124-141, 1999.

[6] A. Eiben and C. Schippers. *On Evolutionary Exploration and Exploitation.* Fundamenta Informaticae, No. 35, pages 35-50, 1998.

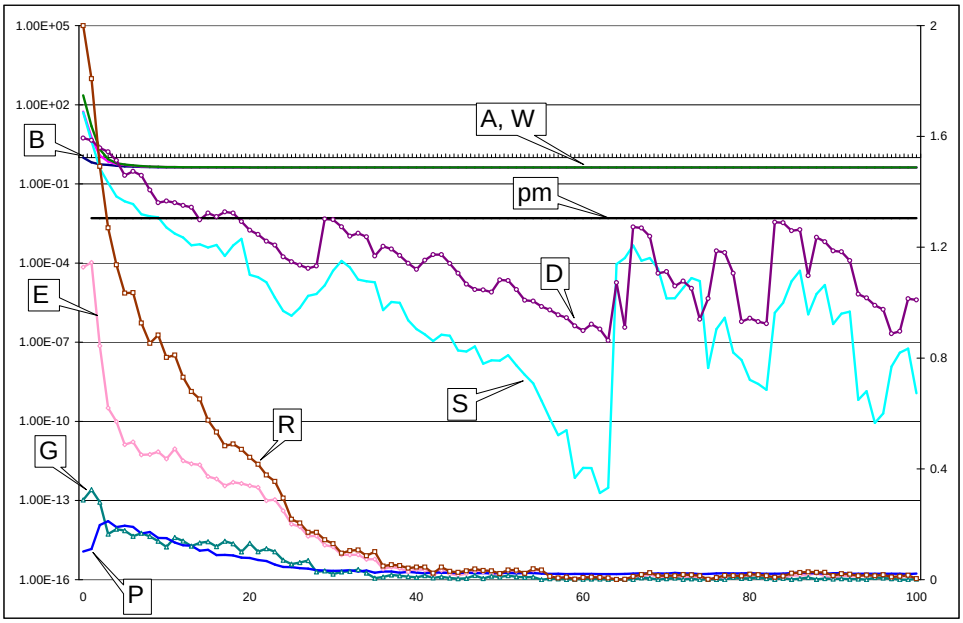


Figure 10: Schaffer approach for f_{17}

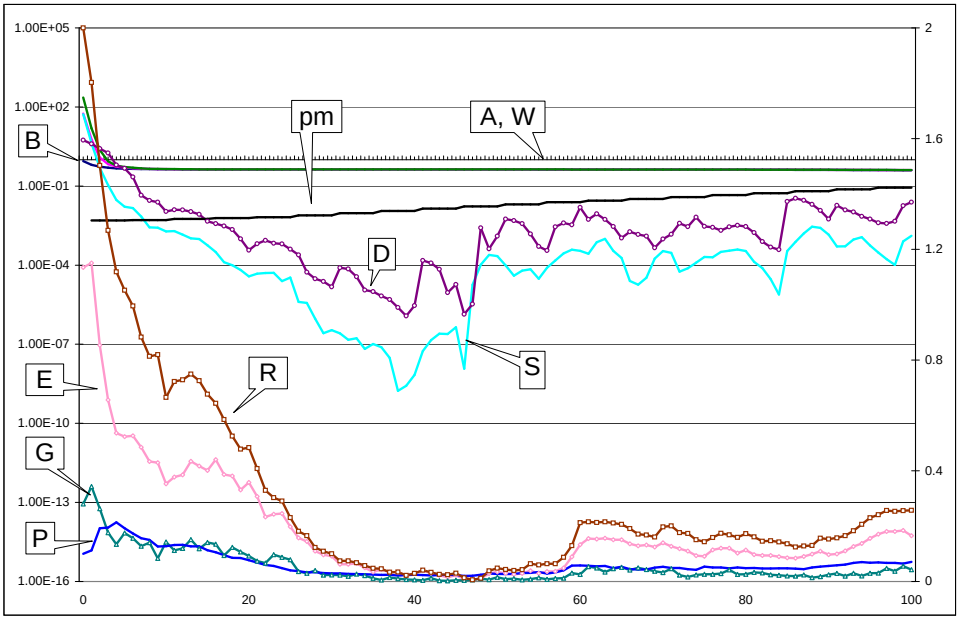


Figure 11: Entropy-driven approach for f_{17}

[7] T.C. Fogarty. *Varying the Probability of Mutation in the Genetic Algorithm.* Proc. 3^{rd} Intl. Conf. on Genetic Algorithms, pages 104-109, 1989.

[8] J.J. Grefenstette. *Optimization of Control Parameters for Genetic Algorithms*. IEEE Trans. on Systems, Man & Cybernetics SMC-16, No. 1, pages 122-128, 1986.

[9] G. Harik and F. Lobo. *A Parameter-less Genetic Algorithm*. Technical Report IlliGAL 9900, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1999.

[10] S.-H. Liu, M. Mernik, and B.R. Bryant. *Parameter Control in Evolutionary Algorithms by Domain-Specific Scripting Language PPCea*. Proc. 1^{st} Intl. Conf. on Bioinspired Optimization Methods and their Applications, pages 41-50, 2004.

[11] W. Lu, and I. Traoré. *A New Evolutionary Algorithm for Determining the Optimal Number of Clusters*. Proc. 17^{th} Intl. Conf. on Tools with Artificial Intelligence, pages 712-713, 2005.

[12] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. 3^{rd} ed., Springer-Verlag, 1996.

[13] PPCea: A Domain-Specific Language for Evolutionary Algorithms.
http://www.cis.uab.edu/liush/PPCea.htm

[14] J. Rosca. *Entropy-Driven Adaptive Representation*. Proc. of the Workshop on Genetic Programming: From Theory to Real-World Applications, pages 23-32, 1995.

[15] J.D. Schaffer et al. *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization*. Proc. 3^{rd} Intl. Conf. on Genetic Algorithms, pages 51-60, 1989.

[16] C. Shannon. *A Mathematical Theory of Communication.* Bell Systems Technical Journal, Vol. 27, pages 379-423, 623-656, 1948.

[17] R. Ursem. *Diversity-Guided Evolutionary Algorithms*. Parallel Problem Solving from Nature - PPSN VII, Springer-Verlag LNCS, No. 2439, pages 462-471, 2002.

[18] S. Whitehead. *Learning from Delayed Rewards.* Ph.D. thesis, King's College, Cambridge University, England, 1992.

[19] X. Yao, Y. Liu, and G. Lin. *Evolutionary Programming Made Faster.* IEEE Trans. on Evolutionary Computation, Vol. 3, No. 2, pages 82-102, 1999.

Stopping Criteria for a Constrained Single-Objective Particle Swarm Optimization Algorithm

Karin Zielinski and Rainer Laur
Institute for Electromagnetic Theory and Microelectronics, University of Bremen, Germany
{zielinski,rlaur}@item.uni-bremen.de

Keywords: constrained single-objective optimization, evolutionary algorithms, particle swarm optimization, stopping criteria

Received: November 3, 2006

When using optimization algorithms the goal is usually clear: The global optimum should be found. However, in general it is not clear when this goal is achieved, especially if real-world problems are optimized for which no knowledge about the global optimum is available. Therefore, it is not easy to decide when the execution of an optimization algorithm should be terminated. Although different mechanisms can be used for the detection of an appropriate time for ending an optimization run, only two of them are frequently used in the literature. Unfortunately, both methods have disadvantages, particularly for the optimization of real-world problems. Because especially for practical applications it is important when an optimization algorithm is terminated as they usually contain computationally expensive objective functions, the performance of several stopping criteria that react adaptively to the state of an optimization run is evaluated for a Particle Swarm Optimization algorithm in this work. The examination is done on the basis of a constrained single-objective power allocation problem. Suggestions from former work concerning stopping criteria for unconstrained optimization are verified and comparisons with results for Differential Evolution are made.

Povzetek: Ovrednoteni so ustavitveni kriteriji za optimiranje z roji delcev (angl. particle swarm optimization) in rezultati primerjani z rezultati algoritma diferencialne evolucije.

1 Introduction

Evolutionary algorithms (EAs) are a class of population-based stochastic optimization algorithms that incorporate mechanisms from evolution for optimization processes. The most famous representatives from this class are possibly Genetic Algorithms [5] but in the last years also e.g. Particle Swarm Optimization (PSO) [7] and Differential Evolution (DE) [9] had a lot of success.

For theoretical aspects of evolutionary algorithms stopping criteria are usually not important. However, for practical applications the choice of stopping criteria can significantly influence the duration of an optimization run. Due to different stopping criteria an optimization run might be terminated before the population has converged, or computational resources might be wasted because the optimization run is terminated late. Real-world problems mostly contain computationally expensive objective functions that may result in optimization runs that take several days, thus wasting of computational resources has to be prevented.

In the literature mostly two stopping criteria are applied in single-objective optimization: Either an error measure in dependence on the known optimum is used or the number of function evaluations is limited to fe_{max}. These criteria are perfectly suitable for e.g. comparing the performance of different algorithms but for solving real-world problems there are some drawbacks. The first mentioned method has the disadvantage that the optimum has to be known, so it is generally not applicable to real-world problems because the optimum is usually not known a priori. The second method is highly dependent on the objective function. Because generally no correlation can be seen between an optimization problem and the required number of function evaluations, fe_{max} has to be determined by trial-and-error methods usually. Evolutionary algorithms include randomness in the optimization process, thus the number of function evaluations that is needed for convergence is subject to fluctuations, so a safety margin for fe_{max} is needed. The fluctuations can be significant as can be seen in [17] where a test suite of 24 functions has been examined, and the standard deviation of function evaluations for reaching a predefined error measure was up to 180,000. If a real-world problem with an unknown optimum would result in a similar standard deviation, it would be difficult to choose fe_{max}.

As a result, it would be better to use stopping criteria that consider knowledge from the state of the optimization run. The time of termination would be determined adaptively, so function evaluations could be saved.

Several stopping criteria are reviewed in [19] and [20] that are sensitive to the state of the optimization run by observing the improvement, movement or distribution of the population members. In [19] stopping criteria are tested for unconstrained single-objective optimization using Particle Swarm Optimization and Differential Evolution, while in

[20] the criteria have been adapted for constrained single-objective problems using DE because real-world problems normally include constraints. In this work it will be examined if the suggestions regarding stopping criteria for PSO from [19] hold for the constrained real-world problem of optimizing a power allocation scheme. Additionally, a comparison with the results for DE in [20] will be done.

This work is organized as follows: In Section 2 related work is discussed. In Section 3 the Particle Swarm Optimization algorithm is described and Section 4 provides a short introduction to Differential Evolution. In Section 5 the stopping criteria that are used in this work are reviewed. In Section 6 results are shown and Section 7 closes with conclusions.

2 Related Work

Every optimization algorithm includes a stopping rule but there are only few works concentrating explicitly on stopping criteria. In [16] convergence of a Particle Swarm Optimization algorithm is detected by computing a maximum swarm radius, by doing a cluster analysis or by calculating the rate of change in the objective function. Most stopping criteria are applicable not only to PSO but also to other population-based optimization algorithms, e.g. in [1] the difference between maximum and minimum objective function value is used as stopping criterion for a Differential Evolution algorithm. In [13] not only termination criteria for evolutionary algorithms but also for other optimization algorithms are discussed. Often criteria similar to the ones used in the work are also applied in hybrid algorithms to determine the moment when global search should be replaced by local search [4, 6, 15].

3 Particle Swarm Optimization

Particle Swarm Optimization is derived from the behavior of social groups like bird flocks or fish swarms. Although the "survival of the fittest" principle is not used in PSO, it is usually considered as an evolutionary algorithm. A thorough discussion of this topic can be found in [7]. Like in this work, PSO is mostly used for the optimization of continuous functions.

Optimization is achieved by giving each individual in the search space a memory for its previous successes, information about successes of a social group and providing a way to incorporate this knowledge into the movement of the individual. Therefore, each individual (called particle) is characterized by its position $\vec{x}_i$, its velocity $\vec{v}_i$, its personal best position $\vec{p}_i$ and its neighborhood best position $\vec{p}_g$. Several neighborhood topologies have been developed [10]. In this work the *von-Neumann* topology is used as it showed promising results in the literature, e.g. in [8].

The dynamic behavior of PSO is generated by the update equations for velocity and position of the particles:

$$\begin{aligned}\vec{v}_i(t+1) &= w \cdot \vec{v}_i(t) \\ &\quad +c_1 r_1[\vec{p}_i(t) - \vec{x}_i(t)] \\ &\quad +c_2 r_2[\vec{p}_g(t) - \vec{x}_i(t)]\end{aligned} \tag{1}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{2}$$

Due to these equations the particles are drawn towards their personal best position and their neighborhood best position, and furthermore the velocity of the previous iteration is kept weighted with the inertia weight w. Other parameters are c_1 and c_2 which influence the impact of the cognitive and social component, respectively. To add a stochastic element to the movement of the particles, the numbers r_1 and r_2 are chosen randomly from the interval [0,1] in each iteration. Further parameters of PSO are the population size NP and the maximum velocity V_{max} that is used for preventing oscillations with increasing magnitude [7].

The control parameter settings for this examination are derived from a parameter study using the same optimization problem (yet unpublished): $w = 0.6$, $c_1 = 0.4$, $c_2 = 1.4$, $NP = 64$, $V_{max} = \frac{1}{2}(X_{max} - X_{min})$.

Constraint-handling is done by modifying the replacement procedure for personal and neighborhood best positions [11]. In unconstrained single-objective optimization a personal or neighborhood best position is replaced if the current position has a lower objective function value (for minimization problems as in this work). For constrained single-objective optimization this rule is altered so that in a comparison of two solutions $\vec{a}$ and $\vec{b}$, $\vec{a}$ is preferred if

- both vectors are feasible and $\vec{a}$ has a better objective function value or
- both vectors are infeasible and $\vec{a}$ has the lower sum of constraint violation or
- $\vec{a}$ is feasible and $\vec{b}$ is infeasible

where feasibility means that all constraints are satisfied. In contrast to several other constraint-handling techniques, no additional parameters are needed for this method [2]. For unconstrained problems the modified PSO algorithm is exactly the same as the original PSO.

4 Differential Evolution

The main characteristic of Differential Evolution is an adaptive scaling of step sizes that results in fast convergence behavior. Using DE the population members are evolved from one generation to the next by applying the operators mutation, recombination and selection. The first two operators generate new vectors by linearly combining several population members and afterwards exchanging some vector components. The third operator decides based on objective function values and constraint violation which vectors will be kept for the next generation. Because no deterioration with regard to the objective function value

is possible, the DE selection scheme is called greedy [14]. More specific information about the here mentioned DE algorithm can be found in [20].

5 Stopping Criteria

Stopping criteria are needed to terminate the execution of optimization algorithms. In contrast to using a maximum number of function evaluations as a stopping condition, other criteria have the advantage of reacting adaptively to the state of the optimization run, thus function evaluations can be saved. Unfortunately, it seems to be impossible to define a stopping criterion without introducing one or more parameters. The parameter settings generally depend on the given optimization problem. However, it should be investigated if there are stopping criteria for which the parameter settings are robust to changes or if parameters can be set depending on certain aspects of the problem. It is assumed that the general behavior of different optimization problems to stopping criteria is similar. It should be kept in mind that limiting the number of function evaluations as a stopping criterion also incorporates the choice of a problem-dependent parameter fe_{max}. Hence, it is favorable to examine other possibilities for stopping that contain the advantage of reacting adaptively to the state of the optimization run.

In the following the stopping criteria that incorporate information about the state of the optimization run are reviewed shortly. Note that there is a change compared to [19]: Instead of using the current positions $\vec{x}_i$ for the calculation of stopping conditions, the personal best positions $\vec{p}_i$ are used here. The reason is that the current positions have many fluctuations whereas the development of the personal best positions is more smooth, so decisions about termination of an optimization run should be easier.

Improvement-based criteria terminate an optimization run if only small improvement is made because usually in the beginning of an optimization run large improvements are achieved whereas in later stages the improvement becomes small. Three different conditions are used here:

- *ImpBest*: The improvement of the best objective function value is monitored. If it falls below a given threshold t for a number of generations g, the optimization run is terminated.
- *ImpAv*: Similar to *ImpBest*, but instead of observing the best objective function value, the average value computed from the whole population is checked.
- *NoAcc*: It is observed if any new $\vec{p}_i$ are accepted in a specified number of generations g. For DE this criterion is slightly different because in DE there are no personal best positions (instead, the acceptance of new population members is considered).

For *movement-based criteria* not the improvement but the movement of individuals is regarded. Two variants of movement-based criteria are considered that differ in the regarded space:

- *MovObj*: The movement of the individuals with respect to their objective function value (objective space) is examined if it is below a threshold t for a number of generations g. *MovObj* is different from *ImpAv* only if the regarded algorithm allows deterioration of the individuals' objective function value. This is the case for PSO in contrast to DE, but as $\vec{p}_i$ are considered here instead of $\vec{x}_i$, *MovObj* = *ImpAv* holds in this case also. Therefore, this criterion is not regarded further in this work.
- *MovPar*: The movement with respect to positions (parameter space) is checked if it is below a threshold t for a number of generations g.

The *distribution-based criteria* consider the diversity in the population. If the diversity is low, the individuals are close to each other, so it is assumed that convergence has been obtained.

- *StdDev*: It is checked if the standard deviation of positions is below a threshold m.
- *MaxDist*: The distance from every population member to the best individual is observed. The optimization run is stopped if the maximum distance is below a threshold m.
- *MaxDistQuick*: *MaxDistQuick* is a generalization of *MaxDist*. Instead of using the whole population for the computation of the maximum distance to the best population member, a quicksort algorithm is employed for sorting the individuals due to their objective function value, and only the best $p\%$ of the individuals are regarded. The background for this criterion is that there are optimization runs where most of the population has converged to the optimum but because of the remaining individuals which are still searching, the optimization run is not stopped although they do not contribute any new information. Using *MaxDistQuick* an optimization run can be stopped earlier than using *MaxDist*, so wasting of computational resources is avoided. However, the percentage p must not be set too low for a reliable detection of convergence.
- *Diff*: The difference between best and worst objective function value is checked if it is below a threshold d. A further demand is that at least $p\%$ of the individuals are feasible because otherwise *Diff* could lead to undesired results if e.g. only two individuals are feasible and they are close to each other incidentally. In contrast to the previous three criteria that are used in parameter space, *Diff* considers objective space.

Because functions have different features it may be beneficial to couple several criteria. Up to now two *combined criteria* have been regarded:

– *ComCrit*: This criterion is a combination of *ImpAv* and *MaxDist*. Only if the condition of *ImpAv* is fulfilled, *MaxDist* is checked.

– *Diff_MaxDistQuick*: *Diff* is a criterion that is rather easy to check, but it fails with flat surfaces. Therefore, if its condition has been fulfilled, the *MaxDistQuick* criterion is checked afterwards.

6 Results

As a basis for the examination a real-world problem was used that consists of optimizing a power allocation scheme for a Code Division Multiple Access (CDMA) system [20]. The overall power is minimized considering the powers of 16 individual users as parameters. Because multiple users send data simultaneously in a CDMA system, multi-user interference degrades the system performance. The application of a parallel interference cancellation technique allows estimation of the multi-user interference, so it can be subtracted from the received signal before detection, resulting in improvement of the system performance. The convergence of the parallel interference cancellation technique has to be incorporated in the optimization problem as a constraint.

In the following results are shown sorted according to the type of stopping criterion. The global optimum is considered to be reached if an objective function value of $f(x) \leq 18.5$ has been found [20]. As performance measures the convergence rate and the success performance (mean number of function evaluations weighed with the total number of runs divided by the number of successful runs) are given. A high convergence rate and a small success performance indicate good performance. To allow easy comparisons, figures showing success performances are scaled to 20,000. A maximum number of generations $G_{max} = 1000$ is used to terminate the algorithm if the examined stopping criteria do not lead to termination in appropriate time. If a run is not stopped before G_{max} is reached, the run is considered as unsuccessful.

6.1 Improvement- and Movement-Based Criteria

Because *ImpAv*, *ImpBest* and *MovPar* rely on similar mechanisms, the convergence rate and success performance of these criteria are displayed together. Considering the convergence rate, almost no dependence on the number of generations g is observable (Figure 1(a)). For decreasing values of the improvement threshold t generally the convergence rate increases, except for *MovPar* that was not able to terminate several runs before reaching G_{max} for small settings of t.

The success performance of *ImpAv*, *ImpBest* and *MovPar* is slightly increasing with growing g (see Figure 1(b)). The results regarding t are similar for *ImpAv* and *ImpBest*: For high settings of t the success performance is large because of the small convergence rate. After a strong decrease the success performance increases again for smaller values of t because of the growing average number of function evaluations for convergence.

The smallest success performance of *MovPar* is in the same range as for *ImpAv* and *ImpBest*. The difference in the average number of function evaluations for different settings of t is larger for *MovPar* than for *ImpAv* or *ImpBest*, thus the success performance grows quickly for decreasing t. As a result the success performance is better for $t = 10^{-2}$ than for $t = 10^{-4}$ although the convergence rate of $t = 10^{-2}$ is worse.

The success performance of *ImpAv* and *MovPar* has similar characteristics as for DE in [20]. For *ImpBest* the results are different: The success performance for $g = 5$ is considerably better for PSO. Furthermore, the success performance is dependent on t and almost independent from g whereas for DE it depends more on g than on t. The reason for the different results is not clear yet.

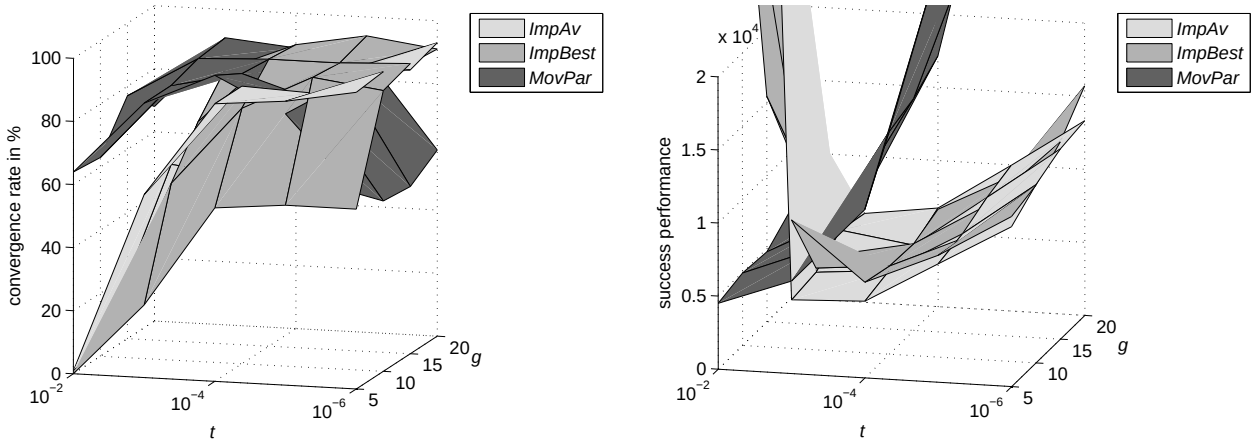
The results for *ImpAv* and *ImpBest* are considerably better here than in [19] for unconstrained single-objective problems. For *ImpAv* the reason might be that the personal best positions are regarded here instead of the current positions, but criterion *ImpBest* did not change because only the global best result is regarded. In contrast, for *MovPar* the results are worse, but it has to be kept in mind that the results are slightly dependent on the setting of G_{max} because it influences the convergence rate.

Unfortunately, suitable parameter settings for *ImpAv* and *ImpBest* cannot be derived from knowledge about the optimization problem. Besides, it is indicated in [19] that problems arise for functions with a flat surface, but it is usually not known in advance if a function possesses this property. Therefore, it will be necessary to do examinations on parameter settings for the application of these stopping criteria. Based on the examined problem parameter settings of $g \approx 10 \ldots 15$ and $t \approx 10^{-5} \ldots 10^{-4}$ are recommended. However, these settings are dependent on the optimization problem and the desired accuracy. It has to be noted also that these criteria may not be as reliable as others because improvement often occurs irregularly in evolutionary algorithms.

Criterion *NoAcc* showed good results for DE in [20] but not a single run could be terminated before reaching G_{max} for PSO. Apparently, the personal best positions improve too often to allow a stopping criterion like *NoAcc*.

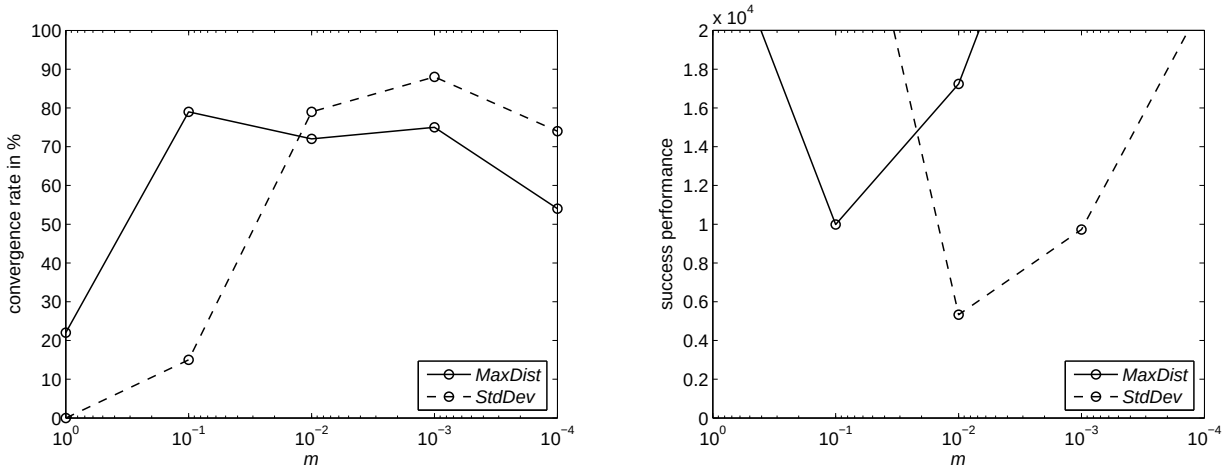
6.2 Distribution-Based Criteria

For *MaxDist* the convergence rate does not get above 80% because of runs that could not be terminated before reaching G_{max}. The results for *StdDev* are shifted in contrast to *MaxDist* and higher convergence rates are reached (Figure 2(a)). Furthermore, *StdDev* yields a lower minimum success performance than *MaxDist* (Figure 2(b)). For both criteria the performance is highly dependent on the setting



(a) Convergence rate (b) Success performance

Figure 1: Results for criteria *ImpAv*, *ImpBest* and *MovPar*



(a) Convergence rate (b) Success performance

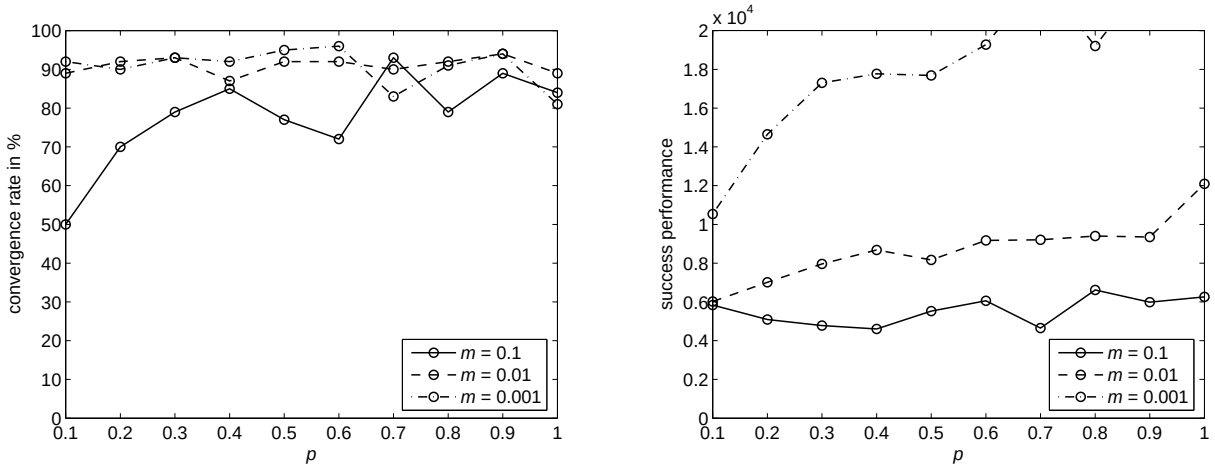
Figure 2: Results for criteria *MaxDist* and *StdDev*

of m. However, it is connected to the desired accuracy of the result. Similar effects have been found in [20] for DE. The same settings of parameter m yield the lowest success performances for *MaxDist* and *StdDev* for PSO as for DE, respectively.

The convergence rate and success performance of *MaxDistQuick* is given for $10^{-3} \leq m \leq 10^{-1}$ in Figures 3(a) and 3(b). Other parameter settings are omitted because the success performance was above 20,000. The convergence rate is fluctuating for $m = 0.1$ with different settings of p, indicating that the performance is not robust for this parameter setting. For $m = \{10^{-2}, 10^{-3}\}$ and varying p the convergence rate is approximately constant but the success performance rises with increasing p. Hence, a similar result is obtained as in [19]: Because less function evaluations are needed for convergence if smaller values of p are used and the convergence probability is not compromised, it is recommended to use e.g. $0.3 \leq p \leq 0.5$. In spite of the increased computational effort for the incorporated quicksort algorithm [18], *MaxDistQuick* is considered to be superior to *MaxDist* and *StdDev* for PSO, particularly because the increased computational effort is usually negligible when compared to computationally expensive objective function evaluations of real-world problems. For future work also a similar generalized criterion based on standard deviation instead of maximum distance should be evaluated.

For DE the success performance depends less on p [19, 20], so *MaxDistQuick* does not have advantages over *MaxDist* for DE. This behavior is supposed to be connected with the greediness of the DE selection scheme.

It may be confusing that the success performance for *MaxDistQuick* with $p = 1$ is not equal to the results of *MaxDist*. The reason is that the success performance is sensitive to even small changes in the number of success-



(a) Convergence rate

(b) Success performance

Figure 3: Results for criterion *MaxDistQuick*

ful runs. If the average number of function evaluations is examined, the results from *MaxDistQuick* with $p = 1$ and *MaxDist* are similar (not shown here).

For criterion *Diff* no definite trend can be observed regarding the demanded percentage p of feasible individuals in the population (Figures 4(a) and 4(b)) which is assumed to be due to the fact that all individuals get feasible quite fast here. Similar results were found for DE in [20]. As expected, the success performance depends on the difference threshold d. Like parameter m of the other distribution-based criteria, the setting of d is connected with the desired accuracy of the result. The highest convergence rate is achieved with $d = 10^{-2}$ but although $d = 10^{-1}$ results in a worse convergence rate, the success performance is better.

Criterion *Diff* is advantageous in contrast to the distribution-based criteria in parameter space if several parameter combinations yield the same objective function value. In this case the distribution-based criteria in parameter space may waste computational resources while the algorithm tries to converge to one point in the search space, with no or only little improvement of the objective function value. However, *Diff* is likely to produce bad results for functions with a flat surface [19].

6.3 Combined Criteria

The convergence rate and success performance for both combined criteria are given for $m \geq 10^{-2}$ because smaller values of m lead to success performances larger than 20,000 (Figures 5(a), 5(b), 6(a) and 6(b)). The results are different than for DE as the success performance increases less with decreasing value of m. Especially for *Diff_MaxDistQuick* the results are almost independent from m. However, a strong dependence on d can be seen, in particular for the success performance.

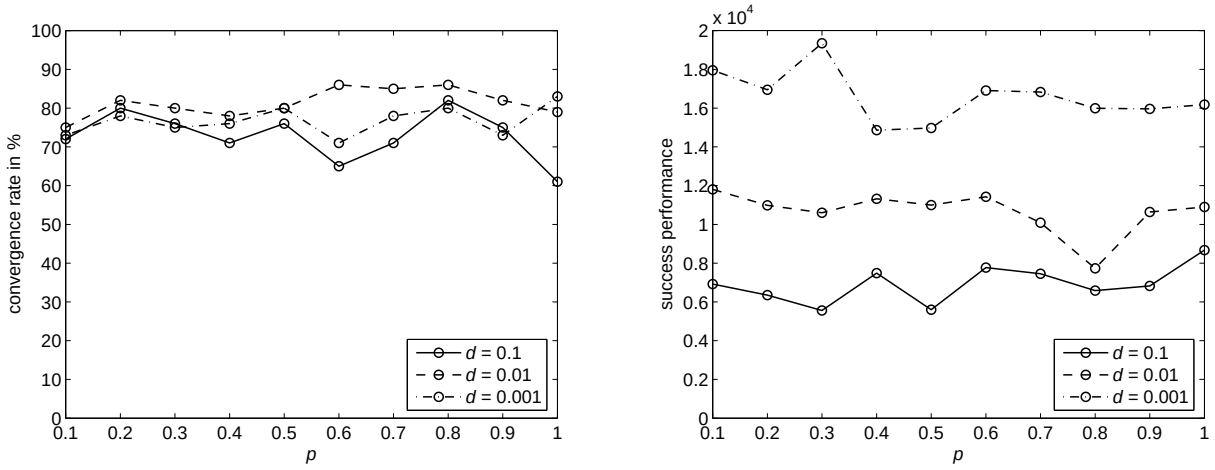
For the combined criteria generally more parameters have to be set than for the individual criteria and furthermore the dependence of parameter settings on the desired accuracy of the results cannot be seen anymore, so in general it might be easier to use the individual criteria.

6.4 Summary

Although the improvement-based criteria *ImpAv* and *ImpBest* yielded good results in this work, they are considered as rather unreliable because generally improvement occurs irregularly in evolutionary algorithms. To prevent early termination, parameter g must not be chosen too low when using these criteria. The movement-based criterion *MovPar* has similar problems. The third improvement-based criterion *NoAcc* was not able to stop a single optimization run during the given maximum number of generations, so it is classified as unsuitable for PSO although it showed a good performance for DE in [20].

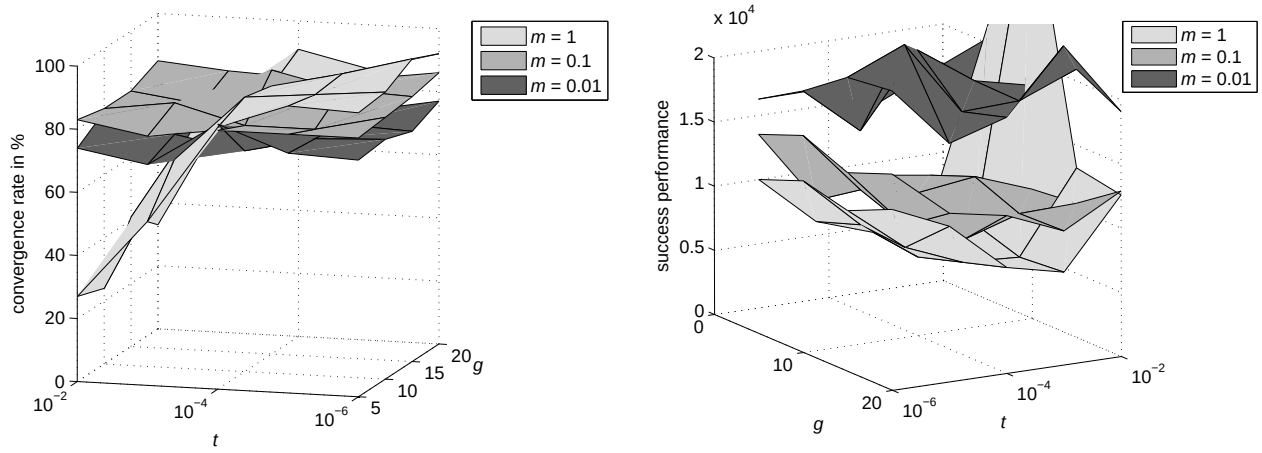
Based on the considered optimization problem as well as results from [19] it can be concluded that it is beneficial to use the generalization *MaxDistQuick* instead of *MaxDist*. Because *StdDev* performed better than *MaxDist*, a generalization of *StdDev* should be examined in future work. In general the distribution-based criteria in parameter space are classified as reliable means for detecting convergence. The distribution-based criterion in objective space (*Diff*) is also considered to be a reliable criterion with the exception of optimization problems that contain objective functions with a flat surface.

As the combined criteria are combinations of other criteria, they generally incorporate more parameters that have to be adjusted. So far no advantage of combined criteria could be found that would compensate this drawback, so it is recommended to use the individual criteria.



(a) Convergence rate (b) Success performance

Figure 4: Results for criterion *Diff*



(a) Convergence rate (b) Success performance

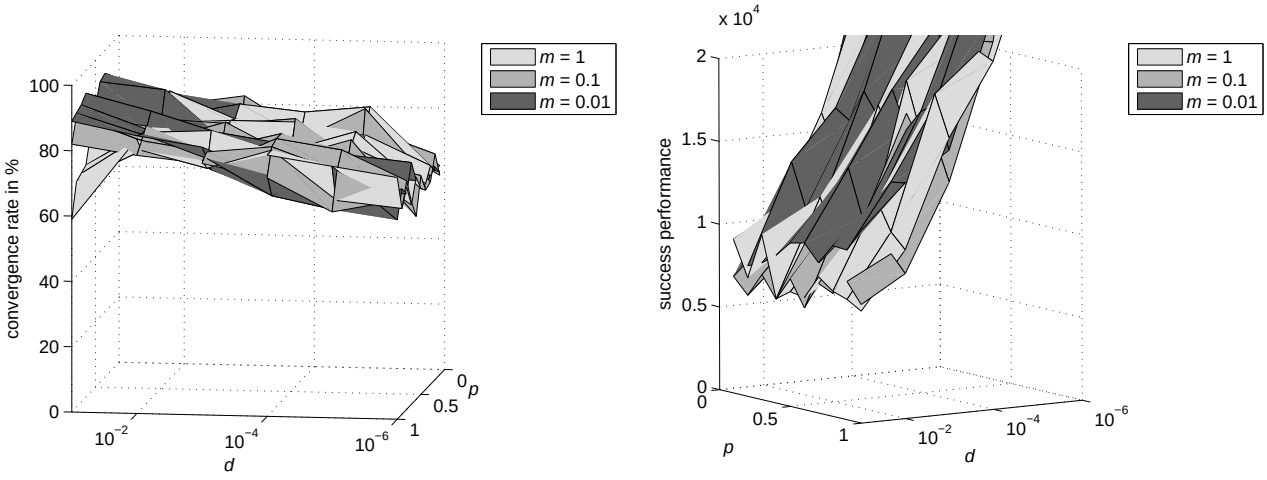
Figure 5: Results for criterion *ComCrit*

7 Conclusions

In this work stopping criteria were studied that react adaptively to the state of an optimization run based on improvement, movement or the distribution of individuals. In contrast to other examinations, not the current positions but the personal best positions were used for the calculations. It was shown that the stopping criteria can be used for constrained problems using PSO. A similar behavior as for DE could be found for several stopping criteria. It would be interesting to make comparisons with other evolutionary algorithms in future work.

Although parameter settings have to be determined in dependence on the used optimization problem, general statements could be derived. It was not possible to determine one criterion that will be best for all problems, but because of their adaptive nature generally improved performance for real-world problems is expected in contrast to termination after a limited number of function evaluations.

For multi-objective optimization the definition of appropriate stopping criteria is even more important because real-world problems usually contain multiple objectives. It will be also even more challenging because usually the population will not converge to one point in the search space but to the Pareto-optimal front, thus using error measures is difficult. One possibility would be to monitor performance measures like hypervolume [3] and calculate e.g. improvement. Another approach from literature is based on observing the development of crowding distances [12]. As only little work is done in this area so far, it is an interesting field of research for future work.



(a) Convergence rate

(b) Success performance

Figure 6: Results for criterion *Diff_MaxDistQuick*

References

[1] B. V. Babu and Rakesh Angira. New Strategies of Differential Evolution for Optimization of Extraction Process. In *Proceedings of International Symposium & 56th Annual Session of IIChE (CHEMCON 2003)*, Bhubaneswar, India, 2003.

[2] Carlos A. Coello Coello. Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002.

[3] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[4] Felipe P. Espinoza. *A Self-Adaptive Hybrid Genetic Algorithm for Optimal Groundwater Remediation Design*. PhD thesis, University of Illinois, 2003.

[5] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[6] Wilfried Jakob. *Eine neue Methode zur Erhöhung der Leistungsfähigkeit Evolutionärer Algorithmen durch die Integration lokaler Suchverfahren*. PhD thesis, Universität Karlsruhe, 2004.

[7] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, 2001.

[8] James Kennedy and Rui Mendes. Population Structure and Particle Swarm Performance. In David B. Fogel, Mohamed A. El-Sharkawi, Xin Yao, Garry Greenwood, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the Congress on Evolutionary Computation (CEC 2002)*, pages 1671–1676, Honolulu, HI, USA, 2002.

[9] Jouni Lampinen and Rainer Storn. Differential Evolution. In Godfrey C. Onwubolu and B.V. Babu, editors, *New Optimization Techniques in Engineering*, pages 123–166. Springer-Verlag, Berlin Heidelberg, 2004.

[10] Rui Mendes, James Kennedy, and José Neves. The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, 2004.

[11] Gregorio Toscano Pulido and Carlos A. Coello Coello. A Constraint-Handling Mechanism for Particle Swarm Optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, volume 2, pages 1396–1403, Portland, OR, USA, 2004.

[12] Olga Rudenko and Marc Schoenauer. A Steady Performance Stopping Criterion for Pareto-based Evolutionary Algorithms. In *Proceedings of the 6th International Multi-Objective Programming and Goal Programming Conference*, Hammamet, Tunisia, 2004.

[13] Hans-Paul Schwefel. *Evolution and Optimum Seeking*. John Wiley and Sons, 1995.

[14] Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[15] Michael Syrjakow and Helena Szczerbicka. Combination of Direct Global and Local Optimization Methods. In *Proceedings of the IEEE International Conference on Evolutionary Computing (ICEC 95)*, pages 326–333, Perth, WA, Australia, 1995.

[16] Frans van den Bergh. *An Analysis of Particle Swarm Optimizers*. PhD thesis, University of Pretoria, 2001.

[17] Karin Zielinski and Rainer Laur. Constrained Single-Objective Optimization Using Particle Swarm Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1550–1557, Vancouver, BC, Canada, 2006.

[18] Karin Zielinski, Dagmar Peters, and Rainer Laur. Run Time Analysis Regarding Stopping Criteria for Differential Evolution and Particle Swarm Optimization. In *Proceedings of the 1st International Conference on Experiments/Process/System Modelling/Simulation/Optimization*, Athens, Greece, 2005.

[19] Karin Zielinski, Dagmar Peters, and Rainer Laur. Stopping Criteria for Single-Objective Optimization. In *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, Singapore, 2005.

[20] Karin Zielinski, Petra Weitkemper, Rainer Laur, and Karl-Dirk Kammeyer. Examination of Stopping Criteria for Differential Evolution based on a Power Allocation Problem. In *Proceedings of the 10th International Conference on Optimization of Electrical and Electronic Equipment*, volume 3, pages 149–156, Braşov, Romania, 2006.

Introducing Open Source Software into Slovenian Primary and Secondary Schools

Mojca Tomazin
Business School in Brezice, Bizeljska cesta 45, 8250 Brezice, Slovenia
E-mail: mojca.tomazin2@guest.arenes.si

Miro Gradisar
University of Ljubljana, Faculty of Economics,
Kardeljeva ploscad 17, 1000 Ljubljana, Slovenia
E-mail: miro.gradisar@ef.uni-lj.si

Keywords: country-specific developments, elementary education, secondary education, software

Received: September 23, 2005

This paper deals with the use of Open Source Software (OSS) in learning environments. Advantages and obstacles of OSS are discussed. Problems and opportunities of introducing OSS into an educational process especially in primary and secondary schools are presented. The survey research, which was carried out in order to study the use of OSS in the educational system of Slovenia is described. The most important characteristics of OSS like reliability, functionality, interoperability, licensing philosophy, values of OS movements and price are examined. The results are presented and compared with those of a similar research in USA. Some interesting similarities and differences are discovered.

Povzetek: Prikazani so rezultati raziskave o uporabi odprte kode v slovenskih osnovnih in srednjih šolah.

1 Introduction

The use of Information and Communication Technology (ICT) is becoming very important in learning environments. And it is very expensive. This causes the global digital divide – the wide disparity between the world's information-rich and information-deprived, which affects educational opportunity [16], [25]. This is among others the reason why the question - which type of software to use – is becoming more and more important.

ICT consists of hardware (HW) and software (SW). Since every SW does not run on every HW, the selection of both is interconnected. But the users are in the first place interested in the functionality of ICT, which depends on SW. Therefore the decision about SW should be made first. There are several different types of SW. In learning environments the most important selection is between proprietary SW (PSW) and OSS. To understand the difference between PSW and OSS we need to explain a few terms first.

We can find a relevant open source (OS) definition on the website of the OS Initiative [20]. As the definition is quite long, we will emphasize only the most important part – distribution terms in the continuation. So the distribution terms of OSS must comply with the following criteria:

- The redistribution must be free.
- The program must include source code, and must allow distribution in source code as well as compiled form.
- The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original SW.
- Integrity of the author's source code must be ensured.
- The license should not be discriminating against any person or group of persons.
- The license must not restrict anyone from making use of the program in a specific field of endeavor.
- The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- License must not be specific to a product.
- License must not restrict other SW.
- License must be technology-neutral.

On the other hand PSW is any closed-source material, which fundamentally means that the user does not control what it does or cannot study or edit the code [27]. Its use, redistribution or modification is prohibited, or requires you to ask for permission, or is restricted so much that you effectively cannot do it freely [11]. It usually means that some individual or company holds the exclusive copyrights on a piece of SW, at the same time denying other people the access to the SW's source and the right to copy, modify and study the SW.

We must also mention the term, which is very close to the term OSS – Free SW (FSW). But we must be very careful with the term 'free software', because it has an

ambiguity problem. An unintended meaning, 'SW you can get for zero price' fits the term just as well as the intended meaning, 'SW which gives the user certain freedoms.' But for most purposes, FSW and OSS can be considered to be the same [24]. To avoid confusion we will only mention OSS in the continuation.

With the rising popularity of OSS there has been increasing interest in both its various benefits and disadvantages [21]. The crucial economic freedom that OSS provides is that of the rejection of licensing costs [5]. Another benefit is the stability and reliability – it may not have as many errors or crashes as PSW. Since anyone can see the source code, bugs can be repaired quickly. With the source code being available for all to see, greater security is also provided, because holes that would allow worms or viruses to do damage are found and fixed at an amazing rate [13].

An important advantage of OSS is a possibility of making practically unlimited modification and customization. It is true, that some users in education try to customize PSW, for example Microsoft Office [10], but these are only customizations 'on the surface' and with other limitations regarding redistribution of their 'added value'.

Proponents of OSS usually emphasize economic and technological reasons [3], [13]: lower Total Cost of Ownership (TCO), better stability and reliability, better security that usually accompanies OSS. But some of them even take the slogan of the French Revolution, 'Liberté, Egalité, Fraternité' and show how each of these ideals is an important part of the OSS movement [12]:

- Liberty
  The user is free to: use SW, understand exactly what the program does, modify SW to his/her needs, distribute the modified version etc..
- Equality
  It is important in this context that even students (in our article the term 'students' means students in primary and secondary schools) that do not have enough money to buy PSW, can use the same SW at home and at school. So effective home-school link strategies can be adopted through the exploration of the permeability of home/school boundaries [15].
- Fraternity
  Here the term fraternity stands in the context of cooperation and mutual help between SW developers and users, between users themselves, between school and families [6].

One important idea in education is also to teach computing concepts [13]. We can use OSS to achieve this goal. Many times too much effort goes into teaching procedural knowledge of specific applications. For example, OSS OpenOffice.org - offers word processing very similar to that of PSW products. They are so similar that learning most features in one will transfer to the other. Word processing as a concept is a valuable tool for students, teaching specific programs is not. By teaching from multiple angles, teachers are providing a greater breadth of information.

All the benefits of OSS have been confirmed also by the latest 'Open Source Software in Schools' study [2], published in May 2005 by the British Educational Communications and Technology Agency (Becta). Their findings namely show that OSS can provide a suitable technical infrastructure and a basic set of applications for classroom use and that it can offer a cost-effective alternative to PSW.

On the education field we can also find some specific obstacles. The first problem can be that even if the main policy framing ICT in education has the provision of HW and infrastructure as its main target – little advice on how they might be used is offered [9]. Namely, one of the most important background variables that affect ICT contrubution to powerful learning environments is teachers' skill in using ICT [23]. This can be especially problematic regarding the introduction of OSS into schools.

In Finland they have also realised, that teachers need to use a lot of time in developing new IT courses and updating the old courses. To address these problems, they have started an Open Source Courseware (OSCu) project, with a fundamental goal to increase cooperation between universities in course development [1].

The foremost concerns that schools express when contemplating migration to OSS are installation and support [7]. Any SW solution requires some service and support and for both OSS and PSW experts depend on email lists and community Web sites as well as contracted support. Some OS companies offer good support contracts, but a school may not be able to afford them. So they mostly depend on OS community volunteers and help may not be as certain or as timely as they wish [17]. A commonly cited problem is also finding time to absorb the new technology to maximize its pedagogic potential [26].

In recent years there have been many political initiatives trying to foster OS movement and to spread the use of OSS in public administration and at schools and universities [22]. In Slovenia in the past, some teachers wanted to introduce OSS into their educational processes on their own. There was no support from educational institutions. Therefore in 2003 the Slovenian Ministry of Education came to realize that educational institutions should be offered - besides PSW (mainly Microsoft SW) - also alternative SW. It has started the Open Source Project - 'OS project' [19] with the intention to make the introduction of OSS into education environments faster and more efficient. The Ministry has invited teachers to get involved into the project. In fact an 'expert group' of enthusiastic teachers was formed. But they soon realized that introducing a new type of SW into educational field is not an easy job.

The main problem in using OSS is the lack of experience. Opportunities for appropriate training are limited, with many educators being self-taught IT users [8]. Sometimes, teachers have ICT-competence training with a measure of success, but it is tempered by a considerable degree of negative reaction to form and content of the training [14]. That is why we have made it our primary assignment to direct teachers and other

educational workers into recognizing and using OSS and didactic applications and materials, based on them.

The OS project is intended to integrate informational environments based on open standards and OSS into educational establishments. This will increase the selection of didactic tools and applications, which can be used by teachers and/or students in the process of teaching and learning. OS project directly supports the use and further development of OSS in educational processes.

The more detailed goals of the OS project are:

- To acquire quality and free (or moderately priced) didactic SW for the purpose of education in educational establishments. All subjects of teaching will be covered by a number of didactically suitable applications and tools. Students and teachers can use the SW freely for educational purposes at school and at home.
- To offer to educational establishments OSS based on open standards, which does not depend on one manufacturer solely.
- To inspire the use of OSS and its further development.
- To stimulate the use and development of didactic applications, which are portable and operating system-independent.
- To educate teachers, so they will be willing to use didactic programs and tools irrelevant of the system environments.
- To teach students how to use applications and programming tools irrelevant of their system environments.

Besides the Slovenian Ministry of Education, this project has also been founded and supported by the educational establishments, The National Education Institute of The Republic of Slovenia, Universities of Ljubljana, Maribor and Koper, Government Center for Informatics, Ministry of Information Society, LUGOS – Linux User Group of Slovenia and individual expert associates.

One way to achieve all these goals is also introducing some kind of 'Reference schools'. A reference school acts as a model and a guide to others and is a center for exchanging knowledge in using OSS in education establishments for their region. Reference schools are intended for educating IT teachers, school ICT administrators, subject teachers and students. Their goal is to encourage everyone to use didactic OSS and material for the purpose of education. Teachers and students will learn how to use it for teaching and learning and how to develop and upgrade didactic SW based on OS.

Subject teachers present the contents, i.e. the demand, for a specific didactic application, whereas IT teachers and administrators, together with students, develop and test this application or work with outside developers in developing and testing it. A reference school will present a motivational environment for everyone involved.

We have also started a research about using OSS in education, because in Slovenia no research was done in the past that would answer the questions like: who, why, how…uses OSS in education. Actually, we knew that some 'enthusiastic' teachers use OSS in the classrooms (and otherwise) on their own, but nobody systematically followed their attempts and analyzed the situation. We also wanted to get information about their experience with introducing OSS, their general opinion of OSS and about solutions they recommend.

We must also emphasize that we do not want to prejudice in advance that OSS is a better solution to education than PSW. Moreover, the coexistence of PSW and OSS is a possible interpretative key for the success of OS movement [4] and we must consider all types of SW equally.

In the remainder of the paper survey methodology is first described then the results of the survey are presented and compared with the results of a similar research in USA. At the end the main findings are summarized and a possible further research is suggested.

2 Survey methodology

Our research is based on a similar research by Northwest Regional Educational Laboratory (NREL), which has been running in USA since 2002. The purpose of this survey was to study the use of OSS in K12 education system [18]. The questionnaire was posted on the Northwest Educational Technology Consortium (NETC) website in November 2002 and data were collected in February 2003. The survey was open to all. Survey analysis only included the fully completed questionnaires and the answers of those survey participants, who were currently working in a K12 school. The participants were able to choose whether they wished to remain anonymous, but many had decided to give their personal information anyway.

The questionnaire was intended for anyone using OSS in K12 schools. To draw the participants from the target audience, they posted the survey link on mailing lists and different websites. Most participants lived and worked in USA (nearly half of them from Oregon and Washington).

Our (Slovenian) survey was entitled 'Using FSW/OSS in the Process of Education and for Administrative Purposes'. We have decided to use the same methodology as in NETC survey, since ours was also open to all users (no password/username required) and the personal information was also optional. The survey included a similar range of target population in primary and secondary schools in Slovenia.

The research tries to answer the following questions about educators:

- Who they are?
- What they use?
- Do they use OSS on their desktops?
- How difficult was it to implement OSS in schools?
- Which solutions do they recommend?
- Their general opinion of OSS.

The questionnaire was available on the website of the OS project. We also posted the survey link on mailing lists of all 'target' schools in Slovenia. As Slovenia nearly has

no other type of school than state/public school (on the primary and secondary level), these schools can be considered as public (non-private) schools. The e-mail invitation to participate in our research was sent to the principals of:

- 151 secondary schools
- 463 primary schools
- 41 primary schools for children with special needs
- 81 music schools

The principals were asked to inform their employees about the research and to ask them to participate. The invitation was sent on 22 January 2004. We ended collecting the data (from the web questionnaire) on 9 March 2004.

The answers in the questionnaire were mainly predefined and were open only if it was not possible to predict the answers of the participants in advance. But basically we have taken the original research and modified it slightly to learn as much as possible about the use of OSS in primary and secondary schools in Slovenia.

The research yielded 433 entries with 280 valid entries (67%). An entry was not considered to be valid, if the questionnaire was not completely filled-in or if the answers were obviously wrong. Some participants only looked at the questions or just filled in a few of them.

3 Results

The results were categorized into 6 sections:

- Research participants
- Use of OSS on desktops
- What influences are important in selecting SW
- The difficulty of transition
- General opinion of OSS
- Use of Information Technology (IT) in education – this section was not a part of the original NETC survey and it is not directly connected with OS. Nevertheless we added this section in order to get information about the general use of IT in education.

3.1 Research participants

There are four questions in this part of the questionnaire. We wanted to find out what position the participants have and if they are teachers, where (primary/secondary level) they work, if they are IT teachers and if they select SW for students. We also wanted to know, if they work mainly in small, medium or large schools.

Table 1 shows the structure of the participants according to their positions in educational establishments.

Teacher	21%
Principal or other executive	21%
Administrator (works in school administration)	13%
Full-time IT maintenance person	45%
Part-time IT maintenance person	0%

Table 1: The structure of the positions of the participants in educational establishments.

As we can see, nearly half of the participants are full-time IT maintainers. About one fifth of them are teachers and another fifth of them principals.

If the option ‘teacher’ (Table 1) was selected, the participants had to select one of four further options regarding the subject they teach (Table 2).

IT subjects at primary school	28%
IT subjects at secondary school	12%
Other, non-IT subjects at primary school	52%
Other, non-IT subjects at secondary school	8%

Table 2: The structure of teachers-participants.

When we look at the structure of teachers-participants we notice that more than a half of them teach non-computer subjects at primary level, while every third teaches computer subjects at primary level. Only every one in five teacher participants were secondary-school teachers (of both, computer and non-computer subjects).

The question, do you select the SW to be installed on other computer-desktops (i.e. students’ computers) has been answered as follows: 58% of the participants are responsible for selecting SW to be used on students’ desktops, while 42% have no influence in this aspect.

In Table 3 the answers to the question, how many students go to your school, are categorized.

Less than 100	6%
100-499	58%
500-999	30%
1000-1999	5%
More than 2000	1%

Table 3: Number of students at the school of the participants.

3.2 Use of OSS on desktops

In this part we tried to answer the question, what kind (if any) of OSS the participants/their students use on their desktops. It consists of eight questions. We focused on the operating system and most popular applications – internet browser and office suite, but we also asked about other OSS–use. The questions refer separately to the use of participants themselves and to the use of their students.

	Yes	No	I don't know
Do you use OS operating system on your computer?	12%	81%	7%
Do you use OS operating system on your student's computer?	5%	88%	7%
Do you use OS browser on your computer?	24%	70%	6%
Do you use OS browser on your student's computers?	12%	82%	6%
Do you use OS office suite on your computer?	28%	66%	6%
Do you use OS office suite on your student's computers?	14%	80%	6%

Table 4: Types of OSS that the participants/their students use on their desktops.

In general OSS is very poorly represented (Table 4). The participants do use OS office suite on their own computers, yet only 28% of them. They also use OS browsers and operating systems but in very low percentages and mostly on their home computers rather than on their students' computers.

Other OSS that you use on your computer or on your students' computers? Only 8% of all participants entered something in this area. It turned out that the participants still have problems with the term OS, because many of them entered different kinds of Microsoft SW, which does not belong in this category. Those answers will therefore not be analyzed further. We can determine, however, that it will take much time and effort to educate teachers and other education-related professionals of different kinds of SW equipment.

3.3 Factors influencing the selection of SW

In this section the participants had to choose the importance of predefined factors influencing their decisions about using OSS. We can see the results for three types of SW: OS operating system (Table 5), office suite (Table 6) and browser (Table 7).

	Very important	Important	Not so important	Unimportant	I don't know
Customization	56%	16%	8%	4%	16%
Desirable features	61%	26%	4%	0%	9%
Interoperability	65%	22%	4%	0%	9%
Price	82%	9%	0%	0%	9%
Reliability	82%	9%	0%	0%	9%
Reputation	9%	17%	57%	13%	4%
Teachers/ students can use SW at home	40%	30%	17%	9%	4%

Table 5: Factors influencing the selection of OS operating system.

The most important criteria in deciding, which operating system to use, turned out to be price and reliability, followed by interoperability and desired functionality. Its openness (students/teachers can use SW at home) and reputation do not seem to be so important.

	Very important	Important	Not so important	Unimportant	I don't know
Customization	62%	21%	13%	0%	4%
Desirable features	58%	33%	0	0	9%
Interoperability	79%	13%	0%	0%	8%
Price	79%	13%	0%	0%	8%
Reliability	67%	25%	0%	0%	8%
Reputation	8%	21%	46%	21%	4%
Teachers/ students can use SW at home	55%	21%	8%	8%	8%

Table 6: Factors influencing the selection of OS Office suite.

In the aspect of office suite, the importance of price and that of interoperability were balanced. As seen before, reputation is not very important.

	Very impor-tant	Impor-tant	Not so impor-tant	Unim-portant	I don't know
Customi-zation	42%	29%	21%	4%	4%
Desirable features	46%	46%	4%	0%	4%
Interope-rability	54%	21%	13%	8%	4%
Price	58%	13%	21%	4%	4%
Reliability	62%	17%	17%	0%	4%
Reputa-tion	9%	13%	52%	26%	0%
Teachers/ students can use SW at home	37%	21%	17%	21%	4%

Table 7: Factors influencing the selection of OS Internet browser.

Here, the situation is very similar to that in operating system and office suite categories. The results show that the participants do not find price so important as we have seen before. One can expect that the reason for this is the fact that Microsoft Internet Explorer is also free of charge. Reputation, again, does not seem to play a major role in the decision.

3.4 The difficulty of transition

It was interesting for us to see the results of this section, because OSS is usually considered to be harder to implement than similar solutions. Table 8 shows how hard it was (for the participants) to implement the OSS solution (technically) when compared to similar solutions (PSW).

Harder	Similar	Easier	I don't know
9%	47%	3%	41%

Table 8: The difficulty of implementation of OSS solution, compared to similar solutions.

The participants mostly believe that OSS is technically just as simple (or difficult) to implement as similar commercial solutions.

And how satisfied are the participants with the OSS solution when compared to similar ones (PSW)? The results are shown in Table 9.

Very satisfied	Similar	Not satisfied	I don't know
9%	47%	3%	41%

Table 9: How satisfied are the participants with OSS solution, compared to similar solutions?

Similar to the previous question, most participants are just as (dis) satisfied with this solution as they are with other commercial ones.

We could also expect that the participants got some reluctance from superiors and/or users while implementing OSS solution. Table 10 shows their opinion about the level of this reluctance.

Much	Some	Very little	I don't know
18%	18%	26%	38%

Table 10: Reluctance while implementing OSS solution.

Almost one in five participants encountered much reluctance (mostly from superiors and/or end users) when trying to implement a new solution and just as many of them encountered some reluctance,.

Would you like to recommend any specific OSS? (open-type question)

The participants mostly (40%) recommend OS office suite OpenOffice.org, which is followed by Linux operating system (20%) and other applications.

3.5 Factors influencing the selection of SW

What is your general opinion about OSS? Here, the participants had to select the level of agreement or disagreement with some predefined statements:

- Some OSS is ready to be used in education.
- I personally wish to use OSS wherever possible.
- I am not interested in the competition between different licensing philosophies and such, I only want to satisfy my needs.
- OS values and philosophy influence my decision in the selection of SW.

The following levels of agreement/disagreement were available: I strongly agree, I agree, Neutral/I don't know, I disagree, I strongly disagree. Nearly a half (48%) of the participants agree or strongly agree with the statement that some of OSS is mature enough to be used in education. Over a half (52%) of the participants are eager or very eager to use OSS wherever possible. The answers to the third statement show that the participants are not interested in the competition between different values and license philosophies and are only interested in what will help them achieve their goal (79%). However, one in every three participants is influenced by the values and the philosophy of OS movement when selecting SW, while others are neutral.

3.6 Factors influencing the selection of SW

As previously mentioned, we added this part of the questionnaire to find out a general use of IT. Table 11 shows, how and where IT is used in primary and secondary schools.

	Often	Rarely	Never
For preparing lessons and materials	70%	16%	14%
For lessons/work in IT classroom	55%	19%	26%
For presentations	48%	29%	23%
At laboratory work	16%	28%	56%
For administrative tasks	76%	15%	9%
For e-mail communication and similar tasks	81%	10%	9%
For browsing on the Internet	81%	10%	9%

Table 11: The use of IT in primary and secondary schools.

IT is mostly used for e-mail communication and for browsing the Internet. This is followed by administrative tasks and after that comes the use of IT for preparing lessons and materials. Computers have generally proved to be hardly ever used in laboratory work. Besides these predefined options, the participants also entered other activities where IT is to be used: in libraries, for writing music, for home schooling etc.

The last but one of the most important questions was, how important are different obstacles in using IT in education (Table 12)?

	Very imp.	Partly imp.	Not imp.
Poor education teachers get in using IT	61%	37%	2%
Nonexistent equipment in schools	48%	47%	5%
Inefficiency of using IT in schools	27%	61%	12%
Lack of materials, documentation and other support	39%	55%	6%
Unsuitable system of educating teachers for using ICT	42%	53%	5%

Table 12**:** The importance of different obstacles in using IT in education.

The most important obstacle in using IT in education has proved to be the poor education teachers get in using IT. Besides that, the participants listed old or nonexistent equipment in schools. The participants also had a chance to enter their own thoughts on the subject, but no such answers were given.

What is your suggestion for improving the use of IT in education? As this was an open-type question, it resulted in many different answers. These answers could be grouped into different categories, the most obvious one being educating teachers in using IT, which (again) seems to be the basic problem for teachers.

4 Comparison with a similar research on NREL

In this section we compare the results of our survey and those of a similar NREL survey by individual questions.

Who are the participants? NREL: Most participants (44%) are working for school districts (not individual schools) and are not teachers. Most of them are responsible for administration, selecting also the SW to be used on other people's computers. Minority of participants were teachers (13%). Our survey: Nearly half of the survey participants are full-time IT administration/maintainers, some 20% are teachers (one fifth working in high school, others primary school) and about the same amount of them are working in management. A fair half (58%) is in a position to select SW to be used on other computers (i.e. for students). Due to the differences between systems of education in USA and Slovenia, this part of survey was hard to compare. Despite the differences, however, we can notice certain similarities: in both surveys, most participants came from the area of IT administration and only a small percentage were teachers. Similarly, a large portion of participants in both surveys select SW on other people’s computers.

What are they using and are they using OSS? Except in the use of OSS office suites, which is almost identical (in percentages) in both surveys, the use of OSS is clearly more widespread in the NREL survey (on participants' computers as well as on student’s computers). For example, in NREL survey 42% of respondents use OSS operating system (e.g. Linux) on their computers, which is 30% more than in our survey.

When we are talking about OSS operating systems, the most important quality in NREL survey is reliability, followed by desired features and price. After that comes interoperability. These four criteria were also the most important ones in our survey.

Similarly as in the case of operating systems, in NREL and in our survey the office suite and web browser fulfills the following most important criteria: price, reliability, interoperability and desired features.

An interesting situation evolves around the reputation criterion, which proved to be of higher priority to the participants of NREL survey than to those included in our survey. On the other hand, the participants of our survey believe that being able to legally use the SW at home is very important, while to the American participants this is not so.

In the NREL survey, the participants believe that implementing most OSS solutions is easier or at least just as demanding as similar commercial solutions. On their homepage there is additional material – comments by the participants, interviews etc. The answers and comments they have entered show that the transition and implementation can be somewhat difficult in the beginning, however the solutions later prove to be more reliable and satisfactory in the long run. Most participants did not experience a considerable reluctance by their administration, as is frequently the case in other technological reorganization. Some mention slight

reluctance, which is cleared when they get to know the new solution.

The obvious difference between the answers to the question, how demanding it was to implement OS solution, when compared to similar PSW solutions, is due to a much larger number of undecided answers (don't know) in our survey. This option was far more frequent than in the NREL survey, which is the reason, why we cannot directly compare these results. We can, however, determine that option 'harder' was in both surveys surpassed by the number of participants selecting 'easier' and 'about the same'.

The answers to the question, how satisfied you are with the OS solution compared to similar PSW solutions, shows that in both surveys there is only a trivial number of users who are not satisfied with the solution.

We can notice a higher percentage of participants in Slovenia complaining about the reluctance from their management and users than in USA.

In the NREL survey regarding the open-type question, which OSS you would recommend to other users, most participants recommend Red Hat Linux distribution, including the American K12 version (K12LTSP), which is based on it. Most popular backend solutions prove to be Apache web server and SquidGuard, while OpenOffice.org office suite, The Gimp and web browser Mozilla are most recommended in the frontend section. Similarly to NREL survey, our survey also mentions Red Hat Linux and OpenOffice.org as the most recommended OSS solutions.

By expressing general opinion of OS (Some OSS is ready to be used in education) only a minimal number of participants selected the negative options (disagree and strongly disagree) – 1-2 %. The Slovenian participants were more inclined towards neutral options than the American, whereas there was an almost three-times higher number of positive ('strongly agree') answers in America than in Slovenia.

Answers show that the participants of the NREL survey were more inclined to use OSS anywhere possible than the Slovenian participants.

The Slovenian participants are far less interested in the competition between different licensing policies and more in finding and using the best solution to suit their needs.

34% of the Slovenians and 54% of the Americans agree or strongly agree with the statement: The values and the philosophies of OS movement influence my decision on using a certain SW. On the other hand, 8% of the Slovenian and 13% of the American participants disagree or strongly disagree with it. From these results we can conclude that NREL survey participants are more easily influenced by the philosophy and values of OS movements.

5 Summary and some concluding remarks

Finally, we will summarize the most important findings of our research:

- In general, OSS in education is very poorly represented.
- The participants still have problems with the 'OS' term, so it will take much time and effort to educate teachers and other education-related professionals of different kinds of SW.
- The fact that students and teachers can use SW at home and reputation of SW are not as important to the participants as one might expect.
- The participants mostly believe that OSS is technically just as simple to implement as similar commercial solutions.
- The participants mostly recommend OS office suite OpenOffice.org and Linux operating system Red Hat.
- About a half of the participants agree or strongly agree with the statement that some of OSS is mature enough to be used in education and are eager or very eager to use OSS wherever possible.
- The most important obstacle in using IT in education has proved to be poor education teachers get in using IT. Besides that, the participants listed old or nonexistent equipment in schools.
- The most important suggestion for improving the use of IT in education that the participants gave was to educate teachers in using IT, which seems to be the basic problem for teachers.

The most interesting findings in comparison between the results obtained in Slovenia and USA are:

- Except in the use of OSS office suites, which is almost identical in both surveys, the use of OSS on participant's computers as well as on student's computers is clearly more widespread by the NREL survey.
- The most important qualities of OSS in both surveys are reliability, desired functionalities, price and interoperability.
- The reputation criterion proved to be of higher priority to the participants of NREL survey than those of our survey, but the ability to legally use SW at home is more important for the Slovenian participants.
- We can notice a higher percentage of participants in Slovenia complaining about the reluctance of their management and users than in USA.
- The Slovenian participants are far less interested in the competition between different licensing policies and more in finding the best solution to suit their needs. NREL survey participants are more influenced by the philosophy and values of OS movements.

Finally, we would like to mention that the OS philosophy is not only about SW. An interesting movement that is also going on in the world is an initiative to make open to the public all kinds of learning resources as e-resources. Wikipedia might be the most famous example of this movement. It is a Web-based free content encyclopedia that is openly edited and freely readable. A small step towards these temptations is also a decision of the Slovenian Ministry of Education to

analyze possibilities to publish some student-books and workbooks in an e-form on Internet, so that students and parents could more or less freely use them. So these may be the most important directions for future research of this field.

Acknowledgement

This research was supported by Ministry of Higher Education, Science and Technology of the Republic of Slovenia under the grant No. P2-0037.

References

[1] Ala-Mutka, K., & Mikkonen, T. (2003). Experiences with Distributed Open Source Courses. *Informatica*, 27(3), pp. 243-254.

[2] Becta-British Educational Communications and Technology Agency (2005): 'Open Source Software in Schools'. Available at http://-www.becta.org.uk/corporate/press_out.cfm?id=4681, last accessed 22 September 2005

[3] Bensberg, F., & Dewanto, B. L. (2003). TCO VOFI for eLearning Platforms. Available at http://www.campussource.de/org/opensource/docs/bensbergVor.doc.pdf, last accessed 9 March 2005.

[4] Bonaccorsi, A., & Rossi, C. (2003). Why Open Source software can succeed. Available at http://opensource.mit.edu/papers/rp-bonaccorsirossi.pdf, last accessed 9 March 2005.

[5] Browne, C. (2001). The Economics of Free Software. Available at http://cbbrowne.com/info/-freeecon.html, last accessed 9 March 2005.

[6] Brunelle, M., & Bruce, B. (2002). Why free software matters for literacy educators. *Journal of Adolescent & Adult Literacy*, 45(6), pp. 514-519.

[7] Butcher, M. (2002). Linux in education: Open Source provides a better solution to schools. *Newsforge – The Online Newspaper of Record for Linux and Open Source*. Available at http://newsforge.com/article.pl?sid=02/05/17/1319208&mode=thread&tid=46, last accessed 9 March 2005.

[8] Carmichael, P., & Honour, L. (2002). Open Source as Appropriate Technology for Global Education. *International Journal of Educational Development*, 22(1), 47-53. Also available at http://www.col.org/tel99/acrobat/carmichael.pdf, last accessed 9 March 2005.

[9] Dale, R., Robertson, S., & Shortis, T. (2004). 'You can't not go with the technological flow, can you?' Constructing 'ICT' and 'teaching and learning'. *Journal of Computer Assisted Learning*, 20(6), pp. 456-470.

[10] Deacon, A., Jaftha, J., & Horowitz, D. (2004). Customising Microsoft Office to develop a tutorial learning environment. *British Journal of Educational Technology*, 35(2), pp. 223-234.

[11] GNU Project (2004). Available at http://www.gnu.org/philosophy/categories.html, last accessed 8 March 2005.

[12] Har'El, N. (2001). Free Software 'Liberté, Egalité, Fraternité'. Available at http://nadav.harel.org.il/-essays/lfe.html, last accessed 4 January 2005.

[13] Hart, T. (2004). Open Source in Education. Available at http://moodle.ntjcpa.edu.tw/file.php-/1/osined-1.0.pdf, last accessed 4 January 2005.

[14] Henning, E., & Van der Westhuizen, D. (2004). Crossing the digital divide safely and trustingly: how ecologies of learning scaffold the journey. *Computers & Education*, 42(4), pp. 333-352.

[15] Kent, N., & Facer, K. (2004). Different worlds? A comparison of young people's home and school ICT use. *Journal of Computer Assisted Learning*, 20(6), pp. 440-455.

[16] Kirkwood, A. (2001). Shanty Towns around the Global Village? Reducing Distance, but Widening Gaps with ICT. *Education, Communication & Information*, 1(2), pp. 213-228.

[17] NETC (2002a) - Northwest Educational Technology Consortium: Deploying and maintaining OSS. Available at http://-www.netc.org/openoptions/pros_cons/deployment.html#expertise , last accessed 8 March 2005.

[18] NETC (2002b) - Northwest Educational Technology Consortium: Making Decision About Open Source Software (OSS) for K-12. Available at http://www.netc.org/openoptions/ , last accessed 8 March 2005.

[19] OS (2003) Project: Introducing opensource and free software into educational processes. Available at http://oko.edus.si/index.php?module=ContentExpress&func=display&ceid=153 , last accessed 8 March 2005.

[20] OSI (2005) Open Source Initiative. Available at http://www.opensource.org/docs/definition_plain.php , last accessed 8 March 2005.

[21] Payne, C. (2002). On the security of open source software. *Information Systems Journal*, 12(1), pp. 61-78.

[22] Schmidt, K. M., & Schnitzer, M. (2002). Public Subsidies for Open Source? Some Economic Policy Issues of the Software Market. Available at http://www.vwl.uni-muenchen.de/ls_schmidt/research/disc/pdf/opensource.pdf , last accessed 9 March 2005.

[23] Smeets, E. (2005). Does ICT contribute to powerful learning environments in primary education?. *Computers & Education*, 44(3), pp. 343-355.

[24] Tan Wooi Tong (2004). Free/Open Source Software: Education. Asia-Pacific Development Information Programme – e-Primers on Free/Open Source Software. Available at http://eprimers.apdip.net/series/foss/edu-toc , last accessed 8 March 2005.

[25] Tiene, D. (2002). Addressing the Global Digital Divide and its Impact on Educational Opportunity. *Educational Media International*, 39(3-4), pp. 212-222.

[26] Waite, S. (2004). Tools for the job: a report of two surveys of information and communications technology training and use for literacy in primary

schools in the West of England. *Journal of Computer Assisted Learning*, 20(1), pp. 11-20.

[27] Wikipedia (2005): Proprietary Software. Available at http://en.wikipedia.org/wiki/Proprietary_-software, last accessed 8 March 2005.

Usable Collaborative Email Requirements Using Activity Theory

Lorna Uden and Aravind Kumaresan
Staffordshire University Faculty of Computing, Engineering and Technology
The Octagon Beaconside, Stafford, ST18 OAD. UK
E-mail: l.uden@staffs.ac.uk

Kimmo Salmenjoki
University of Vaasa, Department of Computer Science, Box 700, 65101 Vaasa, Finland
E-mail: ksa@uwasa.fi

Keywords: requirement analysis, email, interface design, context, activity theory

Received: November 24, 2006

Email is the most common collaborative tool in use today. Although originally designed as an asynchronous communication tool, it is being used increasingly for information management, coordination and collaboration tasks. For effective collaborative work, email must be designed that meets users' needs and their experience. The traditional approach to designing interfaces has been increasingly criticised because of the gaps between research results and practical design, especially concerning requirements. Requirements elicitation is a key to the success of the development of all email applications. Activity theory incorporates the notions of intentionality, history, mediation, motivation, understanding, culture and community into design. In particular, it provides a framework in which the critical issue of context can be taken into account. This paper describes the use of activity theory for the requirements analysis of a collaborative email system for a manufacturing company, XBC Ltd.

Povzetek: Predstavljena je uporaba elektronske pošte za skupne aktivnosti.

1 Introduction

Among collaborative tools such as List servers, Newsgroups, Web Conferencing, Internet Relay Chat (IRC), Internet Phone, Internet Radio, and Desktop Video Conferencing, email is the most commonly used tool for electronic collaboration because of its asynchronous information sharing capability. Another reason is, virtually everyone who has ever touched a computer understands email. Email has become an important tool for communication in our modern life. Although email was originally designed as a tool for asynchronous communication, it has become more like a habitat than an application (Ducheneuaut & Bellotti 2001). Email today is being used for tasks such as information management, coordination and collaboration in organisations. It is also increasingly being used as a portal for access to online publications and information services, thus making email a personal information management tool for many purposes. Research from Ducheneuaut and Bellotti (2001) found that email is used for many information management features such as 'to-dos' (by marking up or resending oneself messages) and contact management (by sorting-by-name and filtering). According to Khoussainov & Kushmerick (2005), emails are still designed mainly to manipulate individual messages. Their features in automated email management are confined to filtering individual messages and simple message threading.

Email has been increasingly used as a time management tool (Gwizdka 2001; Whittaker & Sidner 1996). In-boxes are often used to keep messages referring to future events that cannot be dealt with on arrival. Messages are also used as reminders about non email tasks and events. Microsoft Outlook provides a number of features supporting various aspects of managing pending tasks (e.g. a to-do list, a calendar, general email flags, specialised remainder flags along with a type of action required). It also has temporal information organisation (e.g. Journal). Despite the provision of these features, very few users actually use them. The reason is being a lack of integration of email along with other software applications or media (Gwizdka 2004) leading to usability problems for users.

Usability is concerned with how easy it is to use and learn to use the system as well as how efficient and effective is the application. Users would only use the system if it is easy to use and allows them to carry out their tasks effectively and efficiently. The flow of tasks by the users in collaboration should be easily managed, shared and monitored. Another role of email is task management. However, current email systems are not effective in managing tasks (Whittaker & Sidner 1996).

Central to the design of usable email applications that meet users' needs is requirements analysis. Effective and efficient requirements elicitation is absolutely essential if software systems are to meet the expectations of their customers and users, and are to be delivered on time and within budget (Al-Rawas & Easterbrook 1996).

Goguen and Linde (1993) have provided a comprehensive survey of techniques for requirements elicitation, focusing on how these techniques can deal with the social aspects of this activity. They raise the important concept of social order in requirements elicitation and conclude that the requirements elicitation problem is fundamentally social and, thus, unsolvable if we use methods that are based entirely around individual cognition and ignore organisational requirements.

Organisational requirements are those requirements that are captured when a system is being viewed in a social context rather than from a purely technical, administrative or procedural view of the functions to be performed. Sources of such requirements could be power structures, roles, obligations, responsibilities, control and autonomy issues, values and ethics (Avison & Wood-Harper 1990). These types of requirements are so much embedded in organisational structure and policies that often they cannot easily be directly observed or articulated.

Most established techniques, however, do not adequately address the critical organisational and 'softer', people-related issues of software systems. From the above discussion, it would seem that current models could not provide a theoretical basis for understanding 'regularly patterned' human activity (Probert 1999). In order to overcome the above mentioned problems, it is necessary to have a methodology and tools that can support the continuous evaluation of a statement of requirements as these evolve against a highly complex and dynamic problem situation. What is needed is to shift the focus from fixed and final requirements to those of a more dynamic nature. In particular, it is necessary to consider human information which, in social terms, does not have a physical reality and is not objective like physical information. Instead, it is based on individual, group or organisational needs. Such information informs action in organisations and is thus closely related to organisational activity and organisational form.

Organisational activity is itself a function of the social purposes of individuals, groups and organisations and is affected by issues outside the boundary of the organisation. Human information is subjected to change and is ongoing. One reconceptualisation of human information that allows for social organisation processes is Activity Theory (McGrath & Uden 2000). Activity Theory has increasingly been suggested by researchers in recent years as an ideal candidate for system design, and Human Computer Interaction (HCI) design (Kuutti 1991; Nardi 1996). We believe that activity theory can be used as a framework for understanding the totality of human work and praxis and the deliberate processes changing this, i.e. a totality encompassing organisational development, design and use of computer artefacts (Bodker 1991).

Besides the requirements problems facing designers with the design of collaborative email, there is also the issue of interface of the application. Researchers in recent years have criticised the gap between research results and practical design in HCI. Bannon (1991) lists several limitations of the traditional cognitive psychology approach. Firstly, in the traditional approach, the human actors are simply passive elements in a system, not an autonomous agent that has the capacity to regulate and co-ordinate his or her behaviour. Secondly, the problem of using predetermined fixed requirements for product design. Instead of considering only a single individual, features of co-operation, communication, and coordination are often vital in the successful performance of tasks. Thirdly, restricted and artificial laboratory experiments have been the trend rather than work practices. Finally, there is a growing recognition that the actual use of a system is a long-term process that cannot be adequately understood by studying just the initial steps of usage. There is an emerging consensus among researchers that the cognitive approach to HCI may be limited. It does not provide an appropriate conceptual basis for studies of computer use in its social, organisational and authorial context, in relation to the goals, plans and values of the user or in the context of development.

Activity theory offers the possibility of seeing use and system design as a multitude of change cycles, where computer applications as well as other parts of the work activity are constantly reconstructed using more or less well-known materials, design tools and techniques, with a more or less clear understanding of the product. An explicit awareness of these cycles may change our way of doing design (Floyd 1987). Also in activity theory, conflicts can be acknowledged and taken seriously in design. This paper argues that activity theory provides an appropriate framework for elucidating requirements. It focuses on the interaction of human activity and consciousness within its relevant environmental context. In this paper, we present a case study involving the use of activity theory in requirements elicitation for the design of a collaborative email application for the XBC organisation. This paper begins with a brief review of activity theory and its implications for collaborative email design. This is followed by the description of the requirements analysis of the XBC collaborative email application using activity theory. The paper concludes with further research suggestions.

2 Activity theory

Activity theory has evolved through three generations of research (Engeström 2001). The first generation of activity theory drew heavily on the work of Vygotsky's conception of mediation (Vygotsky 1978). The idea was crystallised in his famous triangular model in which the conditioned direct connection between stimulus (S) and responses (R) was transcended by a complex mediated act. The limitation of the first generation was that the unit of analysis remained individually focused. This was overcome by the second generation, based on Leont'ev's work (1978). Here Engeström (1999a) advocates the study of tools or artefacts as integral and inseparable components of human functioning. He argues that the focus of the study of mediation should be on its relationship with the other components of an activity system. The third generation of activity theory takes joint activity or practice as the unit of analysis for activity theory rather than individual activity.

Engeström's analysis is concerned with the process of social transformation and incorporates the structure of the social world, with particular emphasis upon the conflictual nature of social practice. Instability and contradictions are the motive force of change and development, and the transitions and reorganisation within and between activity systems are parts of the evolution. The aim of the third generation of activity theory is to understand dialogues, multiple perspectives and networks of interacting activity systems.

2.1 A brief overview

Activity theory focuses on the interaction of human activity and consciousness within its relevant environmental context (Vygotsky 1978; Leont'ev 1981). The basic unit of analysis in activity theory is human activity. Human activities are driven by certain needs where people wish to achieve certain purposes. The need in our example is to have a usable collaborative email application. It is obvious that activity cannot exist as an isolated entity. The very concept of activity implies that there is an agent who acts (an individual or collective 'subject'). An activity is undertaken by a subject (individual or subgroup) using tools to achieve an object (objective) thus transforming objects into outcomes. The subject is the users of the email application. The object in our example is the undeveloped email application. The outcome is the finished email application. Relations between elements of an activity are not directed, but mediated.

The relationship between subject and object of activity is mediated by a tool. A tool can be anything used in the transformation process, including both material tools and tools for thinking. Transforming the object into an outcome requires various tools (e.g., computer, software, methods, idea, procedure, internet, paper, pen etc.). In our example, tools include: computer, programming tools, methods, procedures, technologies, Internet, paper, pen etc). The object is seen and manipulated not 'as such', but within the limitations set by the tools (Kuutti 1996). Artefacts are created and transformed during the development of the activity itself and carry with them a particular culture - a historical remnant of that development.

The relationship between subject and the community is mediated by rules. Rules cover both implicit and explicit norms, conventions and social relations within a community as related to the transformation process of the object into an outcome. Rules in our case consist of organisational practices and policies, working hours, working regulation, etc). The relationship between object and community is mediated by the division of labour - how the activity is distributed among the members of the community, that is, the role each individual in the community plays in the activity, the power each wields and the tasks each is held responsible for. The roles in the email system consists of manager, secretary, users etc. Each of the mediating terms is historically formed and opens to further development (Kuutti 1996). The basic structure of an activity can be illustrated as in Figure 1.

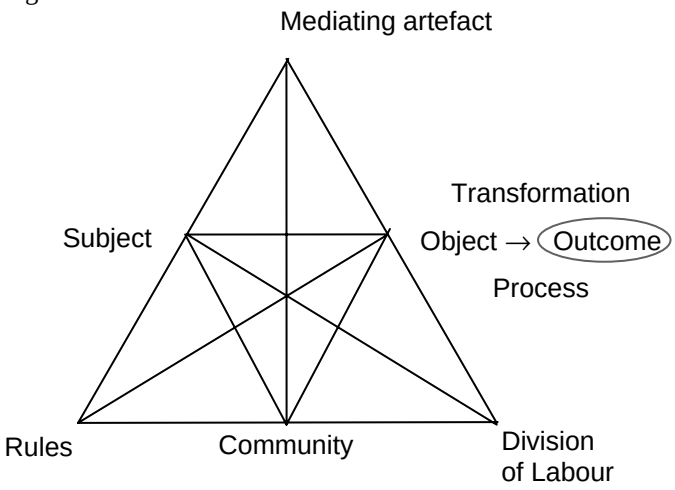


Figure 1: Basic structure of an activity.

In activity theory, the human mind emerges, exists and can only be understood within the context of human interaction with the world and this interaction, i.e., activity, is socially and culturally determined (Kaptelinin *et al* 1999).

According to Kuutti (1996) activities can be considered as having three hierarchical levels: activity, action and operation, which can be individual or cooperative. They can be considered as corresponding to motive, goal and conditions. An activity (global) may be achieved through a variety of actions. The same action may be used as contribution to different activities. Similarly, operators may contribute to a variety of actions. (See Figure 2). Kuutti uses a simple example of these levels to describe the activity (motive) of 'building a house' in which 'fixing the roof' and 'transporting bricks by truck' are at the action level and 'hammering' and 'changing gears when driving' are at the operation level. Every activity has an internal and external component with the subject and object existing as part of a dynamic and reciprocal relationship.

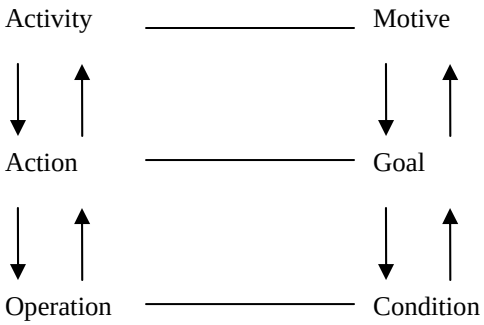


Figure 2: The three levels of activity.

Activity has double nature, both an external and internal side. The subject and object of an activity are in a reciprocal relationship with each other. The subject transforms the object. Conversely, the properties of the object penetrate into the subject and transform him or her. This is called internalisation (Kuutti 1996).
Activities are not isolated units; they are like nodes in crossing hierarchies and networks. They are influenced by other activities and other changes in their environment. According to Kuutti (1996), external activities change some elements of activity, causing imbalances between them. Contradictions are used to indicate a misfit between elements, between different activities or different development phases of the same activity. Contradictions manifest themselves as problems, ruptures, clashes and breakdown. Activity theory sees contradictions as a source of development. Activities are always in the process of working through some of these contradictions.

2.2 Advantages of activity theory for email design

There are several advantages of applying activity theory for collaborative email design. According to Bardram (1998a) these include:

- Activity theory provides a philosophical framework for understanding collective human work activities as embedded within a social practice (e.g. an organisation), and mediated by artefacts, including computer-based artefacts.
- By building on a dialectical notion between doing and developing work, activity theory provides a foundation for understanding both the dynamics of cooperative work changing over time and for understanding changes in work caused by employing new technology.
- The same conceptual basis can be used by activity theory to reflect on the user interface, cooperative work activities and the design process.

3 Implications of activity theory for design

Basic principles of activity theory include object orientedness, internalisation/externalisation, mediation, contradiction and development (for detailed discussions see Engeström 1987; Kuutti 1996; Kaptelinin *et al* 1999). The most immediate benefit of activity theory is in providing a triangular template for describing these relationships and looking for points of tension as new goals, tools or organizational changes create stress with the current roles, rules and artefacts. Some of the principles of activity theory that have important implications for collaborative email design are:

3.1 Design is evolving

The design of a collaborative email system requires an understanding of psychological, social and cultural phenomena. It has to comprehend development as a basic feature. The design approach of the email application is concerned with making artefacts for human use. Design is a complex set of technical and social components and relationships that together constitute an activity system (Engeström 1987). Design in activity theory is not a conscious goal or aim. It is not even a single object, but an ensemble of elusive and constantly changing objects, both material and ideal (Zappen & Harrison 2005).

3.2 Design as mediated activity

Design is a heteropraxial activity involving groups of people with different backgrounds and motivation in the process. During design, users and designers constitute a reified, implicit common understanding of the prototype. In activity theory, all human endeavours are mediated by socially constituted artefacts (Engeström 1987; Leont'ev 1978). This means that email design is mediated by artefacts. The designer (subject) shapes the design object by means of some design artefacts. The design object is the artefact produced in the design, the outcome the design activity is directed to. Design activity is mediated by design artefacts such as programming languages, methodologies, theories, technologies, etc. The prototype in collaborative design serves two purposes. It is the continuously moving object of the design activity and is also a design artefact mediating the creation of insights and vision of the new system (Bertelsen 2000).

3.3 Design artefacts mediate across heterogenous activities

In activity theory heterogeneous activities of the different users contribute to design by tying together through their joint use of artefacts and their joint focus on the same object. Design artefacts tie different communities of practice together, maintaining meaning across groups, but by making sense in different ways to different groups. They not only take different shapes and serve different purposes to different groups; they also take different functions with one group across time, during use and design.

3.4 The hierarchy of activity structures must be understood

Activity should be the unit of analysis in design of email application. This is a conceptual level about the level at which most business analysis takes place, i.e. at the level of action, which is undertaken towards specific goals (Hasan 2000). Typically in most computer systems, actions which are routine and standardised can become automatic when driven to a lower level of operation under certain conditions.

Email use may be the core business activity at the top level. In activity theory, the email application is not an end in itself, but, more often; it is a support for other business activities at all three levels in the activity theory structure. Email management is an explicit adjunct to core business activity, with value adding projects such as customer relations management is at the second level. These systems are viewed by activity theory as actions towards specific goals, but not as core business activities themselves. The third level in the activity theory hierarchy is that of operations where email systems are seen as primary tools automating basic organisational management processes.

3.5 Analyse collective activity through context

One of the main limitations of the traditional approach to designing interfaces based on the cognitive science approach may be due to their omission of context. This notion of context needs to be conceptualised. Kuutti and Juustila (1998) stress the importance of focusing on work activities as the context of information systems, saying, "We are never developing information systems, but the whole of the work activity where it will be utilised." How do we conceptualise work activities? The highest level of contextualisation is usually the task level. Task analysis identifies the outer behaviour of work activities and is a popular basis for defining the uses to which a computer interface will be put. This analysis may have an important function for describing job requirements. However, the distinction between human and computer tasks such as analysis is rather limited in relation to identifying the psychological processes in work activities. Focusing on observed behaviour does not say much about the inner structure of activity, as the same observed behaviour may correspond to different motives and goals of the individual. Operating a computer can be a playing, learning or working activity, thus having a different personal sense for subjects. We believe that studying cognition only within its task context does not solve the problem of contextualisation. Human procedures are not determined by the task, but determination is based on special characteristics of the case. For task analysis to have any real significance in design, it needs to be embedded within the work activity. It is impossible to make a general classification of activities, actions or operations because activities are in a constant state of development. The identification is independent of the activity of the individual.

It is important in a collaborative email system to use a collective activity system as the unit of analysis, giving context and meaning to seemingly random individual events. In activity theory, activity and context cannot be separated. The activity system itself is the context. What takes place in an activity system composed of object, actions and operations, is the context. Context is constituted through the enactment of an activity involving person (subject) and artefacts. Context is therefore the activity system and the activity system is connected to other activity systems. People consciously and deliberately generate contexts (activities) in part through their own objects. Context cannot be reduced to an enumeration of people and artefacts. In activity theory, context is not persistent and fixed information. Continuous construction is going on between the components of an activity system. Humans not only use tools, they also continuously renew and develop them either consciously or unconsciously. They not only use rules, but also transform them. It is generally acknowledged that understanding the social and organisational context is critical to the success of systems. The usability of a product depends on its context of use. Products should be designed for a specific context (Maguire 2001). The role of context of use within usability is required by the International Standard ISO 13407 (ISO 1999).

3.6 Historically analyse the activity and its components

An activity system does not exist in a vacuum; it interacts with a network of other activity systems. For example, a project team (activity system) receives rules and instruments from business activity, its members are trained by educational activity and it produces outcomes that are being used for activities in other organizational settings. An activity is also situated in time besides in a network of influencing activity systems. It is important to investigate its temporal interconnectedness (Pettigrew 1990). History is the basis of classification. This means that the activity system and its components shall be understood historically. An activity is not a homogeneous entity. It is comprised of a variety of disparate elements, voices and viewpoints (Engeström 1990). The multiplicity can be understood in terms of historical layers. Activities are not static or rigid, they are constantly evolving. To understand a phenomenon means to know how it is developed into its existing form (Kaptelinin 1996). This applies to all the elements of an activity. The current relationship between subject and object includes a condensation of the historical development of that relationship (Kuutti 1996).

3.7 Contradictions in activity systems

Activity systems are interrelated, providing each other with input and serving as instruments for each other. Contradictions are inevitable, occurring within and between activity systems; they lead to transformation of the processes. Activity is constantly developing as a

result of contradictions and instability, and due to the construction of new needs. Activity theory understands human beings as dialectically recreating their own environment. Subjects are not merely choosing from possibilities in the environment, but actively creating the environment through activity.

According to Engeström (1987), any activity system has four levels of contradictions that must be attended to in analysis of a working situation. Level 1 is the primary contradiction. It is the contradiction found within a single node of an activity. This contradiction emerges from tension between use value and exchange value. It permeates every single corner of the triangle and is the basic source of instability and development (Engeström 1987). Primary contradiction can be understood in terms of breakdowns between actions or sets of actions that realises the activity. These actions are poly-motivated. This means that the same action can be executed by different people for different reasons or by the same person as part of two separate activities. This poly-motivation may be at the root of subsequent contradictions.

Secondary contradictions are those that occur between the constituent nodes. For example, between the skills of the subject and the tool he/she is using, or between rules and tools. Tertiary contradiction arises between an existing activity and what is described as a more advanced form of that activity. This may be found when an activity is remodelled to take account of new motives or ways of working. Quaternary contradictions are contradictions between the central activity and the neighbouring activities, e.g. instrument producing, subject-producing and rule producing activities.

Contradictions are present in every collective activity. They indicate emergent opportunities for the activity development. Contradictions are not weakness, but signs of richness, and of mobility and the capacity of an organisation to develop rather than function in a fixed and static mode. They are not points of failure or deficits within the activity system in which they occur. They reveal the growing edge of the activity system – the place where growth buds are able to expand and expansive development takes place (Foot 2001), and are starting places, not ending points. Contradictions are not problems to be fixed, and they cannot quickly transcend through technical solutions. Engeström (2001) defines contradictions as historically accumulating structural tensions within and between activity systems. It is important to identify all the different kinds of contradictions in the email activity. In order to analyse an activity system's development, it is important to identify contradictions. By identifying the tensions and interactions between the elements of an activity system, it is possible to reconstruct the system in its concrete diversity and richness, and therefore explain and foresee its development (Engeström 1999b).

4 Collaborative email design

In XBC, the management of email is important for the company. Microsoft Outlook Exchange is currently being used by XBC. A shared folder is organised so that documents can be accessed by all users that need them. It is currently impossible to track who has updated these documents, since documents updated by users are not made known to the other users. The secretary, Rita, needs to send large documents containing product pictures to multiple clients. Rita often uses the sent folder to find product images to send to the different customers. This means that she cannot delete the sent emails because she needs to keep track of her sending activities. Consequently she uses up all her allocated 20Mb quota. Because Rita has to send multiple file attachments, this prevents her from sending out files more than 5Mb to customers. Rita also uses the email system to maintain her task list, scheduling, study notes, evens management, etc.

John is the manager of the organisation. He likes thread-based email and uses it to track related information. Instead of using Microsoft Outlook Exchange, he uses G-mail, a web-based email product from Google to implement the conversation-based email facility. This results in incompatibility between his system and Rita's, causing many problems, especially integration of information and management of tasks. A collaborative email application was proposed as the solution to overcome the above problems using semantic web. Due to the limited size of this paper, it only concentrates on the requirements analysis of the collaborative email application. The semantic aspects of the development will be discussed in a further paper.

The design of a collaborative email application is basically a socio-cultural phenomenon. It cannot be based solely on the systematic application of quantitative software measures, or any other methods from ideal natural science. The design of a collaborative email application has to include an understanding of psychological, social and cultural phenomena. It has to comprehend development as a basic feature. The design approach of collaborative email application is concerned with making artefacts for human use. Collaborative email application development research is based on a multitude of research methods and strategies such as intervention, field studies, theorising and controlled experiments. It is complicated by many factors, making it necessary for the application of a broad spectrum of modes of enquiry.

Based on the above activity theory framework, the following questions that are relevant for the design of collaborative email application include:

- What are the activities, goals and sub goals to be supported by system?
- What social context elements are to be considered?
- How can we model the collaborative email users?

According to activity theory, the email application is considered as artefact that mediates activities that are related to, or executed during, knowledge management in an organisation. In activity theory, artefacts and activities are in a reciprocal relation. New artefacts cause new or changed goals and activities.

The basic idea of Activity Theory is that an individual's relation to the surrounding world is not immediate, but is always mediated by culturally created artefacts. Individuals use both conceptual and practical tools to plan and realise their actions. Individual's actions are therefore always situated in a culturally-determined context and are impossible to understand outside of that context. We believe that only having a better understanding of human activity will allow us to conceive and design more flexible systems, responsive to human needs and use.

We can regard the development of collaborative email application an assembly of human beings and artefacts being changed or reconstructed to satisfy some motivation. It is therefore important to understand the individual components involved in order to understand the whole process.

5 Requirements analysis for collaborative email design using activity theory

Activity theory helps structure analysis, but does not prescribe what to look for. Activity theory does not offer ready-made technologies and procedures for research (Engeström 1993). Its conceptual tools must be concretised according to the specific nature of the object under study.

Designing email based upon activity requires in-depth understanding of tasks associated with collaboration. The best way to identify these task lists is to observe observing the way people work with the email system. The shared tasks users perform are also affected by factors such as their environment, experiences and culture etc., so addressing these issues is very important.

According to Bellotti and others (2003), a task management system in email should have the properties of differentiating important or outstanding items, indication for updated information, keeping track of threads of activity and discussions, methods to manage deadlines and reminders, embedding task features with in email and gathering related items. Based on her work, it is important the following should be considered when designing a task-based email system.

1. There should be easy way to differentiate important and outstanding items.
2. Days left indicator should be properly shown.
3. Use of conversation thread-based system
4. Mentioning the deadline and remainders.
5. Documents or files should be correlated accordingly with the email message.
6. Task-generated to-do list.

5.1 Clarify the purpose of the activity system

The purpose of this step is twofold: (a) to understand the context within which activities occur and (b) to reach a thorough understanding of the motivations for the activity being modelled, and any interpretations of perceived contradictions. Engeström (1987) emphasises clarification of the motives and goals of the activity system. What are stakeholders' goals and motives? What are their expectations about the outcome? We consider this stage to be the most important step of the process. Several techniques can be used at this initial stage, including the analysis of formal and informal documentation, user observations and interviewing. Given that the application developed must meet users' needs, a thorough understanding of the intentional dynamics of the activity system is critical. Activities always take place with a specific context in a certain situation. Activity context can be modelled using Engeström's Triangle (Engeström 1987) as a network of different elements that influence each other.

It is important to have a clear understanding of the goals of the email application to be built. The goals will help to define the object of the problem that users have. The motives will help to determine what perspectives to represent in the design. For our email application, we would examine users' problems for different uses, what were their problems, their motives and expectations? It is necessary to understand relevant context(s) within which activities occur. From this we can generate a list of problems that users typically have to deal with. We also have to understand the subject, his or her motivations and interpretations of perceived contradictions in the system. This will enable us to generate a list of subject-driven motives and goals for each of the groups involved that might derive the activity.

5.2 Analyse the activity system

This step involves defining, in depth, the components of the given activity, namely, the subject, object, community, rules and division of labour. This study began by interpreting the various components of the activity triangle (Figure 1.) in terms of the situation being examined. The activity system for XBC is shown in Figure 3. It is important to know the perception of the users as their roles in relationship to the goals of the system. Central to the analysis is the identification of the object of the system. The object here is to develop the collaborative email application. Having a clear idea of the object will to fulfil the goals or intentions of the system. The community of practice in XBC comprises the activity system. The community and its rules determine the problem context, and the division of labour determines with whom the user must interact with when working at XBC.

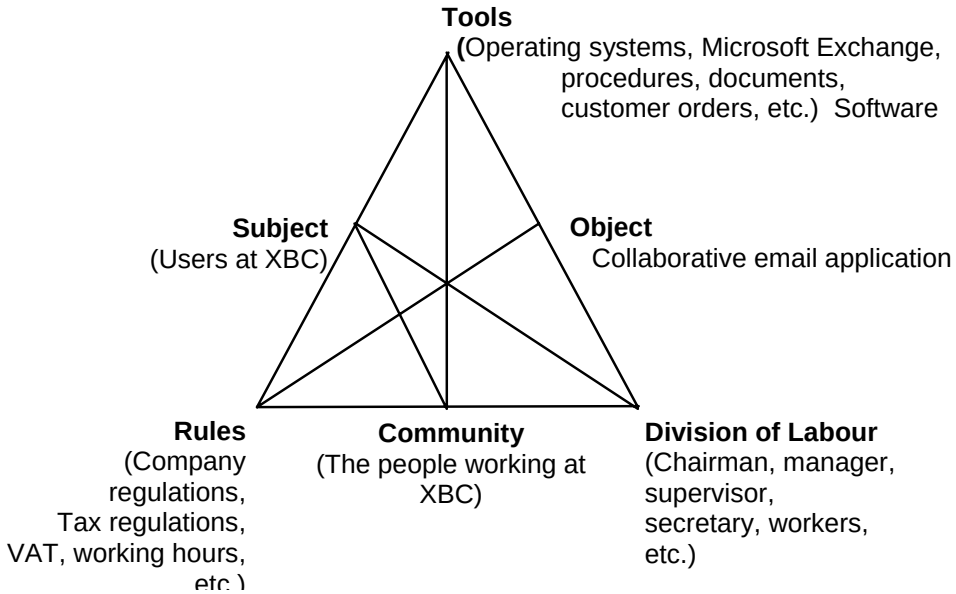


Figure 3: The activity system of XBC.

5.3 Produce an activity system of the application

The above information gathered enables us to acquire basic knowledge about the situation. This is necessary for the purposes of mapping Engeström's model (Figure 1) onto the situation in order to produce an activity system of that situation. This approach helps us to identify areas to be focused on during the investigation and also in deciding what resources would be necessary during the analysis.

5.4 Analyse the activity structure

It is necessary at this stage to analyse the activity structure (all of the activities that engage the subject) that defines the purpose of the activity system. The hierarchy of activity, actions and operations describe the activity structure. This is not unlike traditional task analysis. when defining and identifying activity structures, it is useful to have an understanding of the intentionality of the action or operation of the users. Why are people doing this? The outcome of this step will consist of a description of the activities, actions, and operations that are required to solve the email problem for XBC. To analyse the activity structure requires that we define the activity itself. It is important to identify the activities in which the subjects participate and how the work (actions and operations) have been transformed over time. This means how the work is actually done in practice and what historical phases have there been on the work activities. This is then followed by the decomposition of activities into components actions and operations. For each activity, observe and analyse the actions that are performed and by whom. Conversely, for each action, observe and analyse the operations that the subjects performed.

5.5 Analyse tools and mediators

Components of activity systems (subject, community, object) do not act on each other directly. Instead, their interactions are mediated by tools that provide direct and indirect communication between the objects. Mediators can be instruments, signs, procedures, machines, methods, languages, formalism and laws. Important questions to ask are: What tools are used in the activity? How available are the tools to the users? How have the tools changed over time? Mediators also include formal rules or models. Rules mediate the relationship between the subject and the community or communities in which they participate. It is necessary for us to know what formal or informal rules, laws, or assumptions guide the activities in which people engage? Besides rule mediation there is also role mediation. Who traditionally has assumed the various roles? How does that affect work group?

5.6 Decompose the situation's activity system

The activity system produced so far can be very complex because it incorporates the various sub-activities that together make up the main activity system being analysed. An activity notation can be used to aid the process of breaking down the situation's activity triangle system into smaller manageable units or sub-activity triangles (Mwanza, 2001). Figure 4 shows the activity notation. Each combination within the activity notation should consist of:

- An 'actor' – represented by the ***subject*** or ***community*** component of the triangle model.
- A 'mediator' – represented by the ***tools***, ***rules*** or ***division of labour*** component of the triangle.
- The ***'object'*** on which activity is focused.

Each combination within the activity notation represents a complete sub-activity triangle from the main activity system. For example, it is possible to identify the ***subject-rules-object*** sub-activity triangle representation whose mediated relationship could be arranged in terms of the application of rules as shown in Figure 2.

Actors (Doers)		**Mediator**		**Objective** (purpose)
Subjects	~	Tools	~	Object
Subjects	~	Rules	~	Object
Subjects	~	Division of labour	~	Object
Community	~	Tools	~	Object
Community	~	Rules	~	Object
Community	~	Division of labour	~	Object

Figure 4: Activity Notations.

5.7 Generate questions for each activity notation

Questions that are specific to a particular combination within the activity notation and also representing a sub-activity triangle are then generated. The questions generated can be general or specific to a particular situation. Questions that are specific to a particular combination within the activity notation and also representing a sub-activity triangle are then generated.

5.8 Analyse the context

The traditional approach to analysis ignores real life contexts within which activities take place. Activity theory argues that activity itself is both defined by and defines context. Context is both internal to people (involving particular goals or objects), and also external (involving artefacts, other people and settings). Analysing context is essential for defining the larger activity systems within which activity occurs (subject, community, and object) and the dynamics that exist between the subject and the mediators. The designer is seeking information in order to describe "how things get done in this context". Why? Because different contexts impose distinctly different practices.

There are two types of contexts that need to be identified: internal or subject-driven contextual bounds and external or community driven contextual bounds. (Jonassen & Rohrer-Murphy 1999), Questions to be asked are:

- What are the beliefs, assumptions, models and methods that are commonly held by the users?
- What tools do users use in doing their tasks : How well do they use them?
- What is the structure of the social interactions surrounding the activity?
- What limitations are placed on the activity by the company or outside agencies?

These questions can also be general or specific to a particular situation.

The six general research questions:

i. What **Tools** do the **Subjects** use to achieve their **Object**ive and how?
ii. What **Rules** affect the way the **Subjects** achieve the **Object**ive and how?
iii. How does the **Division of Labour** influence the way the **Subject**s satisfy their **Object**ive?
iv. How do the **Tools** in use affect the way the **Community** achieves the **Object**ive?
v. What **Rules** affect the way the **Community** satisfies their **Object**ive and how?
vi. How does the **Division of Labour** affect the way the **Community** achieves the **Object**ive?

Examples of specific research questions for XBC:

How does the email *(tools)* support collaborative task management *(object)* amongst teams *(subject)?*

How do rules of XBC affect collaborative task management *(object)* amongst individuals and teams *(subject)*?

How do roles of people at XBC *(division of labour)* affect the way collaborative task management *(object)* is achieved amongst the teams *(subject)*?

How does the use of outlook *(tools)* as performance indicators affect the way XBC *(community)* support task management *(object)*?

How does XBC's *(community)* use of email influence the way the organisation manage their tasks (object)?

- How are the tasks organised among members who are working towards the object?
- How are tasks divided or shared among the participants?
- What formal or informal rules guide the activities in which people engage?

5.9 Identify the different types of contradictions

Contradictions can be identified by disturbances in the free running of an activity (Engeström 1999b). In order to identify contradictions, it is necessary to understand the dynamics of the current work and make visible its nuances and identify the disturbances therein. Contradictions are present in every collective activity. They indicate emergent opportunities for the activity development. Contradictions demand creative solutions. The contradictions identified for the SSIL email application are shown in Figure 5. Contradictions are important aspects of an activity because they are used as sources of development (Kuutti 1996). They trigger reflection, thereby helping with the improvement of the activity. Several levels of contradictions were identified at XBC.

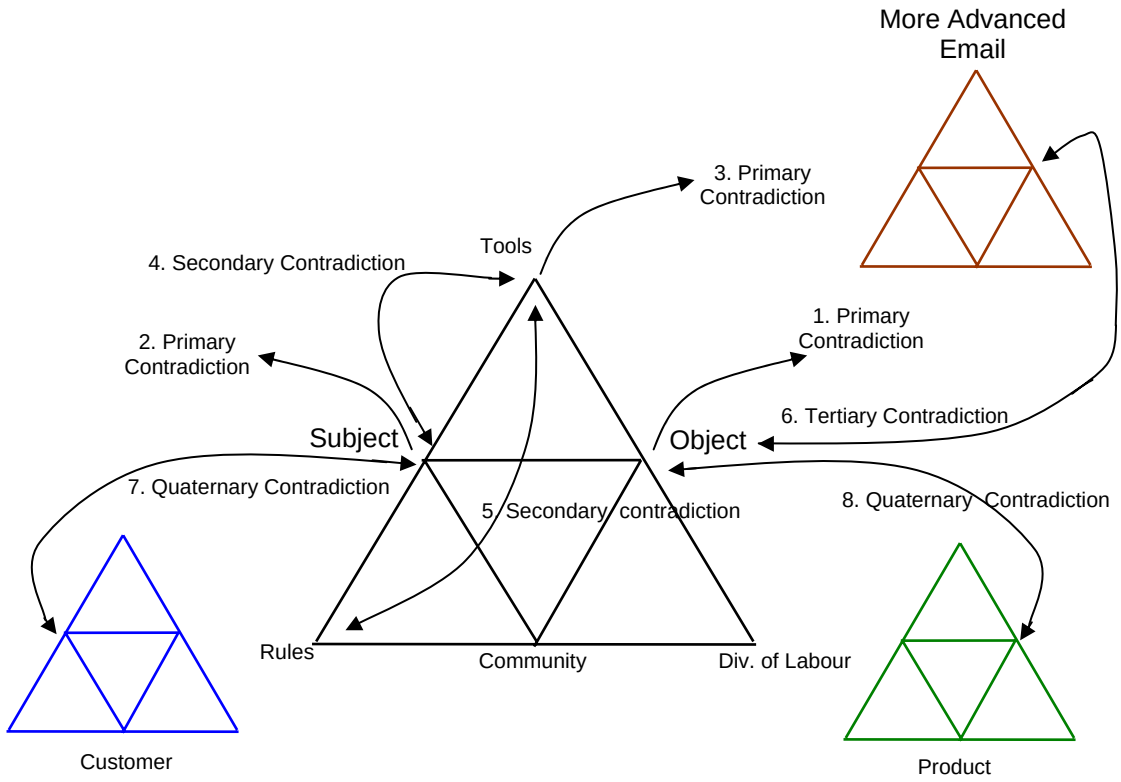


Figure 5: Contradictions of the XBC system

5.10 Potential primary contradictions within the email activity

1. At the object node, there is tension between the requirements of the users. Rita and John have different objectives.
2. At the subject node, the skills and experience of the users vary as a function of their history, experience and training.
3. At the tool node, there are conflicts of interest between the choice of technologies used for the development of the collaborative email system, e.g. traditional approach versus semantic web technologies.

5.11 Potential secondary contradictions within the email activity

4. There is tension between the subject node and the tool node. The subject (the developer) may not have the necessary semantic web skills to produce the collaborative email application.
5. There is a conflict between the choice of tool and the rule of the organisation. Semantic web technologies may cost more than the limited budget the company is able to pay.

5.12 Potential tertiary contradictions within the email activity

6. There is tertiary contradiction between the existing activity and the new collaborative email activity because the subjects would have to learn the new system.

5.13 Potential quaternary contradictions within the email activity

7. There is a fundamental contradiction between the effective mail management and supporting staff at the customer department.
8. There is a contradiction between the aim of the product development and the interests of the staff using the email system.

Semantic web offers open-standards that can enable vendor-neutral solutions, with a useful flexibility (allowing structured and semi-structured data, formal and informal descriptions, and an open and extensible architecture). RDF can be used as a common interchange format for catalogue metadata and shared vocabularies that can be used by all libraries and search engines across the web.
The above identified contradictions were resolved before we designed the email application using semantic web. A screen shot of the Semantic email database schema is shown in Figure 6 on next page.

There are benefits for using semantic web for the design of our collaborative email application. Semantic Web enables automated information access and use based on machine-processable semantics of data. The application was evaluated by the different users of XBC. They found it very useful because it enables them to take care of task management as well as normal email activities.

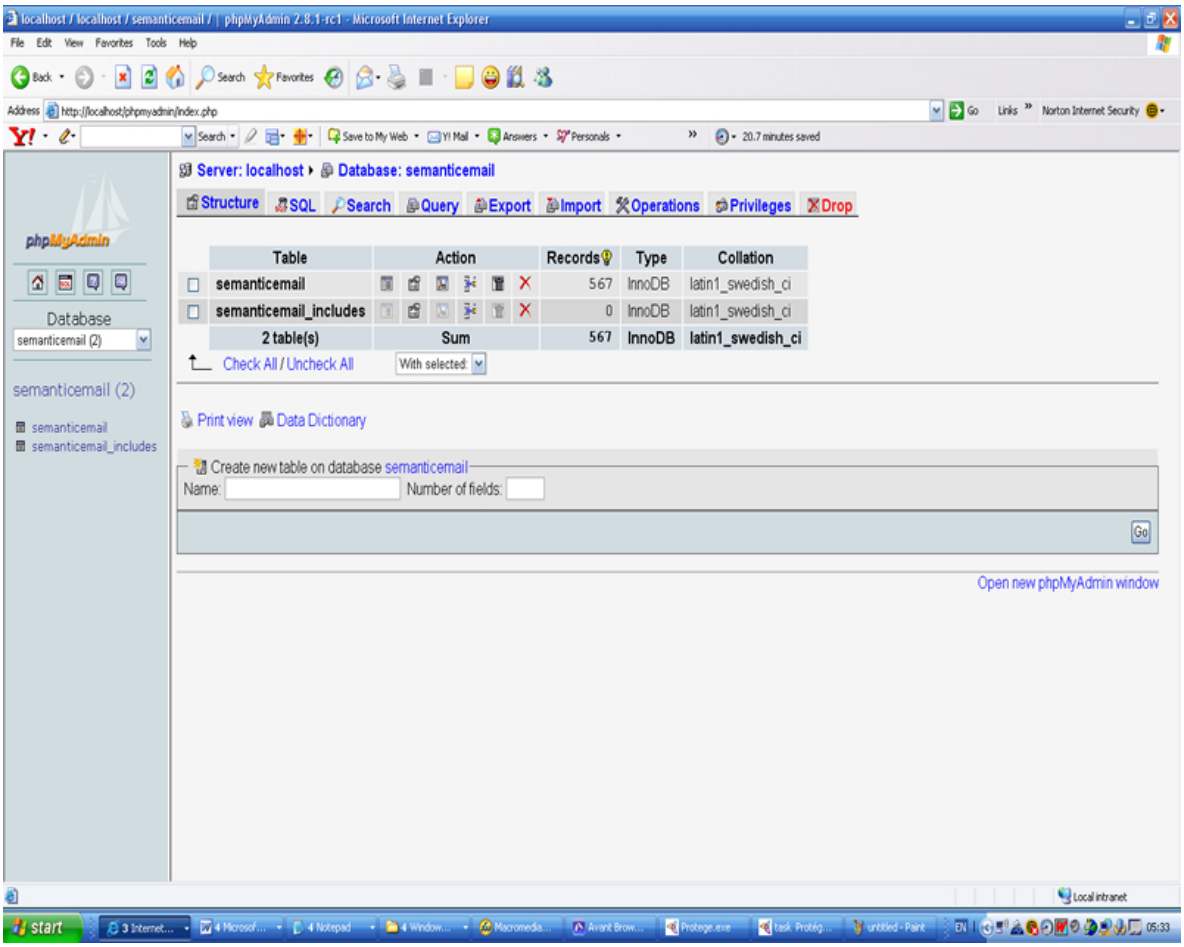


Figure 6: Semantic email database schema

6 Conclusion

The development of effective email applications is a socio-cultural activity. A technical solution is not adequate to address the complexity of the system. To design collaborative email systems without consideration of the different needs of the users' social processes is a recipe for disaster. Email systems are inevitably groupware systems that connect people to people either directly or indirectly through sharing knowledge. To support effective collaborative email systems, it is necessary to understand the interrelationship of cultural, technical and organisational elements. While this is beginning to change, there remains a substantial research challenge in developing activity theory and tools to apply in the design of applications to support work such as such email.

Software systems are not built in a vacuum, but within organisational environments where outcomes are heavily influenced by a myriad of internal and external social-technical factors. Softer issues should be given the same weight in software development and implementation processes as the more technical features. The research presented here is underpinned by these concerns.

Activity theory principles are ideal for making visible the structure and dynamics of work situations, especially with respect to contradictions. Contradictions provide a systematic way of modelling and reasoning about breakdowns and opportunities for email design. The strength of the activity theoretical perspective is the recognition that work systems are inherently dynamic. However, further work is still needed for activity theory to be used as a robust requirement or design method. More research would be needed. The authors are currently working on making the principles of activity theory concrete so that anyone without activity theory knowledge can use the proposed guidelines for requirements analysis.

6.1 Acknowledgement

We acknowledge the collaboration with the XBC Ltd company in providing the business case for this study.

7 References

[1] Al-Rawas, A.. & Easterbrook, S. (1996). Communications Problems in Requirements Engineering: A Field Study. Proceedings of he First Westminster Conference on Professional Awareness in Software Engineering (The Royal Society, London, 1-2 February.

[2] Avison, D.E. and Wood-Harper, T. Multiview: An Exploration in Information System Development. Alfred Walter, Oxford, 1990.

[3] Bannon, L. (1991). From Human Factors to Human Actors: the role of psychology and human-computer interaction studies in system design. In J. Greenbaum & H.M Kyng (eds.), Design at Work: Cooperative Design of Computer Systems. Hillsdale, NJ: Lawrence Erlbaum Associates.

[4] Bannon, L. & Bødker, S. (1991). Beyond the Interface: encountering artefacts in use. In J.M. Carroll (ed.), Designing Interaction: Psychology at the human-computer interface (pp 74-102), Cambridge: Cambridge University Press.

[5] Bellotti, V., Ducheneuaut, N., Howard, S. & Smith, I. (2003). Taking Email to task: the design and evaluation of a task management centred email tool. CHI 2003, April 5-10, 2003, Fort Lauderdale, FL: USA. Vol. 5(1), pp 345-352.

[6] Bertelsen, O.W. (1998). Elements of a Theory of Design Artefacts: a contribution to critical system development research. PhD Thesis. Aachus, Jan. 1998.

[7] Bødker, S. (1991): Activity theory as a challenge to systems design. In: Nissen H-E, Klein HK and Hirscheim R (eds.): Information Systems Research: Contemporary Approaches and Emergent Traditions, Amsterdam: Elsevier, pp. 551–564.

[8] Ducheneuaut, N. & Bellotti, V. (2001). Email as Habit: An exploration of Embedded Personal Information Management. Interactions, 8(5) ACM Press, pp30-38.

[9] Engeström, Y. (1987). Learning by Expanding. Helsinki: Orienta-Konsultit OY.

[10] Engeström, Y. (1990). Developmental work: Research as activity theory in practice: Analysing the work of general practitioners. In Y. Engeström, Learning, Working and imaging: Twelve studies in activity theory, Orienta-Konsultit OY, Helsinki.

[11] Engeström, Y. (1993). Developmental studies of work as a testbed of activity theory. In C. Chaiklin & J. Lane (eds.), Understanding Practice: perspectives on activity and context. Cambridge: Cambridge University Press, pp 64-103.

[12] Engeström, Y. (1999a). Innovative learning in work teams: analysing knowledge creation in practice. In Y. Engeström, R. Miettinen, & R-L Punamaki (eds.), Perspectives on Activity Theory: Learning in Doing: Social, Cognitive and computational perspectives. Cambridge University Press, UK: pp 377-404.

[13] Engeström, Y. (1999b). Expansive Visibilization of Work: An Activity-Theoretical Perspective. Computer Supported Cooperative Work. Journal of Collaborative Computing 9(1-2), pp 63-93.

[14] Engeström, Y. (2001). Expansive Learning at work: towards an activity theoretical reconceptualisation. Journal of Education and Work, Vol 14 no 1.

[15] Floyd, C. (1987). Outline of a paradigm change in software engineering. In G. Bjerknes, P. Ehn, & M. Kyng (Eds.), Computers and democracy – A Scandinavian challenge (pp. 191-212). Aldershot, UK: Avebury.

[16] Foot, K.A. (2001). Cultural-Historical Activity Theory as Practical theory: Illuminating the development of a Conflict Monitoring Network. Communication Theory Vol. 11, Feb 2001, pp 56 – 83.

[17] Goguen, J.A. and Linde, C. (1993). Techniques for requirements elicitation. Proceedings of the First IEEE International Symposium on Requirements Engineering (RE'93), 152-164.

[18] Gwizdka, J. (2001) Supporting Prospective Information in Email. Proceedings of CHI'2001, ACM Press, pp 135-136.

[19] Hargreaves, D. (1996). Teaching as a Research-Based Profession. London: TTA.

[20] Hasan, H. (2000). The mediating role of technology in making sense of information in a knowledge intensive industry, Knowledge and process management, 6/2, p. 72-82.

[21] ISO (1999). ISO 13407: Human-centred design processes for interactive systems. Geneva: International Standards Organisation. Also available from the British Standards Institute, London.

[22] Jonassen, D.H. and Rohrer-Murphy, L. (1999). Activity theory as a framework for designing constructivist learning environments. Educational Technology Research and Development, 47(1), 62-79.

[23] Kaptelinin, V., Nardi, B. A., Macaulay, C., 1999. The Activity Checklist: A tool for representing the "space" of context. ACM Interactions, 6 (4), 27-39.

[24] Kaptelinin, V., (1996). Activity Theory: Implications for Human-Computer Interaction, In Context and Consciousness: Activity Theory and Human-Computer Interaction. B. Nardi, (ed.), Cambridge, Mass.: MIT Press.

[25] Khoussainov, R. & Kushmerick, N. (2005). Email Task Management: an iterative relational learning approach., http://www.ceas.cc/papers-2005/142.pdf, accessed on 20.11.2006.

[26] Kuutti, K. (1991). Activity Theory and its applications to information systems research and development. In H.E. Nissen, H.K. Klein & Hirschheim (eds.), Contemporary Approaches and Emergent Traditions. Elsevier Science Publications. North-Holland.

[27] Kuutti, K. (1996). Activity Theory as a Potential Framework for Human Computer Interaction Research. In B.A. Nardi (ed.), Context and Consciousness: Activity Theory and Human-Computer Interaction. MIT Press, Cambridge, MA.

[28] Kuutti, K. & Vikkunen, J. (1995). Organisational Memory and Learning Network Organisation: the case of Finnish Labour Protection Inspectors. Proceedings of HICSS 28.
[29] Kuutti, K. & Juustila, T.M. (1998). Information System Support for 'Loose' Co-ordination in a Network Organisation: an Activity Theory perspective. In H. Hasan, E. Gould & P. Hyland (eds.), Information Systems and Activity Theory: tools in context. University of Wollongong Press.
[30] Leont'iev, A.N. (1978). Problems of the Development of the Mind. Moscow, Progress.
[31] Leont'ev, A.N. (1981). Activity, Consciousness and Personality. Englewood Cliffs, Prentice-Hall.
[32] McGrath, M. & Uden, L. (2000). Modelling Softer Aspects of the Software Development Process: An Activity Theory based approach Thirty-third Hawaii International Conference on System Sciences. (HICSS-33) Wailea, Maui, Hawaii, USA - Software Process Improvement. IEEE Computer Society Press. January 2000.
[33] Maguire, M. (2001). Context of Use within usability activities, Int. J. Human-Computer Studies (2001) 55, 453}483 doi:10.1006/ijhc.2001.0486, Available online at http://www.idealibrary.com
[34] Mwanza, D. (2001). Where theory meets practice: A case for an Activity Theory based methodology to guide computer system design. Proceedings of INTERACT 2001: Eighth IFIP TRC13 conference on Human-computer interaction, Tokyo, Japan, July 9-13, 2001.
[35] Nardi, B.A. (ed.), (1996). Context and consciousness: Activity Theory and human-computer Interaction. Cambridge, MA: MIT press.
[36] Pettigrew, A. M. (1990). Longitudinal field research on change: theory and practice. Organization Science 1(3): 267-292.
[37] Probert, S.K. (1999). Requirements engineering, soft system methodology and workforce empowerment. Requirements Engineering, 4, Springer-Verlag, London, 85-91.
[38] Vygotsky, L.S. (1978). Mind in Society. Harvard University Press, Cambridge, MA.
[39] Whittaker, S., & Sidner, C. (1996). Email overload: exploring personal information management of email. In Proceedings of CHI'96, Conference on Human Factors in Computing Systems, ACM, NY, 276-283.
[40] Zappen J.P. & Harrison, M (2005). Intention and Motive in Information-System Design: towards a theory and method for assessing users' needs. P. van den Basselaar & S. Koizumi (eds.), Digital Cities 2003, LNCS 3081 pp 354-368. Springer-Verlag, Berlin, Heidelberg, 2005.

Semantic Web Based Integration of Knowledge Resources for Supporting Collaboration

Vili Podgorelec, Luka Pavlič and Marjan Heričko
Institute of Informatics, University of Maribor, FERI,
Smetanova ulica 17, SI-2000 Maribor, Slovenia
E-mail: vili.podgorelec@uni-mb.si

Keywords: knowledge management, semantic web, collaboration, knowledge sharing

Received: October 27, 2006

In the paper we present the importance of collaboration between researchers for the improvement of their creativity. A unified methodology to support collaboration strategies of researchers and research teams based on knowledge sharing is introduced. We argue that a defined methodology together with an efficient technological system supporting the methodology should improve the creativity of research teams and consequentially facilitate the development.
Based on the required functionalities of such a system, we propose the semantic web as the underlying technology. It is indicated how the semantic web technologies could provide the necessary solutions for the integration of data resources, the transformation of data into valuable knowledge, the effective use of knowledge by intelligent information services and knowledge sharing both within an organization and inter-organizationally. Finally, the prototype architecture of an intelligent agent within a database system is outlined, which serves as an information integration mechanism.

Povzetek: Semantična integracija virov znanja v podporo sodelovanju.

1 Introduction

Creativity is a complex cognitive activity for the performing of which both motivation and knowledge are required. A motivation is partially provided by a working environment in which a researcher or a research team performs its activities. The most important aspects for a high motivation are:

- effective fulfillment of conditions required for creativity, in which the researchers are able to optimally put into practice their potentials, and
- efficient support system that enables researchers to solve their problems and/or overcome obstacles, which they may experience during the creativity process.

Naturally, for the successful solving of problems the second factor of creativity process is of vital importance, namely knowledge. If the researchers are supposed to be creative, they need to possess the knowledge that will enable the creativity. Many organizations performed various studies which confirmed that their research staff is professionally highly skilled, however, their creative results were not comparable with the leading teams. The reason could be an inadequate or an ineffective approach to collaboration between single researchers and/or research teams when performing more complex research projects.

It is hard to believe and yet true that researchers within an organization usually do not know the knowledge and skill profiles of their peer researchers. Consequentially, the problem occurs when individual researchers, who possess a good amount of individual knowledge, face a problem they are not able to solve by themselves and that could be effectively solved by some of their peers. Many a time a solution is not reached because of an inadequate or even non-existing knowledge sharing. On the other hand, the researchers who can not put their knowledge and skills into practice become less and less motivated.

In both situations the consequence is a lower creativeness and in the worst cases even the total suppression of the creative energy. And without creativity there can not be any real development. Knowledge sharing and collaboration is deficient within organizations, even within research departments and institutes. On the inter-organizational level, there is almost the total lack of systematically organized and planned collaboration. Regarding this, the definition and development of a proper methodology supporting collaboration of researchers, effectively initiated into research and development departments, could importantly contribute towards higher creativity and consequentially to faster and more efficient development.

2 Knowledge based collaboration between researchers

Based on the awareness of how important creativity is for the efficient development and the vital part of knowledge

within this process, a lot of researchers studied different aspects of knowledge management and assets management systems for knowledge capturing, representation and sharing [Art06]. Various theories on exploitation of knowledge sources within organizations and workgroups are being introduced by scientists; however, they are not technically and technologically supported. On the other hand many companies decide to set up a knowledge management system that is not efficiently used to their advantage, because of the lack of adequate methodology.

From the technological point of view, the necessity of transition from software products to services has been globally recognized. In this manner, there are many attempts to set up a knowledge-based system based on ontologies and semantically annotated data. Nevertheless, usually the ontologies are used primarily for statically describe data repositories. In the very near future we predict the intelligent approaches to more complex information services which will need to make advantage of semantically annotated repositories.

The literature search showed that presently there is no completely defined and technologically supported methodology to support researchers' collaboration strategies based on knowledge. However, a set of single approaches and solutions gives evidence of a possibility to define and develop such a methodology. In our opinion, there are several attempts which individually provide basic components for further development.

An interesting approach to bridging communities of practice with information technology in pursuit of global knowledge sharing is presented in [Pan03]. A similar approach to knowledge sharing in an emerging network of practice is presented in [Baa05]. They both suggest a use of knowledge portals, which are an extension of well known and recently much used business portals for managing all important business data. An interesting framework for stimulating innovation is presented in [Bre05] that gives evidence for the importance of properly technologically supported methodology of collaboration to improve the creativity. The importance of collaboration based on knowledge is recognized also in [Lom06], where the authors suggest a framework to manage formalized exchanges during collaborative design. The inter-organizational resource sharing decisions in collaborative knowledge creation is especially emphasized in [Sam06]. We also proposed a possible solution to build project teams based on knowledge with the use of information technology [Pod06].

3 Methodology to support collaboration of researchers

The research problem that needs to be addressed is the definition and development of a proper methodology to support strategies of collaboration between researchers based on knowledge management and assets management. Such a methodology should contribute to a more optimal access to knowledge and competences within a research area. Furthermore, it should improve the creativity of researchers and lead towards more efficient development, both from the organizational and the technological point of view.

3.1 Outline of collaboration methodology based on knowledge sharing

We plan to achieve the proposed methodology by using one of the most vibrant of today's information technologies, namely semantic web. In our opinion the semantic web is a proper technology to bridge the technological gaps, outlined in the present approaches to a unified knowledge-based collaboration methodology.

The most important aspect of semantic web technologies is the semantics of data, which allow efficient integration of information resources (both existing and forthcoming) and a possibility of automatic, intelligent inferring on knowledge, retrieved from those information resources.

Having the technology for the efficient integration of information resources and the technology for semantically annotate these data, a step to transform the data into knowledge becomes possible. Having a unified access to knowledge resources of an organization (i.e. a logical organizational unit, like a department) an information service that uses this knowledge becomes reasonable. Having the information services which use the knowledge resources of an organization to provide the useful functionalities a system to share knowledge within an organization and also inter-organizationally exceeds mere theories and hoped-for ideas. The schematic outline of the proposed collaboration methodology based on knowledge sharing is presented on Figure 1.

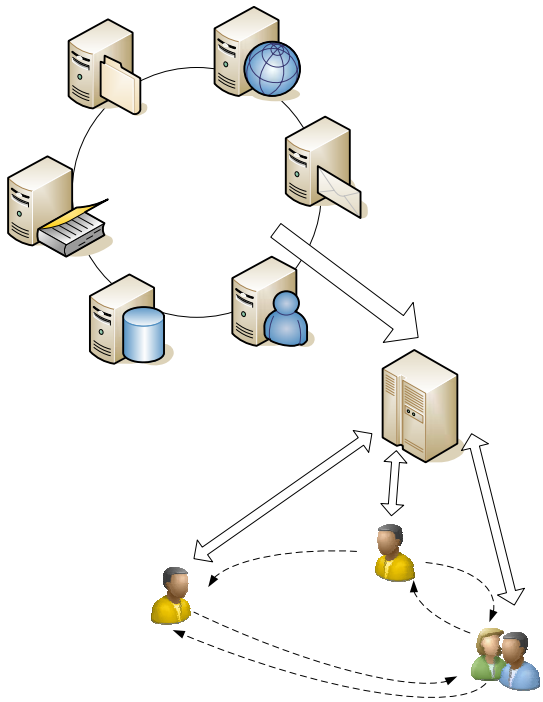


Figure 1: Collaboration based on knowledge sharing.

3.2 Managing individual competences to improve organizational development

Additionally to sharing organizational knowledge in order to improve the overall creativity and development within an organization, it is also important to manage individual knowledge competences in order to establish a knowledge map of an organization [Col03]. In this case a researcher or a research group within an organization is able to locate those who possess the required expertise for solving a specific task or performing a special activity. In this way not only the creativity could be considerably increased, but also the knowledge and skills of individual researchers are improved, because they learn at first hand from colleagues which master a specific issue.

On the other hand, when performing research projects, a project leader or team members can easily recognize which colleagues are appropriate for performing different project tasks. Also the hidden skills, not directly stated in a profile of an individual researcher, can be discovered by a proper skills matching approach. Again, this is possible by having all the data integrated and semantically annotated, what allows the support system to automatically infer on the stored data. A part of the knowledge portal for managing individual profiles is presented on Figure 2.

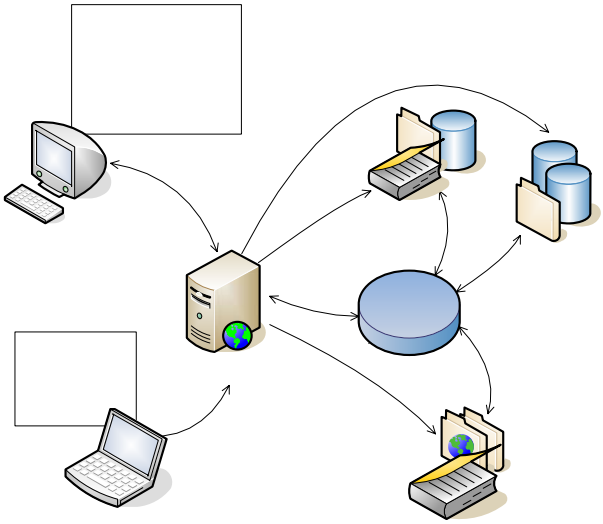


Figure 2: Semantic knowledge portal for managing personal skills profiles.

4 Semantic web as the underlying technology

In order to fulfill the requirements, which are necessary to achieve the proposed methodology, semantic web technologies are in our opinion a very sound choice. The semantic web represents one of the most vibrant of today's information technologies. As it turns out, it is very appropriate for the integration of information resources by semantic annotation of data. Furthermore, the idea of semantic web allows automatic, intelligent inferring on knowledge, retrieved from these information resources.

4.1 Semantic web technologies

The basic idea of semantic web is a different organization and storage of data and consequentially new possibilities to use this data [Ber01]. Although the idea of semantic web is based on well established concepts, such as machine learning and automatic reasoning, the semantic web community has given these areas fresh new move by introducing web-based solutions. The web is still very primitive and actually provides quite useless organization of knowledge, especially when one want to do some searching or knowledge discovery. The barrier that prevents more advanced usage of the web is believed to be semantic poorness of today's world wide web. Data, documents, images and all other kinds of content on the web are presented as very simple, non structured human readable and human understandable materials. The result is the inability to make a real use of the web's enormous amount of "knowledge". Because it can be understood as a huge cross-referenced library, all we have is by default a weak tool called keyword search.

In order to overcome those difficulties the concept of meta-data is introduced on the web. Using meta-data, so called smart agents can be used for searching by content. As a foundation, there has been a lot of work done about common formats for interchange of data and common understanding of common concepts. That allows a person or a machine to browse, understand and use knowledge on the web in a more straightforward way. All those activities and technologies are known by the term "semantic web".

Furthermore, the semantic web ideas and technologies can be used in other areas also, not only on globally available web. They can be used in the enterprise information systems for knowledge management in a different way to introduce new intelligent services.

As already mentioned, we want knowledge (with its meaning!) to be accessible to both people and machines. It is obvious that we need to represent knowledge in a more formal way. There are quite a lot of possibilities. The most appropriate for semantic web were chosen semantic nets. They are very simple nets, consisting of linked concepts. The question is what we need to represent distributed knowledge, such as we have on the web? We need a standardized way of naming things. Two different things should have different names and vice versa, when we talk about the same thing we need to use the same name. Furthermore, we need a standardized way of saying something about things – we need a standardized way of describing things. Also we need common vocabularies. If we talk about coin and bank note, for example, we should automatically know that we are talking about money. And finally, we also need a standaprdized way of giving semantics to data, or said

more technically, we need a standardized technology to connect data with some meta-data.

In semantic web, knowledge is represented as nets, written down in XML-based language called RDF (Resource Description Framework). RDF is dealing with URIs (another W3C standard for naming resources globally unique). Advanced use of semantically annotated data can only be accomplished by using ontologies represented as a RDFS or OWL documents. The whole stack of technologies for semantic web is shown on Figure 3.

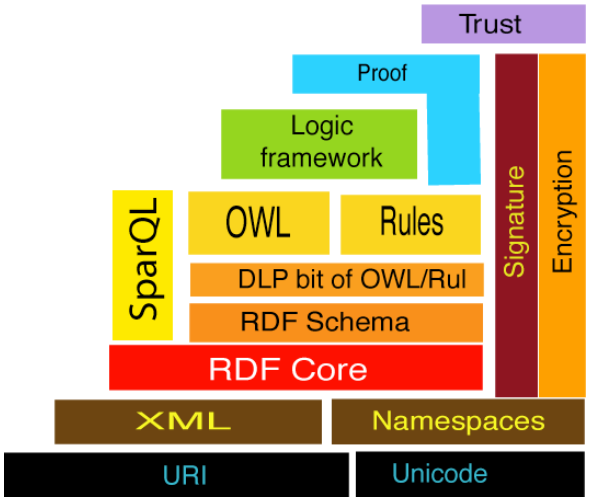


Figure 3: The complete stack of semantic web technologies as proposed by Tim Berners-Lee and W3C.

4.2 The key role of ontology in knowledge management

What is the purpose of ontology in semantic web? Ontology describes the subject domain using notions of concepts, instances, attributes, relations and axioms. In [Gru93] authors define ontology as a formal explicit specification of a shared conceptualization. It is a useful way to organize and share information while offering an intelligent means for knowledge management. Ontology also enhances semantic search in distributed and heterogeneous information services. Ontologies are the key player, if we want to do (automatic) search in more advanced ways, not only keyword search.

There are several benefits of using ontologies for information solutions. Semantic search engines return instances that constitute answers to queries rather than documents containing search strings as in keyword search engines. Semantic search uses meanings (semantics) of the query terms defined in the ontology. The data of ontology constitutes precise answers to user questions. Users can further browse related concept because answers are interconnected through semantics. It can be speculated that using ontology supported systems users will also be able to invoke functionalities or query data using free text input in the future.

4.2.1 The definition of domain knowledge and personal skills ontology

In order to adopt the semantic web technologies in our pursuit of implementing the proposed methodology, there are two key fields which need to be addressed: domain knowledge that we want to share between researchers and the personal skills profiles of the researchers. For those two areas the ontologies need to be defined, which will then allow all further actions, like semantic annotation of data (in accordance with the ontology), integration of data resources, advanced searching and inferring on the data.

The definition of domain knowledge is closely inter-related with the definition of personal skills for this domain. For example: let's imagine the field of software engineering where we want to describe the programming skills of researchers. If we want to adequately define the technical skills of a Java programmer, it is important to know at least the basic attributes of Java programming language. On the other hand, if we want to describe a "knowledge item" such as Java source code, it is very useful to link it with the knowledge requirements of producing such an item. Only in this manner knowledge sharing (intelligent searching, inferring, etc) could be successful.

Naturally, the definition of domain knowledge for a specific area may vary, but there have been some attempts to define personal skills ontology. One of the most important features of semantic web is the fact, that the defined ontology can be easily broadened when a need occurs for a new concept, not previously identified. In this manner, a working system can be expanded with new knowledge on the fly.

4.2.2 Ontology-based personal skills management

An overview of the related work in ontology-based personal skill management is presented in [Bie05]. Already [Sta99; Jar99] promoted the idea of ontology-based modeling of personnel skills and job requirements – as part of comprehensive, workflow-oriented enterprise modeling. There, the following potential applications of ontology-based skill profiles are listed:

- skill gap analysis – at the enterprise level, as a part of strategic HR planning,
- project team building,
- recruitment planning – again a part of strategic HR planning, and
- training analysis – at the level of individual personnel development.

Those approaches were mainly technology-driven and were – to our knowledge – never realized in a large-scale industrial environment. Nor have they been accepted by the HRM departments, translated into HRM people's terminology, embedded into more comprehensive models and procedures of HRM people, and integrated with existing software infrastructures.

After those first publications, there were a number of interesting technology-oriented researches which showed that in particular skill matching can benefit from interesting technological approaches, such as background knowledge exploitation. For instance, [Liao99] employs declarative retrieval heuristics for traversing ontology structures. [Sure00] derives competency statements through F-Logic reasoning and developed a soft matching approach for skill profile matching. [Colucci03] and others use description-logics (DL) inferences to take into account background knowledge as well as incomplete knowledge when matching profiles.

4.3 Integration of data resources

As we believe in the applicability of semantic web technologies for knowledge sharing, ontologies and semantically annotated data are used for describing personal skill profiles and domain knowledge. In this manner the existing knowledge within organizations can be reused, if it is appropriately converted into RDF in accordance with the defined ontology.

The proposed methodology envisages an integration of many data resources, containing the necessary knowledge. In this manner the integrated information resources can be seen as an inter-connected database. To allow an easy access to integrated knowledge resources, our technical solution enables one to use well-known queries, like SQL or XQuery, to access this integrated data universally (Figure 4).

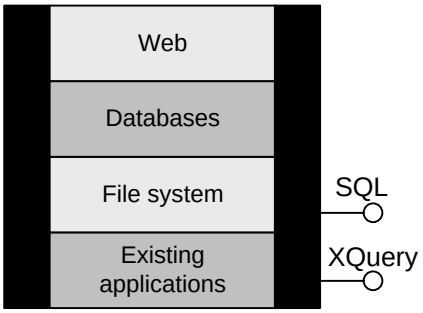


Figure 4: Universal access to integrated data resources.

4.3.1 The architecture of the system

The architecture of the system has been designed in a form of four inter-connected main components (Figure 5). RDF Provider is in the role of an intelligent agent that continuously acquires data from all available data resources. The collected RDFs are stored within a XML database, where they are used by the user interface component. This component is used for storing data, browsing, searching and inferring on the semantically annotated data. The communication with XML database is realized using standardized interfaces, like SQL and XQuery.

The stack of used technologies is presented on figure 6. The system prototype is implemented mainly in Java programming language using open source Jena semantic web development library [JENA]. It provides us with a straight-forward development system, very appropriate for semantic web portal application. Because the inferring technology, as represented in the stack of semantic web technologies proposed by W3c (Figure 3) is not available yet, for the inferring part of the system we chose CLIPS programming environment [CLIPS]. It is a production rule based programming system mainly used for developing expert systems. CLIPS is a productive development and delivery expert system tool which provides a complete environment for the construction of rule and/or object based expert systems. As it turned out, it enabled us with powerful inferring possibilities. Additionally, it is very easy to execute CLIPS rules within Java applications using open source libraries such as JClips [JCLIPS], what further enables one to integrate the inferring rules within an information system. The whole system itself does not consist of many different technologies, which is in our belief good. The fundamentals of the system lies in J2EE platform and XML enabled database.

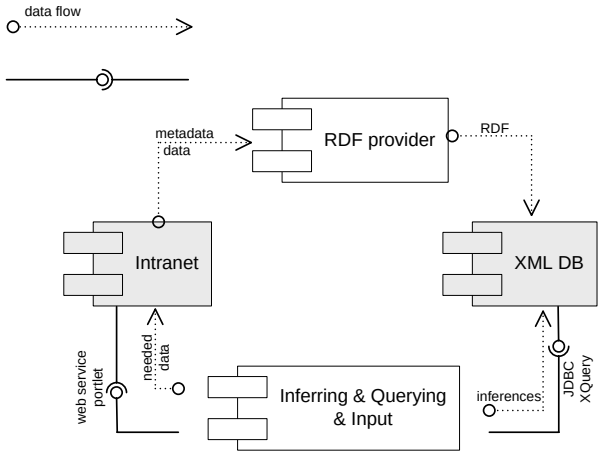


Figure 5: Main components of the systems and their inter-connections.

The most important component, called RDF provider, is responsible for collecting as many internal data in RDF as possible. As it continuously examines all possible data providers within the specified range and domain, it extracts data from existing applications, web pages, databases and file systems. Collected RDFs and presented ontology are persistently stored in XML database and prepared for analysis with reasoning system, based on J2EE, Jena and CLIPS.

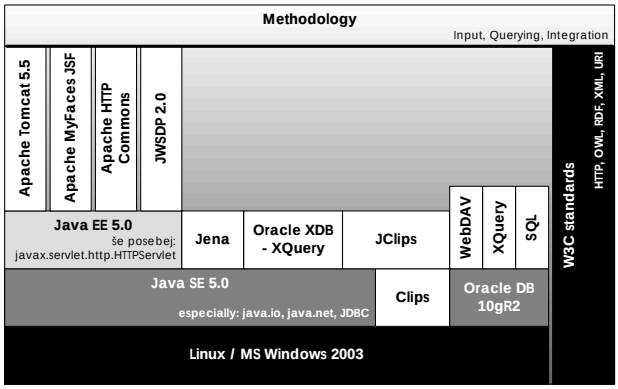


Figure 6: The stack of used technologies.

5 Issues to be resolved

In order to achieve the proposed methodology, a number of current challenges and aspects need to be addressed. Semantic web technologies as an underlying technological framework represent a vibrant new technology with high potential, and yet as a complex approach require several scientific and technological solutions. The semantic web potentials as a technique for integrating existing and forthcoming information solutions with semantics need to be addressed. Also the approaches to automatic annotation of data and the intelligent web services (like automatic discovery of hidden knowledge, automatic profiles matching, project teams building) are an issue. Because the intention is to provide efficient bridging of research communities with information technology based on knowledge, the possibilities of automatic construction of knowledge from data need to be studied (in combination with existing powerful approaches like data mining, text and web mining, knowledge discovery from data), as well as the linkage of ontologies and semantic repositories.

Finally, as of our knowledge, the challenge of reducing complexity by systematic linkage of research groups has not been adequately answered yet. It is our belief that the defined methodology can contribute a great deal to answering also this important question.

6 Conclusion

Our view of a unified methodology to support collaboration strategies of researchers and research teams based on knowledge has been presented in the paper. In order to improve the creativity of research teams and consequentially also facilitate the development, a defined methodology together with an efficient technological system supporting the methodology could be the right way.

Based on the required functionalities of such a system, the semantic web can be used as the underlying technology. It has been indicated how the semantic web technologies can provide the necessary solutions to the integration of data resources, the transformation of data into valuable knowledge, the effective use of knowledge by intelligent information services and knowledge sharing both within an organization and inter-organizationally.

When a proposed methodology is developed, it could be used to semantically describe the competence profiles of researchers, involved in research groups of various organizations. In this way a considerably better collaboration of researchers would be achieved within a scope of scientific, research and development activities. Furthermore, the researchers from academic institutions and industry could be efficiently inter-connected, what would in turn lead towards higher creativity and faster industrial development. In fact a lot of data, needed for the operation of such a system, is already available and stored within different databases (researchers' profiles, publications and research activities, project data and project teams data, description of research projects and their results, …) All this data only needs to be appropriately annotated and integrated, which is an inherent property of the proposed methodology. The existing information services, although not semantically annotated, could be used in efficiently integrated within the system by implementing the required interface wrappers.

References

[Art06] H.A. Artail, Application of KM measures to the impact of a specialized groupware system on corporate productivity and operations, *Information & Management*, 43(4), 2006

[Baa05] P. van Baalen et al., Knowledge Sharing in an Emerging Network of Practice:: The Role of a Knowledge Portal, *European Management Journal*, 23(3), 2005

[Ber01] T. Berners-Lee, Business Model for the Semantic Web, http://www.w3.org/DesignIssues/Overview.html, 2001.

[Bie05] E. Biesalski, A. Abecker, Integrated Processes and Tools for Personnel Development, *Proc. of 11th International Conference on Concurrent Enterprising*, Munich, Germany, June 2005

[Bre05] A. Brennan, L. Dooley, Networked creativity: a structured management framework for stimulating innovation, *Technovation*, 25(12), 2005

[Col03] S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, M. Mongiello, M. Mottola, A Formal Approach to Ontology-Based Semantic Match of Skills Descriptions, *Journal of Universal Computer Science*, Springer Verlag, 9(12), pp. 1437-1454, 2003.

[Gru93] T.R. Gruber, Towards Principles for the Design of Ontologies used for Knowledge Sharing, In N. Guarino & R. Poli (eds.), *Proc. of International Workshop on Formal Ontology*, Padova, Italy, 1993

[Jar99] P. Jarvis, J. Stader, A. Macintosh, J. Moore, P. Chung, What Right Do You Have to Do That?, *Proc. of ICEIS - 1st Int. Conf. on*

Enterprise Information Systems, Portugal, 1999

[Lom06] M. Lombard, L.G. Yesilbas, Towards a framework to manage formalised exchanges during collaborative design, *Mathematics and Computers in Simulation*, 70(5-6), 2006

[Pan03] S.L. Pan, D.E. Leidner, Bridging communities of practice with information technology in pursuit of global knowledge sharing, The *Journal of Strategic Information Systems*, 12(1), 2003

[Pod06] V. Podgorelec, L. Pavlič, M. Heričko, Using semantic web technologies for project team building, *Proc. of International Conference on Knowledge Management in Organizations KMO-2006*, June 2006

[Sam06] S. Samaddar, S.S. Kadiyala, An analysis of interorganizational resource sharing decisions in collaborative knowledge creation, *European Journal of Operational Research*, 170(1), 2006

[Sta99] J. Stader, A. Macintosh, Capability Modelling and Knowledge Management, *Applications and Innovations in Expert Systems VII*, Springer-Verlag, pp. 33–50, 1999.

[CLIPS] –, CLIPS – A Tool for Building Expert Systems, http://www.ghg.net/clips/CLIPS.html

[JCLIPS] –, JClips — CLIPS for Java, http://www.cs.vu.nl/~mrmenken/jclips/

[JENA] –, Jena – A Semantic Web Framework for Java, http://jena.sourceforge.net/

Designing New Ways for Selling Airline Tickets

Mladenka Vukmirović
Industry Development Department, Montenegro Airlines
Beogradska 10, 81000 Podgorica, Montenegro

Michał Szymczak
Department of Mathematics and Computer Science
Adam Mickiewicz University
Umultowska 87, 61-614 Poznań, Poland

Maciej Gawinecki
Systems Research Institute, Polish Academy of Science
Newelska 6, 01-447 Warsaw, Poland

Maria Ganzha,
Elblag University of Humanities and Economics, Elbląg, Poland
ul. Lotnicza 2, 82-300 Elbląg, Poland
and
Systems Research Institute, Polish Academy of Science
Newelska 6, 01-447 Warsaw, Poland

Marcin Paprzycki
Computer Science Institute, SWPS
Chodakowska 19/31, 03-815 Warsaw, Poland
and
Systems Research Institute, Polish Academy of Science
Newelska 6, 01-447 Warsaw, Poland

Keywords: software agents, air-travel ontology, travel support system, e-commerce, e-auctions, OTA, IATA

Received: October 12, 2006

Large body of recent work has been devoted to multi-agent systems utilized in e-commerce; in particular, autonomous software agents participating in auctions. In this context we modify a model agent-based e-commerce system so that it can serve as an airline ticket auctioning system. Such a system can be then combined with a Travel Support System that utilizes ontologically demarcated travel-content. To achieve this goal, air travel data has to be demarcated utilizing an air travel ontology that has to support existing domain-specific real-world standards. One of such standards that steadily gains popularity in the air travel industry (and other travel areas) is the Open Travel Alliance (OTA) messaging system that defines, among others, the way that entities should communicate about air travel related issues. The aim of this paper is to outline our efforts leading toward creating an agent-based system for selling airline tickets that utilizes an air-travel ontology that matches the OTA messaging specification as well as satisfies procedures described in IATA manuals.

Povzetek: Opisan je večagentni system za prodajo letalskih kart.

1 Introduction

Broadly understood e-commerce is often closely associated with software agents, which are to facilitate higher quality information, personalized recommendations, decision support, knowledge discovery etc. [27]. When developed and implemented, agent systems are to be, among others, adaptive, proactive and accessible from a broad variety of devices [42]; and as such are to deal autonomously with information overload (e.g. large number of e-shops offering the same product under slightly different conditions—price, delivery conditions, warranty etc.). Moreover, recent advances in auction theory have produced a general methodology for describing price negotiations [8, 9]. Combination of these factors gave new impetus to research on automating e-commerce [24]. In this context, we have started working on two independent research projects. The first one is devoted to the development of a model agent-based e-commerce system [2–5, 12 and references to our work cited in these papers]. In this system, we model a distributed marketplace where buyer agents approach e-stores and engage in price negotiations with seller agents. What makes our work unique is, among others, an attempt at conceptualizing not only price negotiations, but also a complete process from the moment when User-Cuyer

"decides" to make a purchase of product *P* to the successful purchase (or to a decision that such a purchase is impossible – e.g. due to the market conditions). The second project is an agent-based Travel Support System (TSS) [13, 39, 40]. In the TSS, travelers are to find complete support of their needs including, among others, items like restaurant information, historical points of interest, local weather etc. The central part of the TSS is a Jena-based repository [20, 21] that contains travel-related data is represented as RDF demarcated instances of a travel ontology [13]. Specifically, we have developed a complete ontology of a hotel (understood as a travel-related entity) and a restaurant; and then merged them [35]. The overarching goal of the design of the TSS was delivery of personalized information to users [13].

More recently we have asked, what would happen if our model e-commerce system had to be used in a more realistic scenario, where instead of an unspecified product *P,* airline tickets were to be sold and the system would have to interact with an actual airline reservation system. As a result we have proposed an augmented system in which two additional agents: a *FlightOffer Agent* (FOA) and a *Reservation Agent* were created to interact with Global Distribution Systems (GDS), e.g. AMADEUS or SABRE [1, 33] and facilitate delivery of all necessary air-travel related information.

In the next step we have considered how this augmented system could be integrated with the TSS. Since in the TSS travel data is stored as instances of travel ontologies (currently hotel and restaurant data), air travel related data should be also stored in the same way. Furthermore, air travel ontology that is to be used within the system should be tightly integrated with ontologies already existing in the system. After a thorough analysis of existing air-travel ontologies we have decided to develop our own [40].

The aim of this paper is to summarize our research results up to date. In the next section we present the augmented ticket auctioning system. We follow (in Sections 3 and 4) with a list of existing travel-related ontologies and a summary of the Open Travel Alliance (OTA) messaging system. OTA messages are then used as a starting point to design an air travel ontology, which is outlined in the next section.

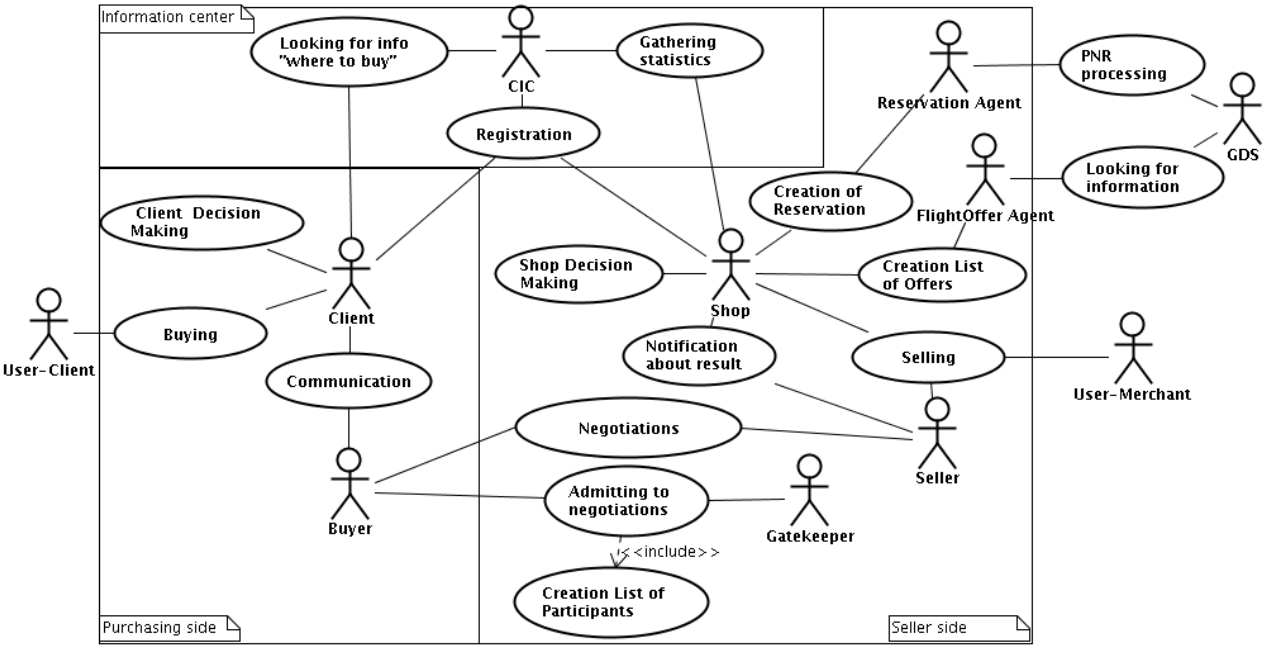


Figure 1: Airline ticket auctioning system – use case diagram.

2 Airline ticket auction system

Before proceeding with the description of the system, let us point some of the assumptions made in our work. (1) In our original agent-based e-commerce system e-stores were "drivers" within the marketplace. In other words, buyers could purchase only products that were available for sale through existing e-stores. We have decided, in the initial phase of our work on airline ticket selling system, to accept this approach (while planning to remove this limitation in the future). Therefore, in the augmented system, multiple "travel agencies" sell tickets to a variety of "popular destinations." They obey basic rules of airline ticket trading, but it is only "them" who decides which tickets to sell. Specifically, if the user of the system would like to fly from Tulsa, OK to San Diego, CA, she may not find such a connection. At the same time, connections between Amsterdam and Detroit, MI may be sold by every e-store. While this assumption may seem limiting, we would like to point out that success of priceline.com (and other auction places that sell airline tickets) makes our model scenario "realistic enough." (2) While we are utilizing the *CIC Agent* that stores "yellow-pages" (what?) and "white-pages" (who?) information as the approach to matchmaking [38], we see possible interesting extensions of its role in the system. It

could be possible to allow the *CIC Agent* to study market trends and sell this information to interested travel agencies. (3) In all situations where it was possible we utilize existing structures that have been described in [2–5, 10] and interested readers should consult these sources for further details.

Let us represent design of the system through its UML use case diagram in Figure 1 (detailed descriptions of the system can be found in [39]). We can distinguish three major parts of the system. (1) The *Information center area* where white-page and yellow-page information is stored and serviced by the *CIC Agent*. As specified above, currently, *User-Merchants* request that their e-stores sell tickets only for specific routes that they believe to be profitable. Each such route is advertised through the *CIC*. Every time the *Client Agent* is searching for an airline ticket for its *User-Client* it communicates with the *CIC* to find out which e-travel agencies sell it. (2) The *Purchasing side* where agents and activities representing the *User-Client* are represented. Here the *User-Client* informs its *Client Agent* which tickets she would like to purchase. While the *Client Agent* should be viewed as an incarnation of a *Personal Agent* [24] that knows preferences of its *User-Client* and autonomously acts on her behalf, their exact interrelations will be established in the future. *Client Agent* obtains from the *CIC* information which e-travel agencies sell requested tickets and sends a *Buyer Agent* to each one of them. *Buyer Agents* engage in price negotiations with *Seller Agents*. Successful price negotiations results in a reservation. *Client Agent* decides which agency to make a purchase from and, if the reservation did not expire and the tickets are still available in the GDS, they are purchased. (3) The *Seller side* involves *Shop Agent* acting on behalf of its *User-Merchant* and attempting at selling air tickets for routes defined by her. It interacts with the *FlightOffer Agent* in creating a list of specific offers that are registered with the *CIC*. Upon successful price negotiation the *Reservation Agent* creates and manages a reservation and, if this is to be the case, is responsible for completing the purchase. Observe that both the *FlightOffer Agent* and the *Reservation Agent* interact directly with the GDS. In this way they act as "wrapper agents" translating data between the outside world (the GDS) and the system. Let us now describe in more details the roles of these agents that have been added, or that act differently than in the original e-commerce system.

2.1 Shop Agent

Shop Agent (*SA*) acts as the representative of the *User-Merchant* and, at the same, time participates in the *Selling* function of the system. As specified above, in our current system design, it is the *User-Merchant* who specifies the input provided to the system. Specifically, for each route that is to be offered, she specifies: departure airport code, destination airport code, booking class, fare basis code, and the initial rule by which seats are to be offered for sale. For example, if *User-Merchant* wants to sell out seats that would have been offered for Advanced Purchase Excursion Fare—APEX [18, 19] but time limit for this fare has expired, *User-Merchant* would specify the number and the period for which she wants to offer seats on specific flights. This info would be used in availability check and price retrieval. The time-period would be needed to set bounds within which flights should be offered. Optionally *User-Merchant* can specify flight number as well. This narrows down the availability list and may be necessary in the case when there is more then one flight per day between two given destinations. Furthermore this can be used also in the case when, for instance, user-merchant wants to offer seats on morning flights, but not on evening flights. In this case she can specify which flight number(s) can be chosen from. In this way, all other possible flight numbers are excluded.

Obviously, it is possible to extend functionality of our system. For instance, while at present our system acts only as a "distributor" of a predefined set of tickets, it is possible to modify it in such a way that the *SA* could start distributing (acquire and put for auction) also tickets for routes that *User-Client*s are looking for. Since the *CIC* agent stores information about all unfulfilled *User-Client* queries, an *SA* could be enabled to obtain an access to this data (e.g. purchase it), analyze it and decide that, for instance, there is a growing need for tickets between Podgorica and Beijing and offer these for sale.

Statechart diagram of the *Shop agent* is depicted in Figure 2. At first the *SA* creates the *Gatekeeper Agent* (which plays here exactly the same role as described in [6]) and waits for a *User-Merchant* order. After receiving such an order the *SA* creates *FlightOffer Agent*, which communicates with the GDS and gathers needed information to create list of offers for the *Shop Agent* (one *FlightOffer Agent* is created for each route to be serviced and exists for as long as tickets for a given route are sold by the *SA*). List of offers includes information about every itinerary: data about both (inbound and outbound) flight numbers, number of seats and class of service for both flights etc. *Shop Agent* creates also *Seller Agent*(s), "introduces" them to the *Gatekeeper*, and enters a complex state called *Selling*. Note here that *Seller Agent*s play exactly the same role as that described in [6]; they are to interact with incoming *Buyer Agent*s and through some form of price negotiation mechanism (e.g. an auction) select the *Buyer* that may purchase the ticket. In the *Selling* state the *SA* is listening to its *Seller Agent*(s). After receiving a message from one of the *Seller Agent*s – informing about the result of price negotiations – the *Shop Agent* acts depending on content of that message.

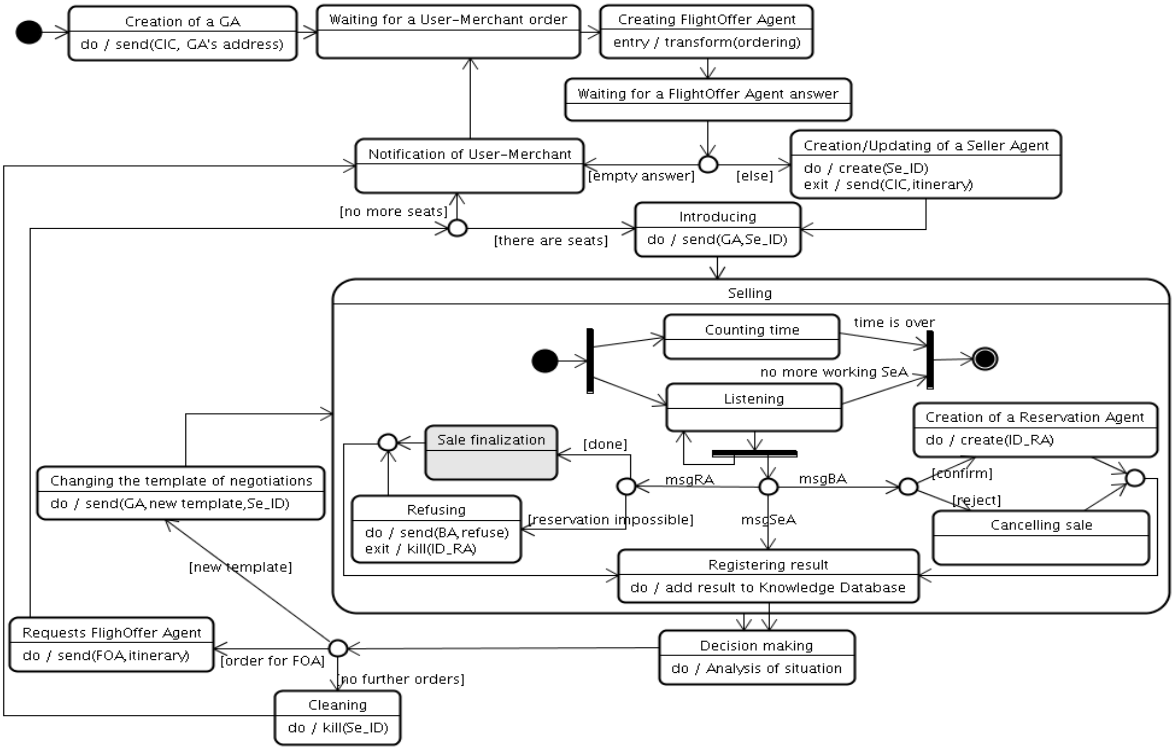


Figure 2: Shop Agent statechart diagram.

1. If the *Seller* informs the *SA* about a winner of price negotiations the *Shop Agent* waits for the corresponding *Buyer Agent* to confirm that it plans to actually buy the ticket (see also [10] for more details). Here, we have to stress, that in our general e-commerce model it is natural that multiple *Buyer Agent*s visit multiple e-stores [10]. Specifically, separate *Buyer* visits each e-travel agency that offers ticket(s) satisfying needed itinerary. The end of price negotiation means that the *Buyer* should consult with the *Client Agent*. Therefore, the *SA* does not know if the winner of price negotiations will actually attempt at making a purchase.
2. If the *Buyer Agent* confirms it wants to buy a ticket, the *Shop Agent* creates a *Reservation Agent* (*RA*), which communicates with the GDS to make a reservation. There are then the following possibilities:
   - If the *RA* was able to reserve tickets (it is possible that while the negotiations were taking place all tickets available in a given class of service etc. are already gone), it sends the reservation data to the *Shop Agent*. Upon reception of the data (all communication in the system is carried using ACL messages) the *Shop Agent* transfers it further to the *Buyer Agent* and carries out standard procedures involved in completing the sale (Figure 1, state "Sale finalization").
   - In the opposite case (the *RA* was not able to secure the reservation) the *Shop Agent* notifies the *Buyer Agent* that reservation is impossible and kills the *Reservation Agent*.
3. If the Buyer Agent sends message that it does not want to make a purchase, this fact is registered in a local Knowledge Database. More precisely, all information about processes that take place within the shop when it is attempting to sell tickets is recorded in the Knowledge Database. In the future, this information will be used by the *SA* to adapt its behavior. Currently we denote this fact by introducing the *Decision Making* box, which denotes multi-criterial decision making. For instance, one of important factors that influences the way that the *SA* interacts with incoming *BA*s is trust (see for instance [7, 28]). It should also be mentioned that in our system we utilize a modified negotiation framework [3, 4, 6] introduced originally by Bartollini, Jennings and Preist [8, 9]. In this framework, the negotiation process was divided into a generic *negotiation protocol* and a *negotiation template* that contains parameters of a given negotiation. These parameters specify, among others, the price negotiation mechanism itself. Observe, in Figure 2, that one of possible results of *Decision Making* is change of the negotiation template. In other words, the *SA* may decide that since only very few tickets are left but there is also only very short time to sell them, it will deeply discount them and sell them with a fixed price, or through a very short time lasting English auction with a low threshold value and a relatively large increment.
4. If there is no winner, the *Shop Agent* writes information into the *Knowledge Database* and starts to analyze the current situation (the *Decision Making* box in Figure 2). As a result it may change the negotiation template, or request another itinerary from the *FlightOffer Agent*. Finally, it may establish that for that given route (*User-Merchant* order) either there is nothing more to do (all tickets have been sold) or that nothing can be done (the remaining tickets cannot be sold in the current condition of the market). Then it will remove all

"servant" agents servicing that route and inform its *User-Merchant* about the situation. It is important to note that we assume that in all price negotiation mechanisms the *Seller* institutes a time limit for negotiations. This moment is presented within the *Shop Agent* diagram as a sub-state "Counting time" (within the *Selling* state). If the *Seller* does not sell any tickets within that time the *Shop Agent*, again, registers this information in the *Knowledge Database*, kills this *Seller* and notifies its user-merchant accordingly. Following, the *SA* enters the *Multi-criterial Decision Making* state. As described above, here it can decide, among others, to sell more seats on some specific itinerary or to change the template of negotiations or to conclude that nothing more can be sold and its existence should be completed.

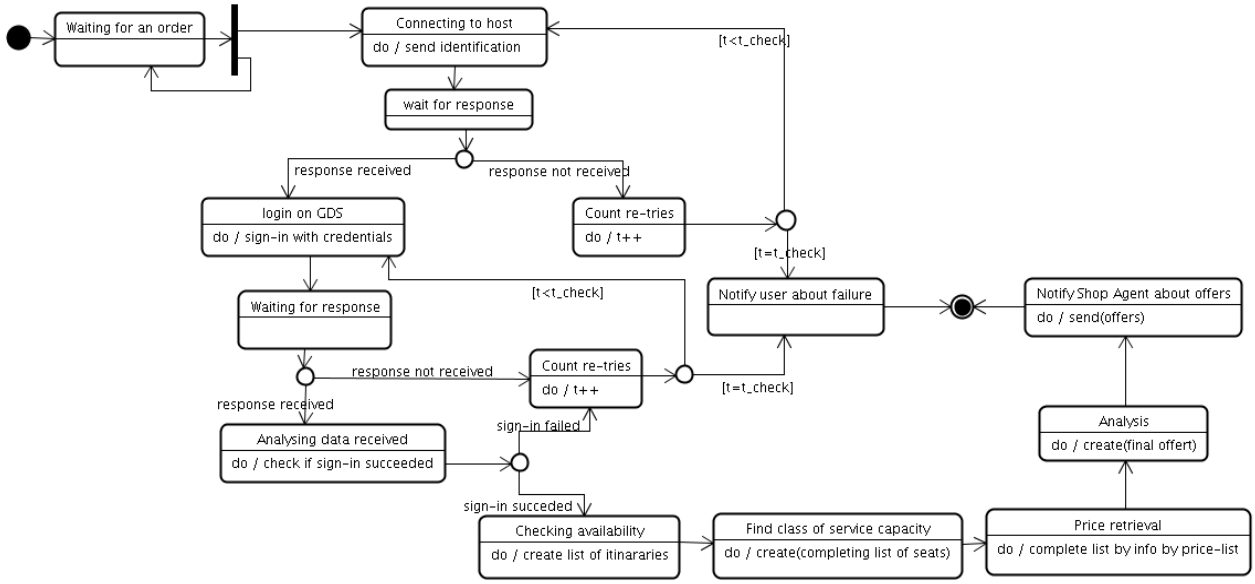


Figure 3: *FlightOffer Agent* statechart diagram.

2.2 *FlightOffer* and *Reservation Agents*

These two agents have been added to the system and their role is to communicate with the GDS. The statechart diagram of the *FlightOffer Agent* is presented in Figure 3.

This agent communicates with the GDS to find information about flights that satisfy conditions specified by the *User-Merchant*. If such flights are available the *FlightOffer Agent* prepares (process represented by actions that are enclosed within multi-state boxes *Checking availability*, *Find Class of service capacity*, *Price retrieval* and *Analyzing module*) a *List of Offers* for the *Shop Agent*. All the multi-state states—*Checking availability*, *Find Class of service capacity*, *Price retrieva*l and *Analyzing module*—involve communication with the GDS. In Figure 4 we present the statechart of the *Price retrieval* sub-state to illustrate the nature of proposed communications between the *FlightOffer Agent* and the GDS. Upon obtaining all the necessary information form the GDS it sends the information to the *Shop Agent*. Note that the role of the *Analyzing module* is to check the request of the *User-Merchant* against the data retrieved from the GDS to assure consistency of the final offer (e.g. if the *User-Merchant* requested 10 seats, but only 5 are available then only 5 can be in the offer). The second agent that communicates with the GDS is the *Reservation Agent*. It is created by the *Shop Agent* after receiving, from the *Buyer Agent*, confirmation of willingness to make a purchase. Its function is to make an actual reservation within the *GDS* server. In case of successful completion of its task the *Reservation Agent* transfers all reservation's data to the *Shop Agent*. If the reservation is impossible it informs about it the *Shop Agent*. Both cases mean that its job is complete and it then self-destructs.

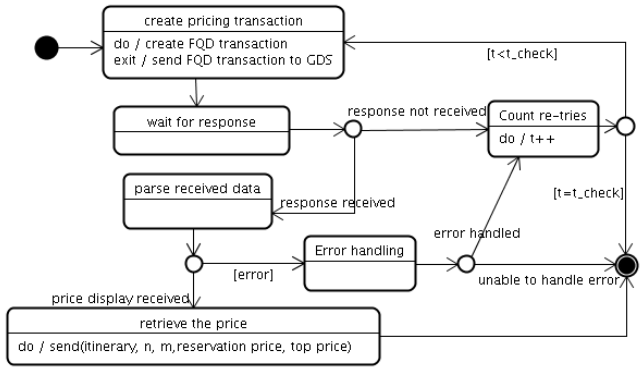


Figure 4: *FlightOffer Agent*'s *Price retrieval* sub-state statechart diagram

Let us now consider the question of integrating this system with the Travel Support System (TSS). While there is a number of interesting questions that would have to be addressed, the one that we are concerned with in this paper is as follows. In the TSS all data is stored in the system in a semantically demarcated

fashion. Furthermore, we envision the augmented e-commerce system as comprising a number of e-travel agencies that utilize methods developed there to sell airline tickets for selected routes. In this case we have to deal with the following situation. Data stored in and provided by the *GDS* is not ontologically demarcated. Hotel and restaurant information stored in the travel agency is ontologically demarcated. Therefore, to be able to combine these two systems one has to provide travel agencies in the e-commerce system with: (1) air-travel ontology, that should be integrated with the two already developed ontologies, and (2) way of translating data provided by the GDS into an appropriate form for the travel agency and the GDS to "understand" each-other. In the remaining parts of this paper we address the first issue, while in the concluding remarks we sketch our proposed solution of the second one.

3 General and travel ontologies

As the first step in the direction of being able to utilize an air travel ontology, we have researched the existing available ontologies.

While the largest general ontology building projects, such as (1) the Cyc project [31], (2) WordNet [41], (3) Suggested Upper Merged Ontology (SUMO) [36], and (4) SENSUS [34] do not provide us with an "ontology of travel," there exist a number of smaller scale attempts at defining such an ontology. (1) Mondeca´s [30] tourism ontology defines tourism concepts based on the WTO thesaurus. (2) The Travel Agent Game in Agentcities (TAGA) is an agent framework for simulating the global travel market on the Web. Its purpose is to demonstrate Agentcities and Semantic Web technologies [37]. In addition to the FIPA content language ontology, TAGA defines (a) basic travel concepts such as itineraries, customers, travel services, and service reservations, and (b) different types of auctions, roles participants play in them, and protocols used. (3) Harmonize is an attempt at ontology-mediated integration of tourism systems following different standards [15]. Its goal is to allow organizations to exchange information without changing data structures. The Harmonize project also involves sub-domains that are only partially related to the world of travel: geographical and geo-spatial concepts, means of transportation, political, temporal, activity/interest, gastronomy etc. These sub-domain concepts can be used within the travel system (directly, as needed) or incorporated into the ontology constructed for the system. It is claimed that the next generation of "eTourism" will be powered by the Semantic Web technology (resulting in an eTourism Semantic Web portal which will connect the customers and virtual travel agents from anywhere at anytime). Goes with out saying that this is a very interesting project, however, airline ticket sales are not included in the current version of Harmonize ontology. (4) Finally, a number of "minimalist" travel ontologies can be found within the DAML language portal [11]. For instance, the Itinerary-ont is an ontology for representing travel itineraries. It reuses the airport codes ontology and involves definitions of terms like Aircraft, Class, Flight etc. Another example is the Trip Report Ontology that defines Airfare, Amount, Date, etc., and models on-line sale.

The complete list of pros and cons for ontologies listed above may be found in [40]. There, we report results of our in-depth analysis of the possibility to utilize any of them in airline ticket sales. Overall, none of them had a fully developed air travel part and that could also interface with an actual GDS, and therefore we had to develop our own, based on the Open Travel Alliance messaging system.

4 OTA and OTA Air Messages

The Open Travel Alliance (OTA) is a non-profit organization working to establish a common electronic vocabulary for exchange of travel information. Such an exchange is to take form of standardized eXtensible Markup Language (XML) messages. OTA specifications have been designed to serve: (a) as a common language for travel-related terminology, and (b) as a mechanism for exchange of information between travel industry members [14]. The OTA Air Messages standard, which is of particular interest in our work, specifies structure and elements of different scenarios involved in selling air travel tickets. Let us note that since this is a specification of messaging, it does not cover any other operations involved in selling air-tickets (e.g. airfare calculations). These operations have to be treated separately. OTA messages have been proposed as pairs of request and response messages (RQ / RS below). Let us summarize their main features (their complete description can be found in [32]).

OTA_AirAvailRQ/RS – establishes airline flight availability for a city pair, specific date, specific number and type of passengers. The request can also be narrowed to a specific airline, flight or booking class. Optional requested information can include: time / time window, connecting cities, client preferences (airlines, cabin, flight types etc.). The response message (*RS*) contains flight availability. Furthermore, a set of origin and destination options is returned, each of which contains one or more (connecting) flights that serve that city pair. For each flight information about: origin and destination airports, departure and arrival date/times, booking class availability, equipment, meal information and code-share information is returned.

OTA_AirBookRQ/RS – requests to book a specific itinerary for one or more identified passengers. The message contains optional pricing information, allowing the booking class availability and pricing to be rechecked as part of the booking process. Optional requested information can include: seat and meal requests, Special Service Requests (SSR), Other Service Information (OSI), remarks, fulfillment

information – payment, delivery details, type of ticket desired. If booking is successful, the RS message contains the itinerary (including the directional indicator, status of booking, and number of passengers), passenger and pricing information sent in the request, along with a booking reference number (PNR Locator) and the ticketing information. The RS echoes back received information with additional information – booking reference from the GDS through which reservation was created.

OTA_AirFareDisplayRQ/RS – allows a client to request information on fares, which exist between a city pair for a particular date or date range. No inventory check for available seats on flights is performed by the server before the RS is send back. The request can optionally contain information indicating that a more specific response (e.g. passenger information, specific flight information and information on the types of fares that the client is interested in) is required. The RS message repeats *FareDisplayInfo* elements, each of which contains information on a specific fare contract including airline, travel dates, restrictions and pricing. It can also return information on other types of fares that exist, but have not been included in the response.

OTA_AirFlifoRQ/RS – requests updated information on the operation of a specific flight (it requires the airline, flight number and departure date; the departure and arrival airport locations can be also be included). The RS includes real-time flight departure and arrival information. It also includes: departure airport, arrival airport, marketing and operating airline names; when applicable, flight number, type of equipment, status of current operation, reason for delay or cancellation, airport location for diversion of flight, current departure and arrival date and time, scheduled departure and arrival date and time, duration of flight, flight mileage, baggage claim location.

OTA_AirLowFareSearchRQ/RS – requests priced itinerary options for flights between specific city pairs on certain dates for a specific number and types of passengers. Optional requested information can include: time / time window, connection points, client preferences (airlines, cabin, flight types etc.), flight type (nonstop or direct), number of itinerary options desired. The RS contains a number of *Priced Itinerary* options. Each includes: a set of available flights matching the client's request, pricing information including taxes and full fare breakdown for each passenger type, ticketing information – ticket advisory information and ticketing time limits, fare basis codes and the information necessary to make a rules entry.

OTA_AirPriceRQ/RS – requests pricing information for specific flights on certain dates for a specific number and type of passengers. The message allows for optional information such as fare restriction preferences and negotiated fare contract codes to be included. The pricing request contains information necessary to perform an availability / sell from availability / price series of entries for an airline CRS or GDS. The RS contains a *Priced Itinerary* that includes: set of flights, pricing information including taxes and full fare breakdown for each passenger type, ticketing information, fare basis codes and the information necessary to make a fare rules entry.

OTA_AirRulesRQ/RS – requests text rules for a specific fare basis code for an airline and a city pair for a specific date. Negotiated fare contract codes can be included in the request. The RS contains a set of, human readable, rules, identified by their codes.

OTA_AirSchedulesRQ/RS – provides customer, or a third party, with ability to view flight schedules. It requires specification of the departure and arrival cities and a specific date. It offers flight information on airlines that provide service between requested cities and could be used when customer: (1) wants to determine what airlines offer service to/from specific destinations, (2) is looking for a specific flight number – by entering the arrival and departure cities, and the approximate arrival or departure time, specific flight number can be found, (3) needs to determine the days of the week that service is scheduled to and from requested destinations, (4) wants to determine aircraft type used to fly that route. Message may request other information that customers are interested in: meal service, duration of flight, on-time statistics and if smoking is allowed. In addition, these messages provide foundation for electronic timetables.

OTA_AirSeatMapRQ/RS – displays seats available on a given flight, as well as their location within the aircraft. It is used o make seat assignments as it identifies all information necessary to request and return an available seat map for a particular flight. Types of information for the seat map request include: airline, flight number, date of travel, class of service and frequent flier status. The RS includes: flight, aircraft and seat description information.

OTA_AirBookModifyRQ/OTA_AirBookRS – requests to modify an existing booking file. It contains all elements of the *OTA_AirBookRQ* plus a general type of modification, i.e. name change, split, cancel or other; as indicated with the attribute *ModificationType*. The modification operation on different elements is either indicated with the existing attribute *Status* (for air segments, SSR's and seat requests) or with attribute *Operation* of type *ActionType* for other elements (i.e. other service information, remarks or *AirTraveler* elements). In the *AirBookModifyRQ*, all data to be changed is submitted and in the *AirReservation* element all existing data may be submitted. This allows the receiving system to perform a consistency check before updating the booking file (but to keep the message small, this part can be omitted). Changes to a booking (1) may result in required updates of the ticket (e.g. revalidation), (2) may imply charges for the change, (3) the pricing may change, and/or (4) some fees may need to be collected. Pricing and fulfillment details required to achieve results of

AirBookModify ticketing, are out of scope and are omitted. The RS confirms changes in the itinerary.

5 Proposed ontology

As indicated above, in Section 3 and in our research [33, 34] we have established that existing air-travel ontologies have been designed mostly as "academic" demonstrator systems – rather than with the goal of actually working within the context of real-life airline reservation systems – and this explains lack of important features when it comes to dealing with genuine air travel data. According to our best knowledge, the only project that actually involves airline industry is the OTA specification (which, as stated above, is only a messaging specification). Therefore, we decided to create new ontology that would: (1) utilize International Air Transport Association (IATA) [14-19] mandated data descriptions and recommended practices; (2) utilize as much as possible from existing travel ontologies – as long as they follow IATA practices, (3) match features included in the OTA specification, and (4) be synchronized with our existing travel ontology. To achieve this goal we have applied a bottom-up approach and our initial goal was to model reservations as defined in the AMADEUS global distribution system.

In the proposed ontology we have divided main classes into following groups: *AirTravelCodes, AirTravel, AirInfrastructureCodes and AirInfrastructure. AirInfrastructure* group encloses most basic terms related to air travel industry such as *Airline*, *Airplane* and *Airport*. While all three are defined in line with specifications presented in [14, 15, 19], the latter one (*Airport*) is a subclass of our *OutdoorLocation* class that was designed for the TSS [11]. In this way it is possible for the traveler to obtain more data regarding the airport than the city name, which usually is the only information that can be obtained from other airline travel related ontologies. Specifically, the *TSS* offers *OutdoorLocation* class that includes, among others, such details as geographical, urban location, address details, nearby attractions etc. To illustrate the results, let us present here the n-triples for this class:

```
base:OutdoorLocation a rdfs:Class;
  rdfs:subClassOf geo:SpatialThing;
  rdfs:comment "Outdoor location.
Geographical and urban references.".

base:address a rdf:Property;
  rdfs:comment "Address details.";
  rdfs:domain base:OutdoorLocation;
  rdfs:range adrec:AddressRecord.

base:attractionCategory a rdf:Property;
  rdfs:comment "Nearby attractions.";
  rdfs:domain base:OutdoorLocation;
  rdfs:range base:AttractionCategoryCode.

base:indexPoint a rdf:Property;
  rdfs:comment "Reference map point.";
  rdfs:domain base:OutdoorLocation;
  rdfs:range base:IndexPointCode.

base:indexPointDist a rdf:Property;
  rdfs:comment "Distance from the reference
map point.";
  rdfs:domain base:OutdoorLocation;
  rdfs:range base:IndexPointCode.

base:locationCategory a rdf:Property;
  rdfs:comment "Location category.";
  rdfs:domain base:OutdoorLocation;
  rdfs:range base:LocationCategoryCode.

base:neighbourhood a rdf:Property;
  rdfs:label "Neighbourhood";
  rdfs:comment "The neighborhood of the
Outdoor
               location.";
  rdfs:range xsd:string;
  rdfs:domain base:OutdoorLocation.

base:crossStreet a rdf:Property;
  rdfs:label "Cross street";
 rdfs:comment "The nearest street that
crosses the street that the restaurant is
on.";
  rdfs:range xsd:string;
       rdfs:domain base:OutdoorLocation.

base:AttractionCategoryCode a rdfs:Class;
  rdfs:comment "Possible categories of places
which might be of interest for
visitors/guests and can be found in the
neighborhood." .

base:IndexPointCode a rdfs:Class;
  rdfs:comment "Possible reference map
points.".

base:LocationCategoryCode a rdfs:Class;
  rdfs:comment "Possible location
categories.".
```

As our system needs recognition of IATA codes to fulfill its aim, we have added three-letter IATA airport code as a property of our class. These codes are represented with a separate class *AirportCode* that was based upon the DAML *AirportCodes* class from the Itinerary-ont ontology, shortly described in Section 3. In this way we were able to offer more complete information about airport and to include information that other ontologies also provide. Following is the N3 notation based depiction of the *Airport* class:

```
base:Airport a rdfs:Class;
 rdfs:subClassOf loc:OutdoorLocation;
 rdfs:comment "Used for airport's city and
geographical location description".

base:airportCode a rdf:Property;
 rdfs:domain base:Airport;
 rdfs:range apc:AirportCode.
```

For the sake of clarity, let us provide the definition and some instances of our *AirportCode* class.

```
base:AirportCode a rdfs:Class;
rdfs:comment "Represents three letter code of
an airport".

#instances of AirportCode class
base:TGD a base:AirportCode.
```

```
base:WAW        a       base:AirportCode.
base:LIS        a       base:AirportCode.
base:MOW        a       base:AirportCode.
```

AirInfrastructureCodes group contains, used in other classes, codes for airports and countries. Included classes are *ISOCountryCode* and *AirportCode*. *AirTravelCodes* group comprises industry codes used in GDSs and CRSs for the itinerary reservation and the ticket issuance: *IATATicketIndicator*, *IATAStatusCode*, *CabinClass*, *BookingClass*, *IATAFareBasis*, *MealCode*, *SSRCode*, *SSRMealCode*, *TicketDestignator* (details can be found in [16-21]). Finally, the *AirTravel* group takes care of upper-level terms that define more complex objects used in the air travel systems. Following classes are included in this group: *OfficeID*, *TerminalID*, *AgentCredentials* – that define credentials of the GDS/CRS user, *AvailabilityDisplay* – that defines available flight options for a certain route, *Flight* – with usual properties together with status statistics, *IATAItinerary* – that defines itinerary for the passenger, *PNR* – Passenger Name Record or, simply described, a reservation with all details of the passenger, the itinerary, special requests and the GDS/CRS locator code, *Pricing* – that describes available prices for a certain route with or without taxes included, *SeatMapPlan* – for a certain flight, *Tariff* - with *Category* properties that are coded as in the *ATPCO*'s (Airline Tariff Publishing Company) recommendation, and *TimetableDisplay* – with timetable of different airlines for a certain route.

As stated above some classes were inherited or used as upper level classes from the TSS. These classes were: *OutdoorLocation*, *IATADiscountCodes**, *MeanOfPayment*, *FareTax*, *Discounts*, *DiscountCodes*, *IATATaxCodes**, *NameRecord*, and *PersonTitle*. Marked with * are classes that were sub classed from classes inherited from the TSS.

One additional, very important, concept in traveling is currency. At first we designed a very simple class that contained only the currency code. Promptly this showed to be insufficient as air travel currency application involved some complicated restrictions. As in the case of air travel ontology, we made an effort to find an already existing ontology of currency, and inject it into our project. We studied several currency ontologies (more details can be found in [35]) and found out that ontology used in Cambia web-service [10] was the most appropriate one. Unfortunately, it was rather broad, and furthermore we had to modify it so that it could be used for currency conversion guided by the IATA conversion rules [21].

OTA message	OTA message element	Related properties of Tariff class from our ontology
OTA_AirFareDisplayRS	FareDisplayInfo ***attributes*:**	Tariff **class properties:**
	• *FareApplicationType*	• *FareAplicationType*
	• *ResBookDesigCode*	• *bk* **range** BookingClass
	• *MilageIndicator*	• *milageInd*
	• *FareStatus*	• *fareStatus*
	FareReference	*farebasis* **range** IATAFareBasis **class**
	RuleInfo ***subelements***	Tariff **class properties** (rules)**:**
	• *MinimumStay*	• *_06* **range** StayLength **class**
	• *MaximumStay*	• *_07* **range** StayLength **class**
	FilingAirline	*carrier* **range** Airline **class**
	DepartureLocation ***attribute*** *LocationCode*	*origin* **range** Airport **class**
	ArrivalLocation ***attribute*** *LocationCode*	*destination* **range** Airport **class**
	Restriction ***attributes***	Tariff **class properties**
	• *GlobalIndicatorCode*	• *globaldirection*
	• *MaximumPermittedMilage*	• *mpm*
	PricingInfo ***attributes***	Tariff **class properties**
	• *NegotiatedFare*	• *_35*
	• *PassengerTypeCode*	• *paxtype*
	• *TicketingDestignatorCode*	• *bk*
	BaseFare ***attributes***	Tariff **class properties**
	• *Amount*	• *ow*, *rt*
	• *CurrencyCode*	• *currency* **range** Currency **class**
	• *DecimalPlaces*	***Defined under Currency class***

Table 1: *Matching the OTA message with the air-travel ontology.*

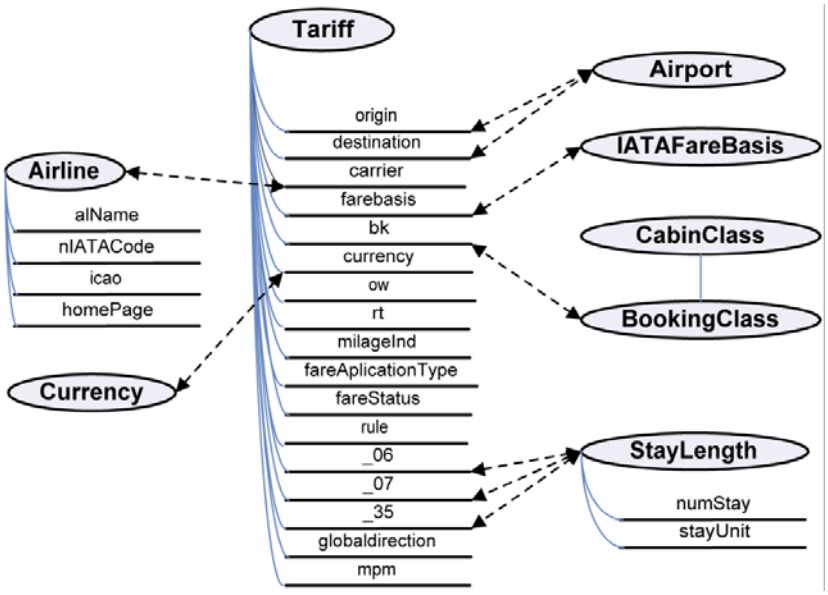


Figure 5: Protégé display of *Tariff* class.

Let us stress that since the OTA was defined as a messaging system used for information exchange, while the proposed ontology was created with intention to describe persistent data in our system, therefore quite often more then one class from our ontology has to be used in association with a single OTA message. As request (RQ) messages contains only data used to make a query, let us illustrate how the RS message matches with the proposed ontology in the case of the *OTA_FareDisplaylRS*. In our ontology an equivalent class is *Tariff*. In Table 1 we depict how elements of the message match elements of our ontology. Furthermore, Figure 5 shows relations of the *Tariff* class with other classes (*Airline, Airport, IATAFareBasis, StayLength, BookingClass*) from our ontology.

Finally, one of the major advantages of utilizing the ontology technologies to demarcate electronic data is that it provides us with a highly readable, customizable and scalable knowledge (data) model. This allows us, among others, to swiftly browse the travel related content, based on the ontology concept references. Figure 6 presents such references between Hotel, Airport, Restaurant, OutdoorLocation, Currency and Person concepts. Obviously, the TSS ontology and its air-ticketing-dedicated extension contain far larger number of inter-concept references; however, presenting them within a single figure would greatly limit its readability.

6 Concluding remarks

In this paper we have summarized results obtained thus far in our attempt in developing an agent-based airline ticket selling system. We have started from presenting an augmented version of a model agent-based e-commerce system and followed with a suggestion that such a system should be merged with an agent-based Travel Support System that we are also developing. To achieve this goal it was necessary to develop ontology of air travel. Based on our analysis of existing travel ontologies we have decided to develop our own ontology that is based on IATA manuals and OTA messaging system. As a result, in this paper we have illustrated how an ontology can be extracted from OTA messages. Overall, when completed (currently, the proposed merged travel ontology it is available for comments at: http://agentlab.swps.edu.pl) our (air) travel ontology should be capable of being used to interface our Travel Support System with an actual GDS (which is one of important goals of our project).

Let us note that there exist already GDS's that allow communication using OTA messaging. Leading this development, AMADEUS in its newly created platform called 'Results CMS' aimed at lowering cost of operations and offered OTA messaging as a way to distribute airline inventory to external travel sites and dynamic package providers. Therefore, as the next step of our research, we plan to develop two parsers. Let us assume that a query that is related to air-travel has been formulated in our system. Obviously, this will be a SPARQL query (as SPARQL is our language of choice to query ontologically demarcated content stored in Jena repository). This query will then be translated into an OTA message and submitted to the GDS. Such a translation will be based on our air-travel ontology. As a response, the GDS will send an OTA response message, containing requested information. This message will be then parsed and information translated into instances of our air-travel ontology. We will then use our display system [25] to present them to the user. We will report on our progress in subsequent publications.

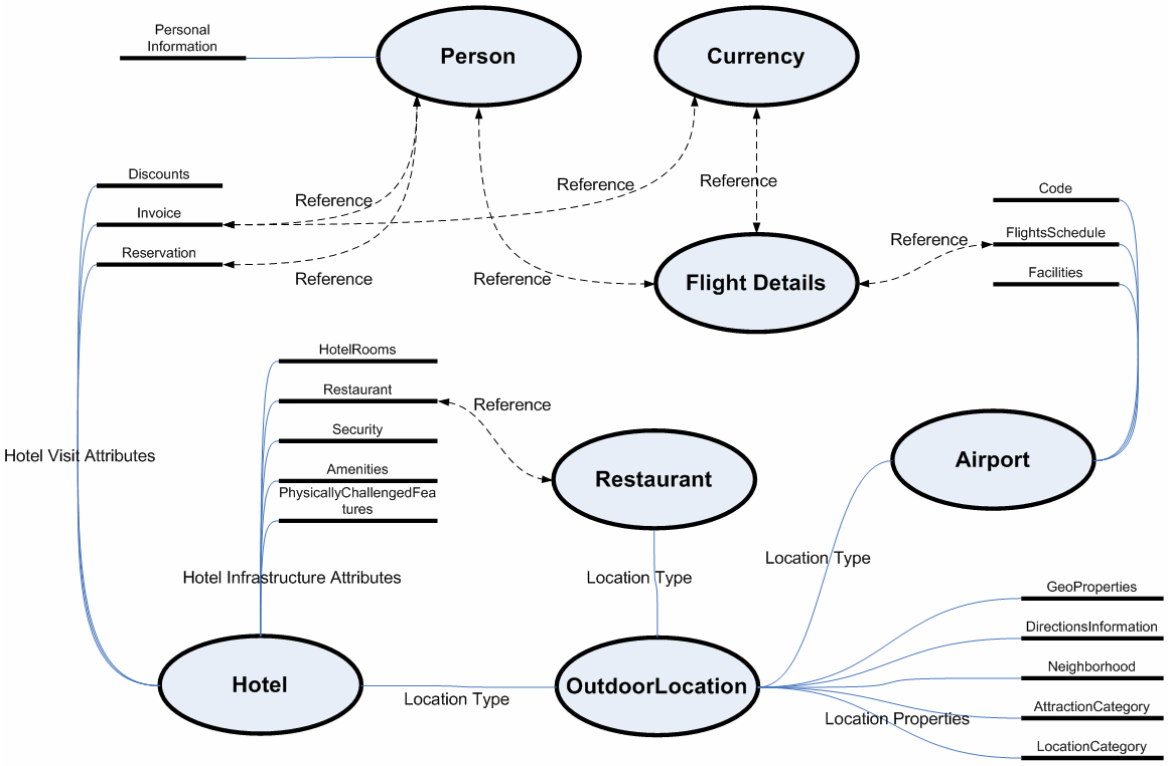


Figure 6: Ontology concept references

Acknowledgement

We want to thank Mr Zoran Djurišić, the President of Board of Directors of Montenegro Airlines for his support of this research. Work of Maria Ganzha, Maciej Gawinecki and Marcin Paprzycki was partially sponsored by the EU IRG grant – project E-CAP.

References

[1] AMADEUS, http://www.amadeus.com/

[2] Bădică, C., Badita, A., Ganzha, M., Iordache, A., Paprzycki M.: Implementing Rule-based Mechanisms for Agent-based Price Negotiations. In: Proceedings of the SAC'2005 Conference (in press)

[3] Bădică, C., Ganzha, M., Paprzycki, M., Pîrv˘anescu, A.: Combining Rule-Based and Plug-in Components in Agents for Flexible Dynamic Negotiations. In: M. P˘echou˘cek, P. Petta, and L.Z. Varga (Eds.): Proceedings of CEEMAS'05, Budapest, Hungary. LNAI 3690, Springer-Verlag, pp.555-558, 2005.

[4] Bădică, C., Ganzha, M., Paprzycki, M., Pîrvănescu, A.: Experimenting With a Multi-Agent E-Commerce Environment. In: V. Malyshkin (Ed.): Proceedings of PaCT'2005, Krasnoyarsk, Russia. LNCS 3606, Springer-Verlag, pp.393-402, 2005.

[5] Bădică, C., Bădită, A., Ganzha, M., Paprzycki, M., Developing a Model Agent-based E-commerce System, in: Jie Lu et. al. (eds.) E-Service Intelligence - Methodologies, Technologies and Applications (to appear)

[6] Bădică C., Ganzha M., Paprzycki M., UML Models of Agents in a Multi-Agent Ecommerce System. In: Proceedings of the ICEBE 2005 Conference, IEEE Press, Los Alamitos, CA, 56-61

[7] Costin Bădică, Maria Ganzha, Maciej Gawinecki, Pawel Kobzdej, Marcin Paprzycki (2006) Towards Trust Management in an Agent-based E-commerce System - Initial Considerations. In: A. Zgrzywa (ed.) Proceedings of the MISSI 2006 Conference, Wroclaw University of Technlogy Press, Wroclaw, Poland, 225-236

[8] Bartolini, C., Preist, C., Jennings, N.R.: Architecting for Reuse: A Software Framework forAutomated Negotiation. In: Proceedings of AOSE'2002: Int. Workshop on Agent-Oriented Software Engineering, Bologna, Italy, LNCS 2585, Springer Verlag, pp.88-100, 2002.

[9] Bartolini, C., Preist, C., Jennings, N.R.: A Software Framework for Automated Negotiation.In: Proceedings of SELMAS'2004. LNCS 3390, Springer-Verlag, pp.213-235, 2005.

[10] Cambia Service, http://zurich.agentcities.whitestein.ch/Services/Cambia.html

[11] DAML Ontologies, http://www.daml.org

[12] Ganzha, M., Paprzycki, M., Pîrvănescu, A., Bădică, C, Abraham, A.: JADE-based Multi-Agent E-commerce Environment: Initial Implementation, In: Analele Universităţii dinTimişoara, Seria Matematică-Informatică, 2005 (to appear).

[13] Gordon M., Paprzycki M., Designing Agent Based Travel Support System. In: Proceedings of the ISPDC 2005 conference, IEEE Computer Society Press, Los Alamitos, CA, 2005, 207-214

[14] http://www.opentravel.org/about.cfm

[15] Harmonize, http://deri.at/research/projects/e-tourism

[16] IATA Airline Coding Directory – Airline Designators, 70th Edition
[17] IATA City Code Directory, 43rd Edition, Effective 9 December 2005 – 31 December 2006
[18] IATA Passenger Services Conference Resolutions Manual, 24th Edition, Effective 1 June 2005 – 31 May 2006
[19] IATA Passenger Tariff Coordination Conferences Manual, Composite, Dec 9, 2005 until Dec 31, 2006
[20] IATA Reservation Service Manual, 23rd Edition
[21] IATA Standard Schedules Information Manual, Mar 1, 2006 until Sep 30, 2006
[22] Jena 2 Ontology API – General concepts, http://jena.sourceforge.net/ontology/index.html#generalConcepts
[23] Jena Documentation, http://jena.sourceforge.net/documentation.html
[24] Kowalczyk, R., Ulieru, M., Unland, R.: Integrating Mobile and Intelligent Agents in Advanced E-commerce: A Survey. In: Agent Technologies, Infrastructures, Tools, and Applications for E-Services, Proceedings NODe'2002 Agent-Related Workshops, Erfurt, Germany. LNAI 2592, Springer-Verlag, pp.295-313, 2002.
[25] Maciej Gawinecki, Minor Gordon, Paweł Kaczmarek, Marcin Paprzycki (2003) The Problem of Agent-Client Communication on the Internet. Scalable Computing: Practice and Experience, 6(1), 2005, 111-123
[26] Maes P., Agents that Reduce Work and Information Overload. Communications of the ACM, 37, 7, 1994, 31-40
[27] Maes, P., Guttman, R.H.,Moukas, A.G.: Agents that Buy and Sell: Transforming Commerce as we Know It. In Communications of the ACM, Vol.42, No.3, pp.81-91, 1999.
[28] Maria Ganzha, Maciej Gawinecki, Pawel Kobzdej, Marcin Paprzycki, Costin Bădică (2006) Functionalizing trust in a model agent based e-commerce system. In: M. Bohanec et. al. (eds.), Proceedings of the 2006 Information Society Multiconference, Josef Stefan Institute Press, 22-26]
[29] Mladenka Vukmirović, Marcin Paprzycki, Michał Szymczak (2006) Designing ontology for the Open Travel Alliance Airline Messaging Specification, In: M. Bohanec et. al. (eds.), Proceedings of the 2006 Information Society Multiconference, Josef Stefan Institute Press, 101-105]
[30] Mondeca, http://www.mondeca.com
[31] OpenCyc, http://www.opencyc.org
[32] OpenTravelTM Alliance, Message Users Guide. 2005B Version 1.0, 2 December 2005
[33] SABRE, http://www.sabre.com/
[34] SENSUS, http://www.isi.edu/natural-language/projects/ONTOLOGIES.html
[35] Szymczak M., Gawinecki M., Vukmirović M., Paprzycki M., Ontological reusability in state-of-the-art semantic languages, Proceedings of the XVIII Summer School of PIPS (to appear)
[36] SUMO, http://www.ontologyportal.org
[37] TAGA, http://www.agentcities.org
[38] Trastour, D., Bartolini, C., Preist, C.: Semantic Web Support for the Business-to-Business E-Commerce Lifecycle. In: Proceedings of the WWW'02: International World Wide Web Conference, Hawaii, USA. ACM Press, New York, USA, pp.89-98, 2002.
[39] Vukmirović M., Ganzha M., Paprzycki M.: Developing a Model Agent-based Airline Ticket Auctioning System. In: Proceedings for the IIPWM Conference, LNAI
[40] Vukmirović M., Szymczak M., Ganzha M., Paprzycki M.: Utilizing Ontologies in an Agent-based Airline Ticket Auctioning System. In: Proceedings of the 28th ITI Conference, IEEE Computer Society Press, Cavtat, Dubrovnik, Croatia, 385-390
[41] WordNet, http://www.daml.org/ontologies/196
[42] Wooldridge, M.: *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.

Discriminatory Algorithmic Mechanism Design Based WWW Content Replication

Samee Ullah Khan and Ishfaq Ahmad
Department of Computer Science and Engineering
University of Texas
Arlington, TX-76019, USA
E-mail: {sakhan, iahmad}@cse.uta.edu

Keywords: data replication, resource allocation, game theory, algorithmic mechanism design, static allocation

Received: November 8, 2005

Replicating data over geographically dispersed web servers reduces network traffic, server load, and more importantly the user-perceived access delays. This paper proposes a unique replica placement technique using the concepts of a supergame. The supergame allows the agents who represent the data objects to continuously compete for the limited available server memory space, so as to acquire the rights to place data objects at the servers. At any given instance in time, the supergame is represented by a game which is a collection of subgames, played concurrently at each server in the system. We derive a resource allocation mechanism which acts as a platform at the subgame level for the agents to compete. This approach allows us to transparently monitor the actions of the agents, who in a non-cooperative environment strategically place the data objects to reduce the user access time, latency, which in turn adds reliability and fault-tolerance to the system. We show that this mechanism exhibits Nash equilibrium at the subgame level which in turn conforms to games and supergame Nash equilibrium, respectively, guaranteeing the entire system to be in a continuous self-evolving and self-repairing mode. The mechanism is extensively evaluated against some well-known algorithms, such as: greedy, branch and bound, game theoretical auctions and genetic algorithms. The experimental results reveal that the mechanism provides excellent solution quality, while maintaining fast execution time.

Povzetek: Opisana je metoda za multipliciranje internetnih strani.

1 Introduction

Web replication aims to reduce network traffic, server load, and user-perceived delay by replicating popular content on geographically distributed web servers (sites). Specifically, a replica placement algorithm aims to strategically select replicas (or hosting services) among a set of potential sites such that some objective function is optimized under a given traffic pattern.

One might argue that the ever decreasing price of memory renders the optimization or fine tuning of replica placement a "moot point". Such a conclusion is ill-guided for the following two reasons. First, studies ([4], [8], etc.) have shown that users' access hit ratio grows in *log*-like fashion as a function of the server memory size. Second, the growth rate of Web content is much higher than the rate with which memory sizes for the servers are likely to grow. The only way to bridge this widening gap is through efficient replica placement and management algorithms.

The decision where to place the replicated data must trade off the cost of accessing the data, which is reduced by additional copies, against the cost of storing and updating the additional copies. Discussions in [20], [22], [25], [26], [30], etc. reveal that client(s) experience reduced access latencies provided that data is replicated within their close proximity. However, this is applicable in cases when only read accesses are considered. If updates of the contents are also under focus, then the locations of the replicas have to be: 1) in close proximity to the client(s), and 2) in close proximity to the primary (assuming a broadcast update model) copy. Therefore, efficient and effective replication schemas strongly depend on how many replicas to be placed in the system, and more importantly where.

The Internet can be considered as a large-scale distributed computing system. We abstract this distributed computing system as an agent-based model, where each agent is responsible for (or represents) a data object. Each agent competes in a non-cooperative environment for the limited available storage space at each server so as to acquire the rights to place the data object which they represent. Motivated by their self interests and the fact that the agents do not have a global view of the distributed system, they concentrate on local optimization. In such systems there is no a-priori motivation for cooperation and the agents may manipulate the outcome of the replica placement algorithm (resource allocation mechanism or simply a *mechanism*) in their interests by misreporting critical data such as objects' popularity. To cope with these *selfish*

agents, new mechanisms are to be conceived. The goal of a mechanism should be to force the agents not to misreport and always follow the rules.

This paper uses the concepts of game theory to formally specify a mechanism with selfish agents. Game theory assumes that the participating agents have *rational* thoughts that enable them to express their preferences over the set of the possible outcomes of the mechanism. In a mechanism, each agent's benefit or loss is quantified by a function called *valuation*. This function is private information for each agent and is very much possible that if the agents act selfishly, they can misreport their valuations. The mechanism asks the agents to report their valuations, and then it chooses an outcome that maximizes/minimizes a given objective function. Of course the grand problem is to stop the agents from misreporting.

In this paper, we will apply the derived mechanism to the fine grained data replication problem (DRP) over the Internet. In essence we sculpt the DRP as a *supergame* that is played infinitely during the entire lifespan of the system. In a discrete time instance t, the supergame is represented by a *game*, which is the collection of independent *subgames* that are played concurrently at each site of the distributed system. It is in these subgames that the actual mechanism can be seen to operate.

The major results of this paper are as follows:

1. We derive a general-purpose distributed mechanism that allows selfish agents to compete at each site in the distributed computing system for the rights to replicate objects in a non-cooperative environment.
2. We show that the concurrently played subgames exhibit Nash equilibrium which in turn guarantees Nash equilibrium for the games and the supergame.
3. The mechanism is compared against some well-known techniques, such as: greedy, branch and bound, genetic and game theoretical auctions, employing various internet topology generators and real user access data. The experimental results reveal that the mechanism provides excellent solution quality, while maintaining fast execution time.

This paper is organized as follows. Section 2 formulates the DRP. Section 3 describes the mechanism. The experimental results, related work and concluding remarks are provided in Sections 4, 5 and 6, respectively.

2 Formal Description of the Data Replication Problem

The most frequently used acronyms are recorded in Table 1.

Consider a distributed system comprising M sites, with each site having its own processing power, memory (primary storage) and media (secondary storage). Let S^i and s^i be the name and the total storage capacity (in simple data units e.g. blocks), respectively, of site i where $1 \leq i \leq M$. The M sites of the system are connected by a communication network. A link between two sites S^i and S^j (if it exists) has a positive integer $c(i,j)$ associated with it, giving the communication cost for transferring a data unit between sites S^i and S^j. If the two sites are not directly connected by a communication link then the above cost is given by the sum of the costs of all the links in a chosen path from site S^i to the site S^j. Without the loss of generality we assume that $c(i,j) = c(j,i)$. This is a common assumption (e.g. see [20], [22], [26], [30], etc.). Let there be N objects, each identifiable by a unique name O_k and size in simple data unites o_k where $1 \leq k \leq N$. Let r_k^i and w_k^i be the total number of reads and writes, respectively, initiated from S^i for O_k during a certain time period t. This time period t determines when to initiate a replica placement algorithm (in our case the mechanism). Note that this time period t is the only parameter that requires human intervention. However, in this paper we use analytical data that enables us to effectively predict the time interval t (see Section 3.4. for details).

Symbols	Meaning
M	Total number of sites in the network.
N	Total number of objects to be replicated.
O_k	k-th object.
o_k	Size of object k.
S^i	i-th site.
s^i	Size of site i.
r_k^i	Number of reads for object k from site i.
R_k^i	Aggregate read cost of r_k^i.
w_k^i	Number of writes for object k from site i.
W_k^i	Aggregate write cost of w_k^i.
NN_k^i	Nearest neighbor of site i holding object k.
$c(i,j)$	Communication cost between sites i and j.
P_k	Primary site of the k-th object.
R_k	Replication schema of object k.
$C_{overall}$	Total overall data transfer cost.
LS	A list of sites that can replicate an object.
L^i	A list of objects that can be replicated onto site S^i.
B_k^i	Benefit of replicating object k onto site S^i.
B_k	Temporary variable to store object valuations.
b^i	Available space at site S^i.
v	Valuation of an agent for an object.
SGRG	Self Generate Random Graphs.
GT-ITM	Georgia Tech Internetwork Topology Models.
GT-ITM PR	GT-ITM Pure Random.
GT-ITM W	GT-ITM Waxman.
SGFCG	Self Generated Fully Connected Graphs.
SGFCGUD	SGFCG Uniform Distribution.
SGFCGRD	SGFCG Random Distribution.
SGRG	Self Generated Random Graphs.
SGRGLND	SGRG Lognormal Distribution.
DRP	Data replication problem.
OTC	Object transfer cost (network communication cost).

Table 1: Notations and their meanings.

Our replication policy assumes the existence of one primary copy for each object in the network. Let P_k, be the site which holds the primary copy of O_k, *i.e.*, the only copy in the network that cannot be de-allocated, hence referred to as primary site of the k-th object. Each primary site P_k, contains information about the whole replication scheme R_k of O_k. This can be done by maintaining a list of the sites where the k-th object is replicated at, called from now on the *replicators* of O_k. Moreover, every site S^i stores a two-field record for each object. The first field is its primary site P_k and the second

the nearest neighborhood site NN_k^i of site S^i which holds a replica of object k. In other words, NN_k^i is the site for which the reads from S^i for O_k, if served there, would incur the minimum possible communication cost, *i.e.*, $NN_k^i = \{Site\ j \mid j \in R_k \wedge \min c(i,j)\}$. It is possible that $NN_k^i = S^i$, if S^i is a *replicator* or the primary site of O_k. Another possibility is that $NN_k^i = P_k$, if the primary site is the closest one holding a replica of O_k. When a site S^i reads an object, it does so by addressing the request to the corresponding NN_k^i. For the updates we assume that every site can update every object. Updates of an object O_k are performed by sending the updated version to its primary site P_k, which afterwards broadcasts it to every site in its replication scheme R_k.

For the DRP under consideration, we are interested in minimizing the total network transfer cost due to object movement, *i.e.* the Object Transfer Cost (OTC). The communication cost of the control messages has minor impact to the overall performance of the system, therefore, we do not consider it in the transfer cost model, but it is to be noted that incorporation of such a cost would be a trivial exercise. There are two components affecting OTC. The first component of OTC is due to the read requests. Let R_k^i denote the total OTC, due to S^is' reading requests for object O_k, addressed to the nearest site NN_k^i. This cost is given by the following equation:

$$R_k^i = r_k^i o_k c\left(i, NN_k^i\right). \quad (1)$$

The second component of OTC is the cost arising due to the writes. Let W_k^i be the total OTC, due to S^is' writing requests for object O_k, addressed to the primary site P^k. This cost is given by the following equation:

$$W_k^i = w_k^i o_k \left(c\left(i, P_k\right) + \sum_{\forall j \in R_k, j \neq i} c\left(NN_k^i, j\right) \right). \quad (2)$$

Here, we made the indirect assumption that in order to perform a write we need to ship the whole updated version of the object. This of course is not always the case, as we can move only the updated parts of it (modeling such policies can also be done using our framework). The cumulative OTC, denoted as $C_{overall}$, due to reads and writes is given by:

$$C_{overall} = \sum_{i=1}^{M} \sum_{k=1}^{N} \left(R_k^i + W_k^i \right). \quad (3)$$

Let X_{ik}=1 if S^i holds a replica of object O_k, and 0 otherwise. X_{ik}s define an $M \times N$ replication matrix, named X, with boolean elements. Equation 3 is now refined to:

$$X = \sum_{i=1}^{M} \sum_{k=1}^{N} \left[\begin{array}{l} \left(1 - X_{ik}\right) \left[\begin{array}{l} r_k^i o_k \min\left\{c(i,j) \mid X_{jk} = 1\right\} \\ + w_k^i o_k c\left(i, P_k\right) \end{array} \right] \\ + X_{ik} \left(\sum_{x=1}^{M} w_k^x \right) o_k c\left(i, P_k\right) \end{array} \right]. \quad (4)$$

Sites which are not the *replicators* of object O_k create OTC equal to the communication cost of their reads from the nearest *replicator*, plus that of sending their writes to the primary site of O_k . Sites belonging to the replication scheme of O_k, are associated with the cost of sending/receiving all the updated versions of it. Using the above formulation, the DRP can be defined as:

Find the assignment of 0, 1 values in the X matrix that minimizes $C_{overall}$, subject to the storage capacity constraint:

$$\sum_{k=1}^{N} X_{ik} o_k \leq s^i \ \forall \left(1 \leq i \leq M\right),$$

and subject to the primary copies policy:

$$X_{p_k k} = 1 \ \forall \left(1 \leq k \leq N\right).$$

The minimization of $C_{overall}$ will have two impacts on the distributed system under consideration: First, it ensures that the object replication is done in such a way that it minimizes the maximum distance between the replicas and their respective primary objects. Second, it ensures that the maximum distance between an object k and the user(s) accessing that object is also minimized. Thus, the solution aims for reducing the overall OTC of the system. In the generalized case, the DRP has been proven to be NP-complete [26].

3 The Mechanism

In game theory, usually mechanisms refer to auctions. Mechanisms are used to make allocation and pricing decisions in a competitive environment where all involved parties act strategically in their own best interests. In recent years, many areas of mathematical sciences research started to focus on strategic behavior and, consequently, we are witnessing the use of mechanisms in areas where pure optimization techniques were dominant in the past. For example, in the context of distributed systems, such mechanisms have been applied to the scheduling problems [18], [29], etc.

One has to be careful when incorporating a "one-size-fits-all" mechanism model as a piece of solution to a problem. Most of the mechanisms were developed and analyzed in microeconomic theory abstraction. Thus, assumptions underlying desirable properties of some mechanisms could be oversimplifying or even contradictory to the assumptions underlying a problem that plans to incorporate such mechanisms in its solution.

3.1 Discriminatory Mechanism

In this paper we limit our analysis to one-shot (single round) mechanisms in which every agent demands a specific entity. Under our DRP formulation we aim to identify a replica schema that effectively minimizes the OTC. We propose a one-shot discriminatory mechanism, where the agents compete for memory space at sites so that they can acquire the rights to place replicas. The mechanism described in this paper is called discriminatory because not all winning agents pay the same amount. In essence it works as follows: In a discriminatory mechanism, sealed-bids are sorted from high to low, and rights to the available memory space are awarded at the current highest bid price until the (memory) supply is exhausted. The most important point to remember is that the winning agents can (and usually do) pay different prices.

It is to be noted that in a discriminatory mechanism, an agent always bids below its valuation for the entity [16]. If the agent bids at or above its value, then its

payment equals or exceeds its value if it wins, and therefore its expected profit will be zero or negative. Since bids are below the agents' value, the discriminatory mechanism is not a demand reveling mechanism [27].

In a discriminatory mechanism, there is no sequential interaction among agents [27]. Therefore, the mechanism environment is non-cooperative in nature. Agents submit the bids only once. Agents are trading between bidding high and winning for certain and bidding low and benefiting more if the bid wins. In [12] the authors have shown that the discriminatory mechanism is a generalization of the first price sealed-bid auction which is strategically equivalent to the Dutch auction. Unlike in the second price sealed-bid and the English auctions, it is not a dominant strategy for a bidder in the first price sealed-bid auction to bid its valuation for the entity. However, the theoretically optimal bidding strategy in both the first price sealed-bid and the Dutch auctions is the same for any given bidder. Since discriminatory auctions are generalization of the first price sealed-bid auctions, the same argument (about the dominating strategies) holds [17]. Therefore, we are confined to a probabilistic analysis of the discriminatory mechanism.

3.2 Preliminaries

Definition 1 (Supergame): *Generally a game in which some simple game is played more than once (often infinitely many times); the simple game is called the "stage" game or the "constituent" game – a game repeated infinitely is called a supergame. If Γ represents a game then $\Gamma(\infty)$ represents a supergame.*

Definition 2 (Stage game (subgame)): *Frequently it is the case that a game naturally decomposes into smaller games. This is formalized by the notion of stage game (more popularly known as subgames).*

Remarks – We explain this concept using decision trees [27]. Let x be a node which belongs to the set of all the nodes, X, in a tree, K, and let K_x be the subtree of K rising at x. If it is the case that ever information set of Γ either is completely contained in K_x or is disjoint from K_x, then the restriction of Γ to K_x constitutes a game of its own, to be called subgame Γ_x starting at x. This decomposition also affects strategies. Let b represent the strategy set for any player i, then the strategy combination b decomposes into a pair (b_{-x}, b_x) where b_x is a strategy combination in Γ_x and b_{-x} is a strategy combination for the remaining part of the game (the truncated game). If it is known that b_x will be played in Γ_x, then, in order to analyze Γ it suffices to analyze the truncated game $\Gamma_{-x}(b_x)$ which results from Γ.

Interestingly, the concept connecting supergame, games, and subgames is the Nash equilibrium.

Definition 3 (Nash equilibrium): *If there is a set of strategies with the property that no player can benefit by changing her strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute the Nash equilibrium.*

Definition 4 (Equilibrium path): *For a given (Nash) equilibrium an information set is on the equilibrium path if it will be reached with positive probability when the game is played according to the equilibrium strategies.*

Lemma 1 ([17])**:** *Nash equilibrium only depends upon subgame strategy profiles played along the equilibrium path.* ■

Theorem 1 ([16]): *In Nash equilibrium each player's repeated game (supergame) strategy need only be optimal along the equilibrium path.* ■

Remarks – In essence Definitions 3 and 4 and Lemma 1 propose that if a game Γ is in Nash equilibrium, it is only so because all subgames Γ_x are in Nash equilibrium. Extending the same concept, Theorem 1 asserts that Nash equilibrium can be reached in a supergame via the equilibrium path followed by games. Recall that a supergame is an infinite play of games. In summary, if all the subgames are in Nash equilibrium, the corresponding game that encapsulates the subgames is also in Nash equilibrium and so is the supergame which is the collection of infinite number of games played over time.

3.3 Mechanism Applied to the DRP

Form the discussion above, we choose the following line of action.

[1] Define the DRP as a supergame.
[2] Define an instance of the supergame as a game.
[3] Split the game into concurrently played subgames. Each identical to each other in terms of:
   a. *Form:* A discriminatory mechanism.
   b. *Valuation:* Obtainable via the system parameters.
   c. *Information:* Independent of any other subgame.
2. Establish the fact that subgames conform to Nash equilibrium provided agents play optimally.
3. Use Lemma 1 to establish that the entire game at instance t is in Nash equilibrium.
4. Use Theorem 1 to establish that the entire supergame is in Nash equilibrium.

1. Supergame: A supergame $\Gamma(\infty)$ is defined as a mechanism that is played infinitely during the lifespan of the distributed system under consideration. The supergame allows the agents to compete for memory spaces of the sites. The purpose of a supergame is to keep the system in a self evolving and self repairing mode.

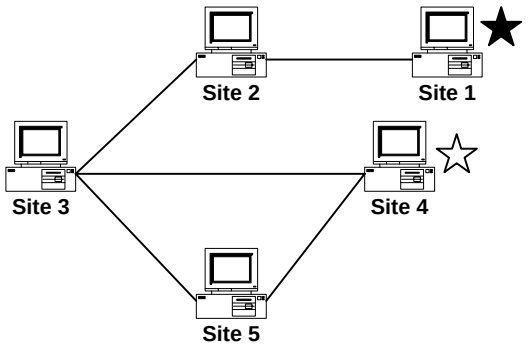


Figure 1(a): The network architecture.

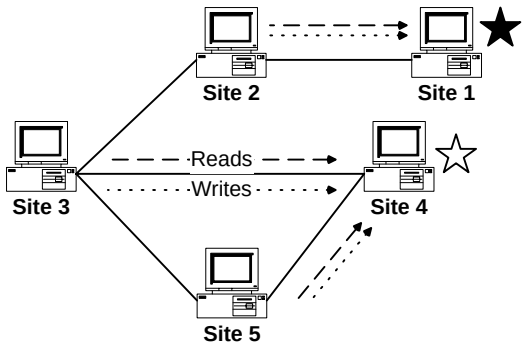


Figure 1(b): Read and write patterns.

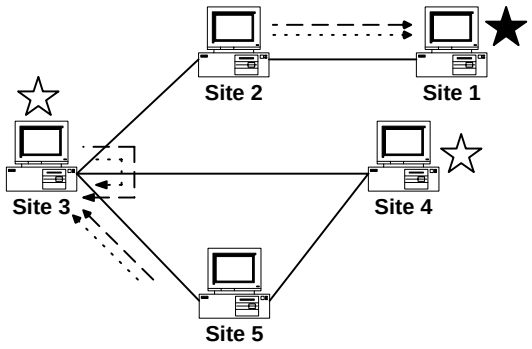


Figure 1(c): Benefits of replication (reads).

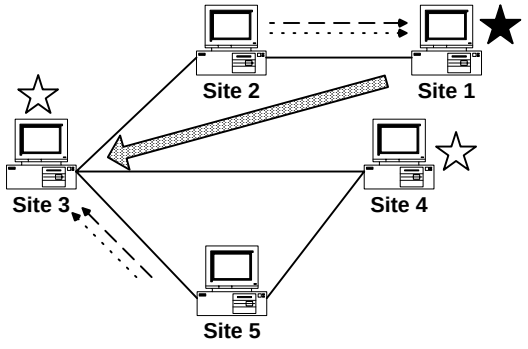


Figure 1(d): Benefits of replication (writes).

2. Game: At any given instance t (t is the instance when a game is invoked, in Section 3.4. we explain what t really means), a game Γ is played. It is to be noted that the sole purpose of defining a game is to observe the solution quality of the replica placements at a given instance t [26].

3. Subgames: A game is split into M concurrently played subgames. Each of these subgames take place at a particular site i. Each agent k competes through bidding for memory at a site i.

3.a. Form: Each site i has a finite amount of space s^i, and available space b^i. It is for this available space b^i that the agents compete. In one-shot all the participating agents submit their bids for the available space. All the bids are sorted in descending order and the first n agents are awarded the rights to place their objects onto site i. Recall that each agent represents an object of size o_k. Therefore, the decision of the first n agents solely depends upon $\sum_{k=1}^{n} o_k \leq b^i, n \leq N$. After the decision is made, the first n agents pay their respective bids. This is discriminatory for the following two reasons. First, all the successful agents pay a different amount for their rights to place an object. Second, the payment is in no relation to the size of the object or the available space at site i. The only connection that the payments have is the benefit that the object brings if replicated to that site. This benefit is the valuation of an agent for its object k if replicated at site i. We describe this valuation below.

3.b. Valuation: Each agent k's policy is to place a replica at a site i, so that it maximizes its (object's) benefit function. This benefit is equivalent to the savings that the object k brings in the total OTC if the object k is replicated at site i. This benefit is given as:

$$B_k^i = R_k^i - \left(\sum_{x=1}^{M} w_k^x o_k c\left(i, P_k\right) - W_k^i \right). \qquad (5)$$

We illustrate the notion of benefit associated with an object k if it is replicated at site i. Figure 1(a) depicts the network with four sites. Site 1 has the primary object represented by □, while Site 4 has the replica of the same object represented by □. If these are the only copies of object k available in the network, then the read and write requests are always sent to the nearest neighbors, where Site 4 is the nearest neighbor of itself (Figure 1(b)). Now what would be the benefit of replicating object k at Site 3? In Figure 1(c), we see that the reads and writes of Site 3 are entertained locally. Moreover, Site 5 can now redirect its request to its newest nearest neighbor, *i.e.*, Site 3. Therefore, the replication of object k at Site 3 clearly reduces the OTC by $RC_k^i = R_k^i + W_k^i$. However (Figure 1(d)), this will cause the Site 1 (location of primary object) to repeatedly send updates of object k to Site 3. Since the local update is already captured by RC_k^i, the increased aggregate updates are given by:

$$\sum_{x=1}^{M} w_k^x o_k c(i, P_k) .$$

From here onwards, for simplicity, we will denote the benefit B_k^i as v (valuation). It is to be understood that to differentiate the valuations between agents k and j we may denote the valuations as v_k and v_j, respectively.

3.c. Information: It is clear that the subgames can operate independently of each other. There is no critical information that is required and is withheld from a subgame. For instance, 1) the frequency of reads and

writes are obtained locally through the site which hosts the subgame, 2) the information about network architecture is globally available since domains can easily pull such information from the routers using the border gate protocol (BGP) [32], and 3) the locations of the primary sites are also available locally since the agents represent the objects, (*i.e.*, they have to know where they originated from,) etc.

4. Subgame Nash equilibrium: To understand the bidding behavior in a discriminatory mechanism, we shall, for simplicity, assume that the agents are ex-ante symmetric. That is, we shall suppose that for all bidders $k = 1,\ldots, N$, $f_k(v) = f(v)$ for all $v \in [0,1]$, where v is the valuation of an agent k for an object, whereas f translates this valuation into something useful, for instance, when bids are required for an object, f can take the form of a bidding function for a valuation v. Note that we only assume that $v \in [0,1]$ for underlying the groundwork for the probabilistic analysis. In reality the valuations are of the form of $v \geq 0$. Clearly, the main difficulty is in determining how the agents, will bid. But note that a rational agent k would prefer to win the right to replicate at a lower price rather than a higher one, agent k would bid low when the others are bidding low and would want to bid higher when the others bid higher. Of course, agent k does not know the bids that the others submit because of the sealed-bid rule. Yet, agent k's optimal bid will depend on how the others bid. Thus, the agents are in a strategic setting in which the optimal action (bid) of each agent depends on the actions of others.

Let us consider the problem of how to bid from the point of view of agent k. Suppose that agent k's value is v_k. Given this value; agent k must submit a sealed-bid, b_k. Because b_k will in general depend on k's value, let's write $b_k(v_k)$ to denote bidder k's bid when his value is v_k. Now, because agent k must be prepared to submit a bid $b_k(v_k)$ for each of his potential values $v \in [0,1]$, we may view agent k's strategy as a bidding function $b_k:[0,1]\rightarrow\Re_+$, mapping each of his values into a (possibly different nonnegative) bid.

Before we discuss payoffs, it will be helpful to focus our attention on a natural class of bidding strategies. It seems very natural to expect that agents with higher values will place higher bids. So, let's restrict attention to strictly increasing bidding functions. Next, because the agents are ex-ante symmetric, it is also natural to suppose that agents with the same value will submit the same bid. With this in mind, we shall focus on finding a strictly increasing β function, $\hat{b}_k:[0,1]\rightarrow\Re_+$, that is optimal for each agent to employ, given that all other agents employ his bidding function as well. That is, we wish to find Nash equilibrium in strictly increasing bidding functions.

Now, let us suppose that we find Nash equilibrium given by the strictly increasing bidding function $\hat{b}(\cdot)$. By definition it must be payoff-maximizing for an agent, say k, with value v to bid $\hat{b}(v)$ given that the other agents employ the same bidding function $\hat{b}(\cdot)$.

Remarks — We explain why we assume that all other agents employ the same bidding function $\hat{b}(\cdot)$. Imagine that agent k cannot attend the auction and that he sends a friend to bid for him. The friend knows the equilibrium bidding function $\hat{b}(\cdot)$ (since it is a public knowledge), but does not know agent k's value. Now, if agent k's value is v, agent k would like his friend to submit the bid $\hat{b}(v)$ on his behalf. His friend can do this for him once agent k calls him and tells his value. Clearly, agent k has no incentive to lie to his friend about his value. That is, among all the values $r \in [0,1]$ that agent k with value v can report to his friend, his payoff is maximized by reporting his true value, v, to his friend. This is because reporting the value r results in his friend submitting the bid $\hat{b}(r)$ on his behalf. But if agent k were there himself he would submit the bid $\hat{b}(v)$.

Let us calculate agent k's expected payoff from reporting an arbitrary value, r, to his friend when his value is v, given that all other agents employ the bidding function $\hat{b}(\cdot)$. To calculate this expected payoff, it is necessary to notice just two things. First, agent k will win only when the bid submitted for him is highest. That is, when $\hat{b}(r) > \hat{b}(v_j)$ for all agents $j \neq k$. Because $\hat{b}(\cdot)$ is strictly increasing this occurs precisely when r exceeds the values of all N-1 other agents. Let F denote the distribution function associated with f, the probability that this occurs is $(F(r))^{N-1}$ which we will denote $F^{N-1}(r)$. Second, agent k pays only when it wins the right to replicate, and pays its bid, $\hat{b}(r)$. Consequently, agent k's expected payoff from reporting the value r to his friend when his value is v, given that all other bidders employ the bidding function $\hat{b}(\cdot)$, can be written as:

$$u(r,v) = F^{N-1}(r)\left(v - \hat{b}(r)\right). \tag{6}$$

Now, as we have already remarked, because $\hat{b}(\cdot)$ is an equilibrium, agent k's expected payoff-maximizing bid when his value is v must be $\hat{b}(v)$. Consequently, Equation 6 must be maximized when $r = v$, i.e., when agent k reports his true value, v, to his friend. So, we may differentiate the right-hand side with respect to r and set the derivative equal to zero when $r = v$. Differentiating yields:

$$\begin{aligned} &d/dr\left(F^{N-1}(r)\left(v-\hat{b}(r)\right)\right) = \\ &\left(N-1\right)F^{N-2}(r)f(r)\left(v-\hat{b}(r)\right) - F^{N-1}(r)\hat{b}'(r) \end{aligned} \tag{7}$$

Setting this equal to zero when r = v and rearranging yields:

$$\begin{aligned} &\left(N-1\right)F^{N-2}(v)f(v)\hat{b}(v) + F^{N-1}(v)\hat{b}'(v) = \\ &\left(N-1\right)vf(v)F^{N-2}(v) \end{aligned} \tag{8}$$

Looking closely at the left-hand side of Equation 8, we see that is just the derivative of the product $F^{N-1}(v)$ times $\hat{b}(v)$ with respect to v. With this observation, we can rewrite Equation 8 as:

$$d/dv\left(F^{N-1}(v)\hat{b}(v)\right)=(N-1)vf(v)F^{N-2}(v). \quad (9)$$

Now, because Equation 9 must hold for every v, it must be the case that:

$$F^{N-1}(v)b(v)= (N-1)\int_0^v xf(x)F^{N-2}(x)dx+constant. \quad (10)$$

Noting that an agent with value zero must bid zero, we conclude that the constant above must be zero.

Hence, it must be the case that:

$$\hat{b}(v)=\frac{N-1}{F^{N-1}(v)}\int_0^v xf(x)F^{N-2}(x)dx, \quad (11)$$

which can be written as:

$$\hat{b}(v)=\frac{1}{F^{N-1}(v)}\int_0^v xf(x)F^{N-2}(x)dx. \quad (12)$$

There are two things to notice about the bidding function in Equation 12. First, as we has assumed, it is strictly increasing in v. Second, it has been uniquely determined. Now since we assumed that each agent is ex-ante in nature, then $F(v) = v$ and $f(v) = 1$. Consequently, if there are N bidders then each employs the bidding function:

$$\hat{b}(v) = \frac{1}{v^{N-1}}\int_0^v xdx^{N-1} \quad (13)$$

$$=\frac{1}{v^{N-1}}\int_0^v x(N-1)x^{N-2}dx$$

$$=\frac{N-1}{v^{N-1}}\int_0^v x^{N-1}dx$$

$$=\left(\frac{N-1}{v^{N-1}}\right)\left(\frac{1}{N}\right)v^N$$

$$=\left(\frac{N-1}{N}\right)v \quad (14)$$

Hence, in conclusion, we have proven the following:

Theorem 2: *If N agents have independent private values drawn from the common distribution, F, then bidding* $\hat{b}(v) = (N\text{-}1/N)v$ *whenever one's value is v constitutes Nash equilibrium of the discriminatory mechanism, where the nature of the bids are sealed-bids.* ■

So, each agent *shades* its bid, by bidding less than its valuation. Note that as the number of agents increases, the agents bid more aggressively. Because $F^{N-1}(\cdot)$ is the distribution function of the highest value among an agent's N-1 competitors, the bidding strategy displayed in Theorem 2 says that each agent bids the expectation of the second highest agent's value conditional on his value being highest. But, because the agents use the same strictly increasing bidding function, having the highest value is equivalent to having the highest bid and so equivalent to winning the right to replicate.

Theorem 3: *If N agents play their bids according to the bidding strategy as:* $\hat{b}(v) = (N\text{-}1/N)v$*, the corresponding game at instance t and eventually the supergame are in Nash equilibrium.*

Proof: It follows from Lemma 1 and Theorem 1. ■

Discriminatory Mechanism

```
Initialize:
01 LS, L^i.
02 WHILE LS ≠ NULL DO
03   PARFOR each S^i ∈ LS DO                    /*M subgames*/
04        FOR each k ∈ O DO
05             B_k = compute (B_k^i × (N-1)/N);  /*compute benefit*/
06             Report B_k to S_i which is stored in array B;
07        END FOR
08        Sort array B in descending order.
09    WHILE b^i ≥ 0
10    B_k = argmax_k(B);              /*Choose the best offer*/
11    Extract the info from B_k such as O_k and o_k;
12    b^i = b^i - o_k;          /*Calculate space and termination condition*/
13    Replicate O_k;
14    Payment = B_k;                          /* Calculate payment*/
15    Delete B_k from B;              /*Update the list for highest bid*/
16    SEND P^i to S^i; RECEIVE at S^i  /*Agent pays the bid*/
17    L^i = L^i - O_k;                                /*Update the list*/
18    Update NN^i_OMAX              /*Update the nearest neighbor list*/
19    IF L^i = NULL THEN SEND info to M to update LS = LS - S^i;
/*update the player list */
20   END WHILE
21  ENDPARFOR
22 END WHILE
```

Figure 2: Mechanism game at instance *t*.

We are now ready to present the pseudo-code (Figure 2) for a game at instance *t*.

Briefly, we maintain a list L^i at each server. The list contains all the objects that can be replicated at S^i (i.e., the remaining storage capacity b^i is sufficient and the benefit value is positive). We also maintain a list LS containing all servers that can replicate an object. In other words, $S^i \in LS$ if and only if $L^i \neq$ NULL. Each player $k \in O$ calculates the benefit function of object (Line 05). The set O represents the collection of players that are legible for participation. A player k is legible if and only if the benefit function value obtained for site S^i is positive. This is done in order to suppress mediocre bids, which, in turn improves computational complexity. After receiving (Line 06) all the bids, the bid vector is sorted in descending order (Line 08). Now, recursively the rights are assigned to the current highest agent (Line 10) as long as there is available memory (Line 09 and 12). It is to be noted that in each step L^i together with the corresponding nearest server value NN_k^i, are updated accordingly.

The above discussion allows us to deduce the following result about the mechanism.

Theorem 4: *In the worst case the mechanism takes* $O(N^2 logN)$ *time.*

Proof: The worst case scenario is when each site has sufficient capacity to store all objects. In that case, the PARFOR loop (Line 03) performs N iterations. The most consuming time is to sort the bids in descending order (Line 10). This will take at least of the order of $O(N\log N)$. Hence, we conclude that the worst case running time of the mechanism is $O(N^2\log N)$. ■

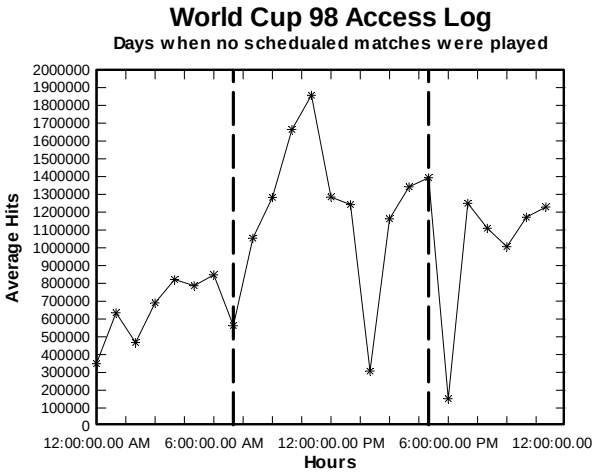


Figure 3(a): Access on days with no matches.

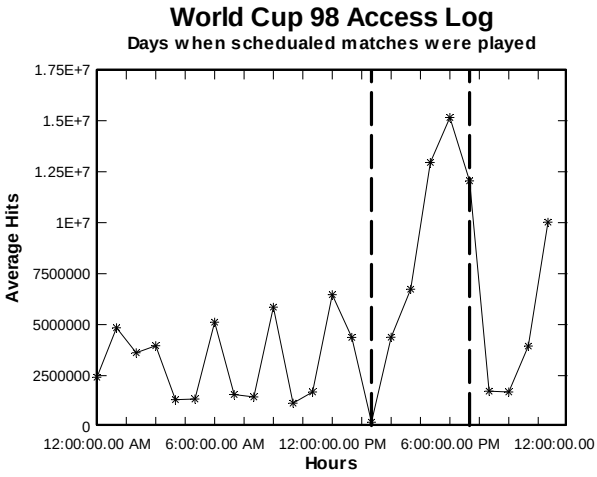


Figure 3(b): Access on days with matches.

3.4 When to invoke the game?

As noted previously (in Sections 2 and 3.3.), the time (interval t) when to initiate the mechanism, *i.e.*, when to play a game at instance t requires high-level human intervention. Here, we will show that this parameter if not totally can at least partially be automated. The decision when to initiate the mechanism depends on the past trends of the user access patterns. The experiments performed to test the mechanism used real user access patterns collected at the 1998 Soccer World Cup website [6]. This access log file has become a default standard over the number of years to benchmark various replica placement techniques. Works reported in [20], [21], [22], [23], [25], and [30] all have used this access log for analysis.

Figures 3(a) and 3(b) show the user access patterns. The two figures represent different traffic patterns, *i.e.*, Figure 3(a) shows the traffic recorded on the days when there was no scheduled match, while Figure 3(b) shows the traffic on the days when there were scheduled matches. We can clearly see that the website incurred soaring and stumpy traffic at various intervals during a 24-hour time period (it is to be noted that the access logs have a time stamp of GMT+1). For example, on days when there was no scheduled match, the traffic was mediocre before 0900 hrs. The traffic increased after 0900 hrs till 2200 hrs. The two vertical dashed lines indicate this phenomenon. These traffic patterns were recorded over a period of 86 days (April 30th 1998 to July 26th 1998). Therefore, on the days when there was no scheduled match, a replica placement algorithm (in our case the mechanism) could be initiated twice daily: 1) at 0900 hrs and 2) at 2200 hrs. The time interval t for 0900 hrs would be t = (2200-0900) = 11 hours and for 2200 hrs would be t = (0900-2200) = 13 hours. On the other hand the days when there were scheduled matches, the mechanism could be initiated at 1900 hrs and 0100 hrs. It is to be noted that the autonomous agents can easily obtain all the other required parameters (for the DRP) via the user access logs and the underlying network architecture.

4 Experimental Setup and the Discussion of Results

We performed experiments on a 440MHz Ultra 10 machine with 512MB memory. The experimental evaluations were targeted to benchmark the placement policies. The mechanism was implemented using IBM Pthreads.

To establish diversity in our experimental setups, the network connectively was changed considerably. In this paper, we only present the results that were obtained using a maximum of 500 sites (nodes). We used existing topology generator toolkits and also self generated networks. In all the topologies the distance of the link between nodes was equivalent to the communication cost. Table 2 summarizes the various techniques used to gather forty-five various topologies for networks with 100 nodes. It is to be noted that the parameters vary for networks with lesser/larger number of nodes.

To evaluate the chosen replication placement techniques on realistic traffic patterns, we used the access logs collected at the Soccer World Cup 1998 website [6]. Each experimental setup was evaluated thirteen times, i.e., the Friday (24 hours) logs from May 1, 1998 to July 24, 1998. Thus, each experimental setup in fact represents an average of the 585 (13×45) data set points. To process the logs, we wrote a script that returned: only those objects which were present in all the logs (2000 in our case), the total number of requests from a particular client for an object, the average and the variance of the object size. From this log we chose the top five hundred clients (maximum experimental setup). A random mapping was then performed of the clients to the nodes of the topologies. Note that this mapping is not 1-1, rather 1-M. This gave us enough skewed workload to mimic real world scenarios. It is also worthwhile to mention that the total amount of requests entertained for each problem instance was in the range of 1-2 million. The primary replicas' original site was mimicked by choosing random locations. The capacities of the sites C% were generated randomly with range from *Total Primary Object Sizes*/2 to 1.5×*Total Primary Object*

Topology	Mathematical Representation	Parameter Interval Variance
SGRG (12 topologies)	Randomized layout with node degree (d*) and Euclidian distance (d) between nodes as parameters.	d={5,10,15,20}, d*={10,15,20}.
GT-ITM PR [9] (5 topologies)	Randomized layout with edges added between the randomly located vertices with a probability (p).	p={0.4,0.5,0.6,0.7,0.8}.
GT-ITM W [9] (9 topologies)	$P(u,v)=\alpha e^{-d/(\beta L)}$	α={0.1,0.15,0.2,0.25}, β={0.2,0.3,0.4}.
SGFCGUD (5 topologies)	Fully connected graph with uniform link distances (d).	d_1=[1,10],d_2=[1,20],d_3=[1,50], d_4=[10,20], d_5=[20,50].
SGFCGRD (5 topologies)	Fully connected graph with random link distances (d).	d_1=[1,10],d_2=[1,20],d_3=[1,50], d_4=[10,20], d_5=[20,50].
SGRGLND (9 topologies)	Random layout with link distance having a lognormal distribution [15].	μ={8.455,9.345,9.564}, σ={1.278,1.305,1.378}.

Table 2: Parameter interval variance characterization for topologies with 100 nodes.

Sizes. The variance in the object size collected from the access logs helped to install enough diversity to benchmark object updates. The updates were randomly pushed onto different sites, and the total system update load was measured in terms of the percentage update requests U% compared that to the initial network with no updates.

4.1 Comparative Algorithms

For comparisons, we selected five various types of replica placement techniques. To provide a fair comparison, the assumptions and system parameters were kept the same in all the approaches. The techniques studied include efficient branch-and-bound based technique (Aε-Star [22]). For fine-grained replication, the algorithms proposed in [23], [25], [26], and [30] are the only ones that address the problem domain similar to ours. We select from [30] the greedy approach (Greedy) for comparison because it is shown to be the best compared with four other approaches (including the proposed technique in [25]); thus, we indirectly compare with four additional approaches as well. Algorithms reported in [23] (Dutch (DA) and English auctions (EA)) and [26] (Genetic based algorithm (GRA)) are also among the chosen techniques for comparisons. Due to space limitations we will only give a brief overview of the comparative techniques. Details for a specific technique can be obtained from the referenced papers.

Performance metric: The solution quality is measured in terms of network communication cost (OTC percentage) that is saved under the replication scheme found by the algorithms, compared to the initial one, i.e., when only primary copies exists.

Aε-Star: In [22] the authors proposed a 1+ε admissible A-Star based technique called Aε-Star. This technique uses two lists: OPEN and FOCAL. The FOCAL list is the sub-list of OPEN, and only contains those nodes that do not deviate from the lowest f node by a factor greater than 1+ε. The technique works similar to A-Star, with the exception that the node selection (lowest h) is done not from the OPEN but from the FOCAL list. It is easy to see that this approach will never run into the problem of memory overflow, moreover, the FOCAL list always ensures that only the candidate solutions within a bound of 1+ε of the A-Star are expanded.

Greedy based technique: We modify the greedy approach reported in [30], to fit our problem formulation. The greedy algorithm works in an iterative fashion. In the first iteration, all the M sites are investigated to find the replica location(s) of the first among a total of N objects. Consider that we choose an object i for replication. The algorithm recursively makes calculations based on the assumption that all the users in the system request for object i. Thus, we have to pick a site that yields the lowest cost of replication for the object i. In the second iteration, the location for the second site is considered. Based on the choice of object i, the algorithm now would identify the second site for replication, which, in conjunction with the site already picked, yields the lowest replication cost. The readers will immediately realize that the bidding mechanism reported in this paper works similar to the Greedy algorithm. This is true; however, the Greedy approach does not guarantee optimality even if the algorithm is run on the very same problem instance. Recall that Greedy relies on making combinations of object assignments and therefore, suffers from the initial choice of object selection (which is done randomly).

Dutch auction: The auctioneer begins with a high asking price which is lowered until some agent is willing to accept the auctioneer's price. That agent pays the last announced price. This type of auction is convenient when it is important to auction objects quickly, since a sale never requires more than one bid. In no case does the auctioneer reveal any of the bids submitted to him, and no information is shared between the agents. It is shown that for an agent to have a probabilistically superior bid than n-1 other bids; agent should have the valuation divided by n [23].

English auction: In this type of auction, the agents bid openly against one another, with each bid being higher than the previous bid. The auction ends when no agent is willing to bid further. During the auction when an auctioneer receives a bid higher than the currently submitted bids, he announces the bid value so that other agents (if needed) can revise their currently submitted bids. In [23], the discussion on EA reveals that the optimal strategy for a bidder i is to bid a value which is directly derived from his valuation.

GRA: In [26], the authors proposed a genetic algorithm based heuristic called GRA. GRA provides good solution quality, but suffers from slow termination time. This algorithm was selected since it realistically addressed the fine-grained data replication using the same problem formulation as undertaken in this article.

4.2 Comparative Game Analysis

First, we concentrate on observing the improvement brought by the discriminatory mechanism (for short we will refer to it as MECH). To this end we observe the solution quality at the game level. In the post-ceding text (Section 4.3.) we shall discuss the results obtained in the supergame setup.

We study the behavior of the placement techniques when the number of sites increases (Figure 4), by setting the number of objects to 2000, while in Figure 5, we study the behavior when the number of objects increase, by setting the number of sites to 500. We should note here that the space limitations restricted us to include various other scenarios with varying capacity and update ratio. The plot trends were similar to the ones reported in this article. For the first experiment we fixed C = 30% and U = 65%. We intentionally chose a high workload so as to see if the techniques studied successfully handled the extreme cases. The first observation is that MECH and EA outperformed other techniques by considerable amounts. Second, DA converged to a better solution quality under certain problem instances than EA. This is inline with the general trends of DA. It outperforms EA when the agents are bidding aggressively. Some interesting observations were also recorded, such as, all but GRA and Greedy showed initial loss in OTC savings with the initial number of site increase in the system, as much as 5% loss was recorded in case of MECH with only a 40 site increase. GRA and Greedy showed an initial gain since with the increase in the number of sites, the population permutations increase exponentially, but with the further increase in the number of sites this phenomenon is not so observable as all the essential objects are already replicated. The top performing techniques (DA, EA, Aε-Star and MECH) showed an almost constant performance increase (after the initial loss in OTC savings). This is because by adding a site (server) in the network, we introduce additional traffic (local requests), together with more storage capacity available for replication. All four equally cater for the two diverse effects. GRA also showed a similar trend but maintained lower OTC savings. This was in line with the claims presented in [22] and [26].

To observe the effect of increase in the number of objects in the system, we chose a softer workload with C = 15% and U = 40%. The intention was to observe the trends for all the algorithms under various workloads. The increase in the number of objects has diverse effects on the system as new read/write patterns (users are offered more choices) emerge, and also the increase in the strain on the overall capacity of the system (increase in the number of replicas). An effective algorithm should incorporate both the opposing trends. From the plot, the most surprising result came from GRA and Greedy. They dropped their savings from 62% to 2% and 69% to 3%, respectively. This was contradictory to what was reported in [26] and [30]. But there the authors had used a uniformly distributed link cost topology, and their traffic was based on the Zipf distribution [33]. While the traffic access logs of the World Cup 1998 are more or less double-Pareto in nature. In either case the exploits and limitations of the technique under discussion are obvious. The plot also shows a near identical performance by Aε-Star, DA and Greedy. The relative difference among the three techniques is less than 3%. However, Aε-Star did maintain its domination. From the plots the supremacy of EA and MECH is observable.

Next, we observe the effects of system capacity increase. An increase in the storage capacity means that a large number of objects can be replicated. Replicating an object that is already extensively replicated, is unlikely to result in significant traffic savings as only a small portion of the servers will be affected overall. Moreover, since objects are not equally read intensive, increase in the storage capacity would have a great impact at the beginning (initial increase in capacity), but has little effect after a certain point, where the most beneficial ones are already replicated. This is observable in Figure 6, which shows the performance of the algorithms. GRA once again performed the worst. The gap between all other approaches was reduced to within 15% of each other. DA and MECH showed an immediate initial increase (the point after which further replicating objects is inefficient) in its OTC savings, but afterward showed a near constant performance. GRA although performed the worst, but observably gained the most OTC savings (53%) followed by Greedy with 34%. Further experiments with various update ratios (5%, 10%, and 20%) showed similar plot trends. It is also noteworthy (plots not shown in this paper due to space restrictions) that the increase in capacity from 13% to 24%, resulted in 4.3 times (on average) more replicas for all the algorithms.

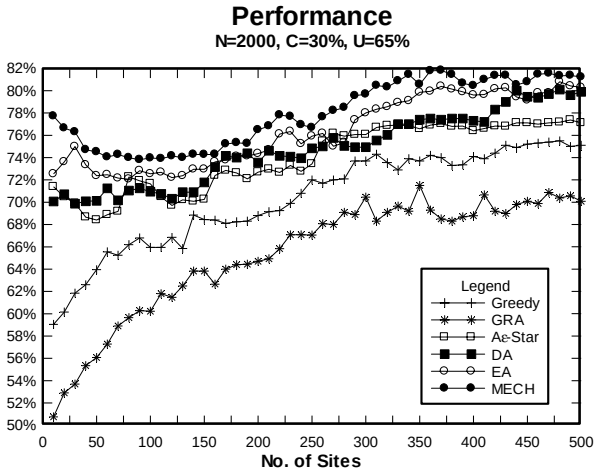


Figure 4: OTC savings versus number of sites.

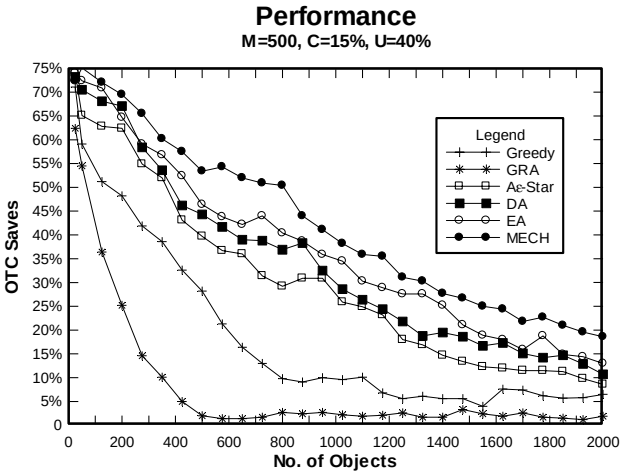


Figure 5: OTC savings versus number of objects.

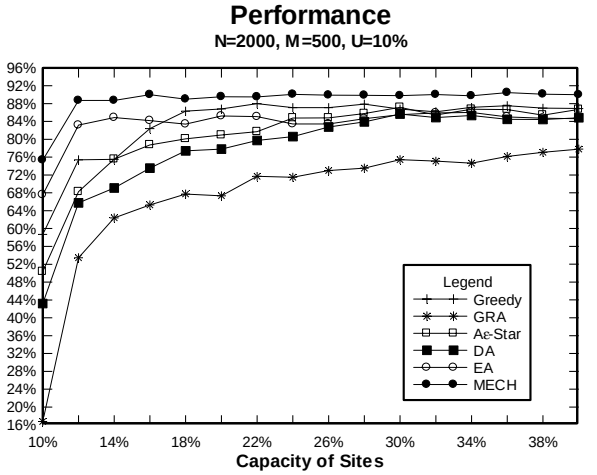


Figure 6: OTC savings versus capacity.

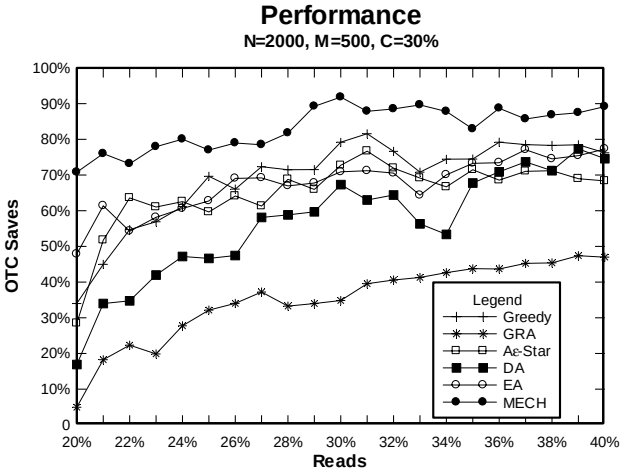


Figure 7: OTC savings versus reads.

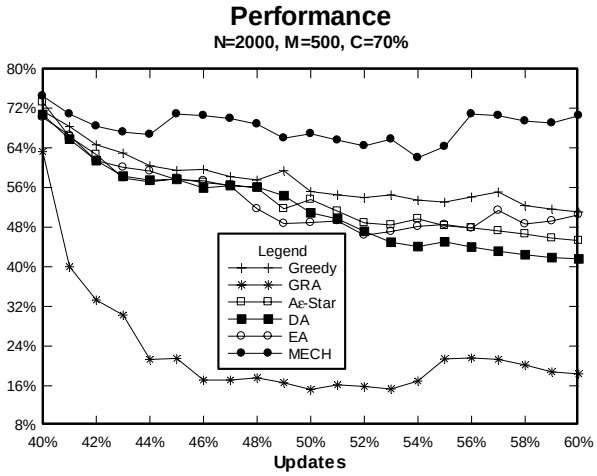


Figure 8: OTC savings versus updates.

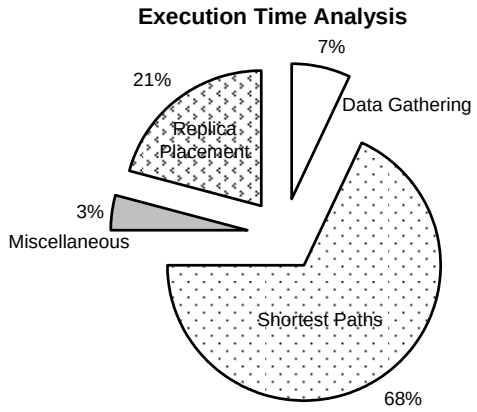


Figure 9: Execution time components.

Next, we observe the effects of increase in the read and update (write) frequencies. Since these two parameters are complementary to each other, we describe them together. In both the setups the number of sites and objects were kept constant. Increase in the number of reads in the system would mean that there is a need to replicate as many object as possible (closer to the users). However, the increase in the number of updates in the

system requires the replicas be placed as close as to the primary site as possible (to reduce the update broadcast). This phenomenon is also interrelated with the system capacity, as the update ratio sets an upper bound on the possible traffic reduction through replication. Thus, if we consider a system with unlimited capacity, the "replicate everywhere anything" policy is strictly inadequate. The read and update parameters indeed help in drawing a line between good and marginal algorithms. The plots in Figures 7 and 8 show the results of read and update frequencies, respectively. A clear classification can be made between the algorithms. Aε-Star, DA, EA, Greedy and MECH incorporate the increase in the number of reads by replicating more objects and thus savings increase up to 89%. Aε-Star gained the most of the OTC savings of up to 47%. To understand why there is such a gap in the performance between the algorithms, we should recall that GRA specifically depend on the initial population (for details see [26]). Moreover, GRA maintains a localized network perception. Increase in updates result in objects having decreased local significance (unless the vicinity is in close proximity to the primary location). On the other hand, Aε-Star, DA, EA, Greedy never tend to deviate from their global view of the problem domain.

Lastly, we compare the termination time of the algorithms. Before we proceed, we would like to clarify our measurement of algorithm termination timings. The approach we took was to see if these algorithms can be used in dynamic scenarios. Thus, we gather and process data as if it was a dynamic system. The average breakdown of the execution time of all the algorithms combined is depicted in Figure 9. There 68% of all the algorithm termination time was taken by the repeated calculations of the shortest paths. Data gathering and dispersion, such as reading the access frequencies from the processed log, etc. took 7% of the total time. Other miscellaneous operations including I/O were recorded to carry 3% of the total execution time. From the plot it is clear that a totally static setup would take no less that 21% of the time depicted in Tables 3 and 4.

Various problem instances were recorded with C =

Problem Size	Greedy	GRA	Aε-Star	DA	EA	MECH
M=20, N=50	69.76	92.57	97.02	**24.66**	39.29	25.24
M=20, N=100	76.12	96.31	102.00	26.97	40.91	**26.35**
M=20, N=150	78.11	100.59	113.79	**31.98**	53.85	35.64
M=30, N=50	94.33	125.93	139.98	38.20	58.98	**38.05**
M=30, N=100	108.18	124.20	148.03	**38.29**	62.97	39.60
M=30, N=150	134.97	148.49	178.84	44.97	67.74	**42.02**
M=40, N=50	126.25	153.93	198.11	**42.34**	75.88	44.66
M=40, N=100	134.06	168.09	236.48	**43.54**	76.27	46.31
M=40, N=150	140.30	204.12	270.10	**47.02**	82.44	48.41

Table 3: Running time in seconds [C=20%, U=25%].

Problem Size	Greedy	GRA	Aε-Star	DA	EA	MECH
M=300, N=1450	206.26	326.82	279.45	**95.64**	178.9	97.98
M=300, N=1500	236.61	379.01	310.12	115.19	185.15	**113.65**
M=300, N=1550	258.45	409.17	333.03	127.1	191.24	**124.73**
M=300, N=2000	275.63	469.38	368.89	143.94	197.93	**142.16**
M=400, N=1450	321.6	492.1	353.08	**176.51**	218.15	176.90
M=400, N=1500	348.53	536.96	368.03	**187.26**	223.56	195.41
M=400, N=1550	366.38	541.12	396.96	**192.41**	221.1	214.55
M=400, N=2000	376.85	559.74	412.17	**208.92**	245.47	218.73
M=500, N=1450	391.55	659.39	447.97	**224.18**	274.24	235.17
M=500, N=1500	402.2	660.86	460.44	**246.43**	284.63	259.56
M=500, N=1550	478.1	689.44	511.69	**257.96**	301.72	266.42
M=500, N=2000	485.34	705.07	582.71	269.45	315.13	**262.68**

Table 4: Running time in seconds [C=35%, U=35%].

Problem Size	Greedy	GRA	Aε-Star	DA	EA	MECH
N=150, M=20 [C=20%,U=25%]	70.27	69.11	73.96	69.91	72.72	**74.40**
N=200, M=50 [C=20%,U=20%]	73.49	69.33	76.63	71.90	**77.11**	75.43
N=300, M=50 [C=25%,U=5%]	69.63	63.45	69.85	67.66	69.80	**70.36**
N=300, M=60 [C=35%,U=5%]	71.15	64.95	71.51	69.26	70.38	**74.03**
N=400, M=100 [C=25%,U=25%]	67.24	61.74	71.26	68.67	70.49	**73.26**
N=500, M=100 [C=30%,U=35%]	65.24	60.77	70.55	69.82	70.87	**72.73**
N=800, M=200 [C=25%,U=15%]	66.53	65.90	69.33	68.95	70.06	**72.95**
N=1000, M=300 [C=25%,U=35%]	69.04	63.17	69.98	69.36	71.28	**72.44**
N=1500, M=400 [C=35%,U=50%]	69.98	62.61	70.41	72.09	72.26	**72.78**
N=2000, M=500 [C=10%,U=60%]	66.34	62.70	71.33	67.67	68.41	**74.06**

Table 5: Average OTC (%) savings under some problem instances.

20%, 35% and U = 25%, 35%. Each problem instance represents the average recorded time over all the 45 topologies and 13 various access logs. The entries in bold represent the fastest time recorded over the problem instance. It is observable that MECH and DA terminated faster than all the other techniques, followed by EA, Greedy, Aε-Star and GRA. If a static environment was considered, MECH with the maximum problem instance would have terminated approximately in 55.16 seconds (21% of the algorithm termination time).

In summary, based on the solution quality alone, the algorithms can be classified into four categories: 1) Very high performance: EA and MECH, 2) high performance: Greedy and DA, 3) medium-high performance: Aε-Star, and finally 4) mediocre performance: GRA. Considering the execution time, MECH and DA did extremely well, followed by EA, Greedy, Aε-Star, and GRA.

Table 5 shows the quality of the solution in terms of OTC percentage for 10 problem instances (randomly chosen), each being a combination of various numbers of sites and objects, with varying storage capacity and update ratio. For each row, the best result is indicated in bold. The proposed MECH algorithm steals the show in the context of solution quality, but Aε-Star, EA and DA do indeed give a good competition, with a savings within 5%-10% of MECH.

4.3 Comparative Supergame Analysis

Here, we present some supplementary results regarding the supergame that strengthen our comparative analysis claims provided in Section 4.2. We show the relative performance of the techniques with load and storage capacity variance. The plots in Figures 10-13 show the recorded performances. All the plots summarize the measured performance with varying parameters observed over a time period of 86 simulation days (this is the entire time period of the logs that are available for the World Cup 1998 web server). Notice that the supergame setup is tested over all the available access logs. We are mostly interested in measuring the median and mean performances of the algorithms. With load variance MECH edges over EA with a savings of 39%. The plot also shows that nearly every algorithm performed well with a grand median of 15.9%. The

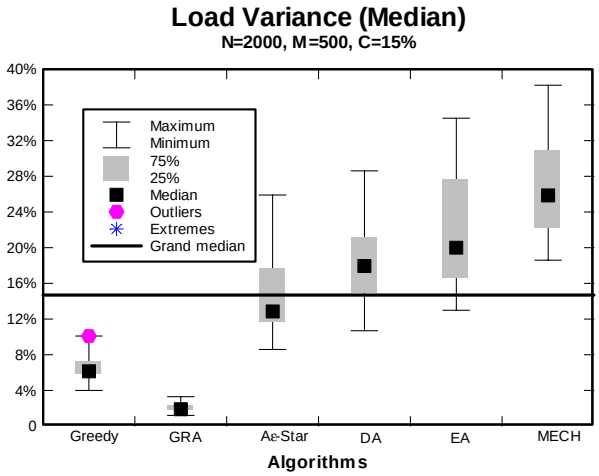


Figure 10: Median load variance.

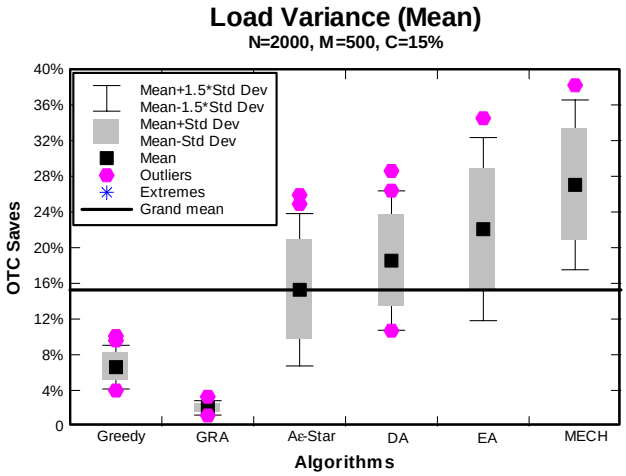


Figure 11: Mean load variance.

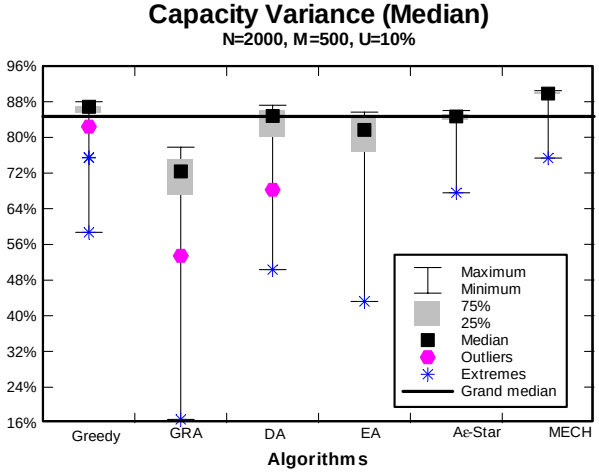


Figure 12: Median capacity variance.

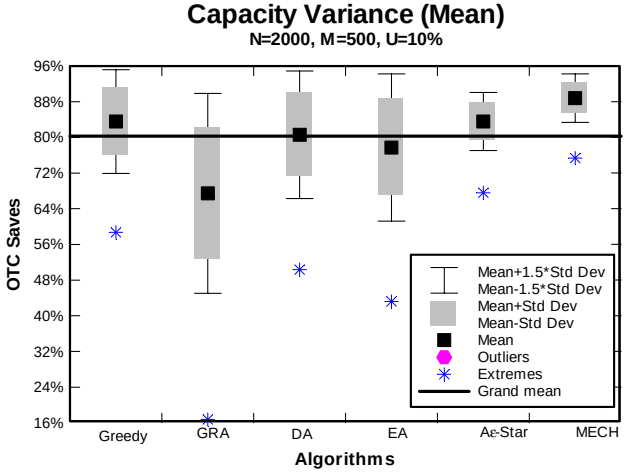


Figure 13: Mean capacity variance.

graphs are self explanatory in nature, and also capture the outliners and extreme points. The basic exercise in plotting these results is to see which algorithms perform consistently over an extended period of time. GRA for example, records the lowest extremes, and hardly any outliners. On the other hand the proposed MECH's performance is captured in a small interval, with high median and mean OTC savings. The readers may notice the difference in the performance of the algorithms with load and capacity variances. This is because load variance captures all the possible combinations of read and update parameters. For example, in a network with 100% updates there will hardly be any measurable OTC Savings. Thus, Figures 10 and 11 show mediocre OTC savings, simply because they encapsulated the performance of the networks where update ratio was extremely high.

5 Related Work

The data replication problem as presented in Section 2 is an extension of the classical file allocation problem (FAP). Chu [11] studied the file allocation problem with respect to multiple files in a multiprocessor system. Casey [10] extended this work by distinguishing between updates and read file requests. Eswaran [14] proved that Casey's formulation was NP complete. In [28] Mahmoud et al. provide an iterative approach that achieves good solution quality when solving the FAP for infinite server capacities. A complete although old survey on the FAP can be found in [13]. Apers in [4] considered the data allocation problem (DAP) in distributed databases where the query execution strategy influences allocation decisions. In [24] the authors proposed several algorithms to solve the data allocation problem in distributed multimedia databases (without replication), also called as video allocation problem (VAP).

Most of the research papers outlined in [13] aim at formalizing the problem as an optimization one, sometimes using multiple objective functions. Network traffic, server throughput and response time exhibited by users are considered for optimization. Although a lot of effort was devoted in providing comprehensive models, little attention has been paid to good heuristics for solving this complex problem. Furthermore access patterns are assumed to remain static and solutions in the dynamic case are obtained by re-executing a time consuming mathematical programming technique.

Some on-going work is related to dynamic replication of objects in distributed systems when the read-write patterns are not known apriori. Awerbuch's et al. work in [7] is significant from a theoretical point of view, but the adopted strategy for commuting updates (object replicas are first deleted), can prove difficult to implement in a real-life environment. In [33] Wolfson et al. proposed an algorithm that leads to optimal single file replication in the case of a tree network. The performance of the scheme for general network topologies is not clear though. Dynamic replication protocols were also considered under the Internet environment. Heddaya et al. [19] proposed protocols that load balance the workload among replicas. In [31], Rabinovich et al. proposed a protocol for dynamically replicating the contents of an ISP (Internet Service Provider) in order to improve client-server proximity without overloading any of the servers. However updates were not considered.

Our work differs from all the above in: 1) Taking into account the more pragmatic scenario in today's distributed information environments, we tackle the case of allocating replicas so as to minimize the network traffic under storage constraints with "read from the nearest" and "update through the primary server" policies, and 2) in using game theoretical techniques.

6 Concluding Remarks

This paper proposed a game theoretical discriminatory mechanism (MECH) for fine-grained data replication in large-scale distributed computing systems (e.g. the Internet). In MECH we employ agents who represent data objects to compete for the limited available storage space on web servers to acquire the rights to replicate. MECH uses a unique concept of supergame in which these agents continuously compete in a non-cooperative environment. MECH allows the designers the flexibility to monitor the behavior and strategies of these agents and fine-tune them so as to attain a given objective. In case of the data replication problem, the object for these agents is to skillfully replicate data objects so that the total object transfer cost is minimized.

MECH was compared against some well-known techniques, such as: greedy, branch and bound, game theoretical auctions and genetic algorithms. To provide a fair comparison, the assumptions and system parameters were kept the same in all the approaches. The experimental results revealed that MECH outperformed the five widely cited and powerful techniques in both the execution time and solution quality.

In summary, MECH exhibited 5%-10% better solution quality and 25%-35% savings in the algorithm termination timings.

References

[4] V. Almeida, A. Bestavros, M. Crovella and A. de Oliveria, "Characterizing reference locality in the WWW," in *Proc. of International Conference on Parallel and Distributed Information Systems*, 1996, pp. 92-103.

[5] P. Apers, "Data Allocation in Distributed Database Systems," *ACM Trans. Database Systems*, 13(3), pp. 263-304, 1988.

[6] M. Arlitt and T. Jin, "Workload characterization of the 1998 World Cup Web Site," Tech. report, Hewlett Packard Lab, Palo Alto, HPL-1999-35(R.1), 1999.

[7] B. Awerbuch, Y. Bartal and A. Fiat, "Competitive Distributed File allocation," in *Proc. 25th ACM STOC*, Victoria, B.C., Canada, 1993, pp. 164-173.

[8] L. Breslau, P. Cao, L. Fan, G. Philips and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. of IEEE INFOCOM*, 1999, pp. 126-134.

[9] K. Calvert, M. Doar, E. Zegura, "Modeling Internet Topology," *IEEE Communications*, 35(6), pp. 160-163, 1997.

[10] R. Casey, "Allocation of Copies of a File in an Information Network," in *Proc. Spring Joint Computer Conf., IFIPS*, 1972, pp. 617-625.

[11] W. Chu, "Optimal File Allocation in a Multiple Computer System," *IEEE Trans. on Computers*, C-18(10), pp. 885-889, 1969.

[12] E. Clarke, "Multi Pricing of Public Goods," *Public Choice*, vol. 11, pp. 17-33, 1971.

[13] L. Dowdy and D. Foster, "Comparative Models of the File Assignment problem," *ACM Computing Surveys*, 14(2), pp. 287-313, 1982.

[14] K. Eswaran, "Placement of Records in a File and File Allocation in a Computer Network," *Information Processing Letters*, pp. 304-307, 1974.

[15] S. Floyd and V. Paxson, "Difficulties in Simulating the Internet," *IEEE/ACM Trans. Networking*, 9(4), pp. 253-285, 2001.

[16] J. Green and J. Laffont, "Characterization of Satisfactory Mechanisms for the revelation of Preferences for Public Goods," *Econometrica*, pp. 427-438, 1977.

[17] T. Groves, "Incentives in Teams," *Econometrica*, pp. 617-631, 1973.

[18] D. Grosu and A. Chronopoulos, "Algorithmic Mechanism Design for Load Balancing in Distributed Systems," *IEEE Trans. Systems, Man and Cybernatics B*, 34(1), pp. 77-84, 2004.

[19] A. Heddaya and S. Mirdad, "WebWave: Globally Load Balanced Fully Distributed Caching of Hot Published Documents," in *Proc. 17th International Conference on Distributed Computing Systems*, Baltimore, Maryland, 1997, pp. 160-168.

[20] S. Jamin, C. Jin, Y. Jin, D. Riaz, Y. Shavitt and L. Zhang, "On the Placement of Internet Instrumentation," in *Proc. of the IEEE INFOCOM*, 2000, pp. 295-304.

[21] S. Jamin, C. Jin, T. Kurc, D. Raz and Y. Shavitt, "Constrained Mirror Placement on the Internet," in *Proc. of the IEEE INFOCOM*, 2001, pp. 31-40.

[22] S. Khan and I. Ahmad, "Heuristic-based Replication Schemas for Fast Information Retrevial over the Internet," in *Proc. of 17th International Conference on Parallel and Distributed Computing Systems, San Fransisco*, U.S.A., 2004, pp. 278-283.

[23] S. Khan and I. Ahmad, "A Powerful Direct Mechanism for Optimal WWW Content Replication," in *Proc. of 19th IEEE International Parallel and Distributed Processing Symposium*, Denver, U.S.A., 2005, p. 86.

[24] Y. Kwok, K. Karlapalem, I. Ahmad and N. Pun, "Design and Evaluation of Data Allocation Algorithms for Distributed Database Systems," *IEEE Journal on Selected areas in Communication*, 14(7), pp. 1332-1348, 1996.

[25] B. Li, M. Golin, G. Italiano and X. Deng, "On the Optimal Placement of Web Proxies in the Internet," in *Proc. of the IEEE INFOCOM*, 2000, pp. 1282-1290.

[26] T. Loukopoulos, and I. Ahmad, "Static and Adaptive Distributed Data Replication using Genetic Algorithms," *Journal of Parallel and Distributed Computing*, 64(11), pp. 1270-1285, 2004.

[27] R. MacAfee and J. McMillan, "Auctions and Bidding," *Journal of Economic Literature*, vol. 25, pp. 699-738, 1987.

[28] S. Mahmoud and J. Riordon, "Optimal Allocation of Resources in Distributed Information Networks," *ACM Trans. on Database Systems*, 1(1), pp. 66-78, 1976.

[29] N. Nisan and A. Ronen, "Algorithmic Mechanism Design," in *Proc. of 31st ACM STOC*, 1999, pp. 129-140.

[30] L. Qiu, V. Padmanabhan and G. Voelker, "On the Placement of Web Server Replicas," in *Proc. of the IEEE INFOCOM*, 2001, pp. 1587-1596.

[31] M. Rabinovich, "Issues in Web Content Replication," *Data Engineering Bulletin*, 21(4), pp. 21-29, 1998.

[32] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," Internet Engineering Task Force, RFC 1771, 1995.

[33] O. Wolfson, S. Jajodia and Y. Hang, "An Adaptive Data Replication Algorithm," *ACM Trans. on Database Systems*, 22(4), pp. 255-314, 1997.

[34] G. Zipf, Human Behavior and the Principle of Least-Effort, Addison-Wesley, 1949.

An Efficient Chaos-Based Feedback Stream Cipher (ECBFSC) for Image Encryption and Decryption

Hossam El-din H. Ahmed, Hamdy M. Kalash, and Osama S. Farag Allah
Menoufia University, Faculty of Electronic Engineering, Dept. of Computer Science & Engineering,
32952, Menouf, Egypt
E-mail of corresponding author: osam_sal@yahoo.com

Keywords: stream cipher, chaos, logistic map, security analysis

Received: September 23, 2005

The chaos based cryptographic algorithms have suggested some new and efficient ways to develop secure image encryption techniques. Towards this direction, this paper presents an efficient chaos-based feedback stream cipher (ECBFSC) for image cryptosystems. The proposed stream cipher is based on the use of a chaotic logistic map and an external secret key of 256-bit. The initial conditions for the chaotic logistic map are derived using the external secret key by providing weightage to its bits corresponding to their position in the key. Further, new features of the proposed stream cipher include the heavy use of data-dependent iterations, data-dependent inputs, and the inclusion of three independent feedback mechanisms. These proposed features are verified to provide high security level. A complete specification for the proposed ECBFSC is given. Several test images are used for inspecting the validity of the proposed ECBFSC. The results of several experimental, key space analysis, statistical analysis, and key sensitivity tests show that the proposed ECBFSC for image cryptosystems provides an efficient and secure way for real-time image encryption and transmission from the cryptographic viewpoint.

Povzetek: Opisan je na teoriji kaosa razvit kriptografski algoritem.

1 Introduction

During the last decade, the use of computer networks has grown spectacularly, and this growth continues unabated. Almost all networks are being installed, interconnected, and connected to the global internet. Today more and more information has been transmitted over the internet. The information is not only text, but also audio, image, and other multimedia. Images have been widely used in our daily life. However, the more extensively we use the images, the more important their security will be. For example, it is important to protect the diagrams of army emplacements, the diagrams of bank building construction, and the important data captured by military satellites. In addition, the number of computer crimes has increased recently. Image security has become an important topic in the current computer world.

Many encryption methods have been proposed in literature, and the most common way to protect large multimedia files is by using conventional encryption techniques. Implementations of popular public key encryption methods, such as RSA or El-Gamal [1] cannot provide suitable encryption rates, while security of these algorithms relies on the difficulty of quickly factorizing large numbers or solving the discrete logarithm problem, topics that are seriously challenged by recent advances in number theory and distributed computing.

On the other hand, private key bulk encryption algorithms, such as Triple DES or Blowfish [2], are more suitable for transmission of large amounts of data. However, due to the complexity of their internal structure, they are not particularly fast in terms of execution speed and cannot be concisely and clearly explained, so that to enable detection of cryptanalytic vulnerabilities.

Chaotic maps present many desired cryptographic qualities such as simplicity of implementation that leads to high encryption rates, and excellent security.

In this paper we propose an efficient chaos-based feedback stream cipher (ECBFSC) for image cryptosystems. The proposed ECBFSC works using an iterative cipher mechanism that is based on the logistic function. The encryption module encrypts the image pixel-by-pixel, taking into consideration, in each iteration, the values of the previously encrypted pixels. This feedback property, combined with the external secret key of 256-bit, makes our stream cipher robust against cryptanalytic attacks. Furthermore a simple implementation of ECBFSC achieves high encryption rates on general-purpose computers.

The rest of this paper is organized as follows: Section 2 surveys some related image cryptosystems. We discuss some characteristics of an image cryptosystem and research issues of an image cryptosystem in Sections 3 and 4 respectively. Section 5 presents chaos and cryptography including characteristics and analysis of

chaotic logistic map. Section 6 examines the step by step procedure of encryption/decryption modules for the proposed ECBFSC. Section 7 explores design Principles of the ECBFSC. Test, verification and efficiency of the proposed ECBFSC are given in Section 8. Section 9 discusses the detailed security analysis of the proposed ECBFSC including key space analysis, statistical analysis, and sensitivity analysis with respect to key and plaintext. Performance evaluation of the proposed ECBFSC is explored in Section 10. Finally, Section 11 concludes this paper.

2 Related Image Cryptosystems

2.1 Picture data encryption using SCAN patterns

First sub Bourbakis and Alexopoulos [3] developed another method to encrypt images. This method converts a 2D image into a 1D list, and employs a SCAN language [4] to describe the converted result. In this language, there are several SCAN letters. Each SCAN letter represents one kind of scan order. Different kinds of combinations of SCAN letters may generate different kinds of secret images. After determining the combination of SCAN letters, the scheme then generates a SCAN string. This string defines the scan order of the original image. Next, this method scans the original image in the determined order and, moreover, encrypts the SCAN string by using commercial cryptosystems. Since the illegal users cannot obtain the correct SCAN string, the original image is therefore secure. There is no image compression in this method. Therefore, the size of the image is very large, and thus it is inefficient to encrypt or decrypt the image directly.

2.2 Novel image encryption technique and its application in progressive transmission

Kuo proposed an encryption method that referred to the image distortion [5]. This method obtains the encrypted image by adding the phase spectra of the plainimage with those of another key image. Since the phase spectra of the original image are randomly changed, the cipherimage is unrecognizable. Thus this method is safe, but no image compression is considered.

2.3 An image encryption scheme based on quadtree compression scheme

Chang and Liou [6] proposed an encryption method for images. This method employs two technologies to achieve the compression and encryption purposes. They are the quadtree data structure and the SCAN language, respectively. This method first compresses the original image by using a quadtree, and then encrypts the compressed data by SCAN. So, this method can compress and encrypt images concurrently. Quadtree is notably a lossless data compression technology. Therefore, this method is also lossless.

2.4 A New Encryption Algorithm for Image Cryptosystems

Chin-Chen Chang, Min-Shian Hwang, and Tung-Shou Chen [7] use one of the popular image compression techniques, vector quantization to design an efficient cryptosystem for images. The scheme is based on vector quantization (VQ), cryptography, and other number theorems. The images are first decomposed into vectors and then sequentially encoded vector by vector.

2.5 Symmetric ciphers based on two dimensional chaotic maps

Fridrich [8] demonstrated the construction of a symmetric block encryption technique based on two-dimensional standard baker map. There are three basic steps in the method of Fridrich [8]: (a) choose a chaotic map and generalize it by introducing some parameter, (b) discretize the chaotic map to a finite square lattice of points that represent pixels, (c) extend the discretized map to three-dimensions and further compose it with a simple diffusion mechanism. Further,

2.6 Fast encryption of image data using chaotic Kolmogrov

Scharinger [9] designed a chaotic Kolmogrov-flow-based image encryption technique, in which whole image is taken as a single block and which is permuted through a key-controlled chaotic system. In addition, a shift register pseudo random generator is also adopted to introduce the confusion in the data.

2.7 A new image encryption algorithm and its VLSI architecture

Yen and Guo [10] proposed an encryption method called BRIE based on chaotic logistic map. The basic principle of BRIE is bit recirculation of pixels, which is controlled by a chaotic pseudo random binary sequence. The secret key of BRIE consists of two integers and an initial condition of the logistic map.

2.8 A new chaotic key based design for image encryption and decryption

Further, Yen and Guo [11] also proposed an encryption method called CKBA (Chaotic Key Based Algorithm) in which a binary sequence as a key is generated using a chaotic system. The image pixels are rearranged according to the generated binary sequence and then XORed and XNORed with the selected key.

2.9 Chaotic encryption scheme for real time digital video

Recently, Li et al. [12] have proposed a video encryption technique based on multiple digital chaotic systems which is known as CVES (Chaotic Video Encryption Scheme). In this scheme, 2n chaotic maps are used to generate pseudo random signals to mask the video and to perform pseudo random permutation of the masked video. Very recently,

2.10 A symmetric image encryption based on 3D chaotic maps

Chen et al. [13] have proposed a symmetric image encryption in which a two-dimensional chaotic map is generalized to three-dimension for designing a real time secure image encryption scheme. This approach employs the three-dimensional cat map to shuffle the positions of the image pixels and uses another chaotic map to confuse the relationship between the encrypted and its original image.section text.

3 Characteristics of An Image Cryptosystem

A good information security system is able to not only protect confidential messages in the text form, but also in image form. In general, there are three basic characteristics in the information security field: privacy, integrity, and availability [14].

1. Privacy: an unauthorized user cannot disclose a message.
2. Integrity: an unauthorized user cannot modify or corrupt a message.
3. Availability: messages are made available to authorized users faithfully.

A perfect image cryptosystem is not only flexible in the security mechanism, but also has high overall performance. Thus, besides the above characteristics, the image security also requires the following characteristics:

1. The encryption system should be computationally secure. It must require an extremely long computation time to break, for example. Unauthorized users should not be able to read privileged images.
2. Encryption and decryption should be fast enough not to degrade system performance. The algorithms for encryption and decryption must be simple enough to be done by users with a personal computer.
3. The security mechanism should be as widespread as possible. It must be widely acceptable to design a cryptosystem like a commercial product.
4. The security mechanism should be flexible.
5. There should not be a large expansion of the encrypted image data.

4 Research issues of An Image Cryptosystem

According to the analyses stated in Section 2, there are forth research issues on image cryptosystems as follows:

The first issue is to encrypt the image data using the same method as for text data. Images are usually represented as 2D arrays. They should be converted into 1D arrays before enciphering. Various encryption techniques can be used and applied on the 1D lists such as in [3-5]. Since the image is large, it is inefficient to encrypt or decrypt the picture directly. Applying compression techniques to images and then encrypting the compressed images is also a way to use standard text encryption algorithms.

The second issue is to use the special features of images. The main feature of an image is that it allows a bit of distortion. Therefore, picture data can be compressed before transmitting, and be lossy decompressed with a small distortion after receiving the image compression. There are many lossy compression techniques for images. This issue is to encrypt the compressed image using the same method as for the text data. Since the size of the compressed image is usually larger than that of text data, it is also invalid to reduce the size of the picture by image compression before enciphering. Chang and Liou's image cryptosystem [6] is in this form.

The third issue depend on the use of vector quantization (VQ), cryptography, number theorems and compression. The auxiliary data is encrypted only by some encryption algorithms such as in [7]. Since the size of the auxiliary data is usually less than that of the compressed image, the time complexity for enciphering auxiliary data is less than that of the above two issues.

The forth issue depends on chaotic maps that are considered as candidate for design of chaos based encryption techniques [8-13] which are good for practical use as these techniques provide a good combination of speed, high security, complexity, reasonable computational overheads and computational power etc. The proposed ECBFSC belongs to this issue.

5 Chaos and Cryptography

The close relationship between chaos and cryptography makes chaos based cryptographic algorithms as a natural candidate for secure communication and cryptography

chaos based encryption techniques are considered good for practical use as these techniques provide a good combination of speed, high security, complexity, reasonable computational overheads and computational power etc.

5.1 Characteristics of the chaotic maps

The characteristics of the chaotic maps have attracted the attention of cryptographers since it has many

fundamental properties such as ergodicity, sensitivity to initial condition and system parameter, and mixing property, etc [15-16].
Most properties are related to some requirements such as mixing and diffusion in the sense of cryptography. Therefore, chaotic cryptosystems have more useful and practical applications.

5.2 The logistic map and its analysis

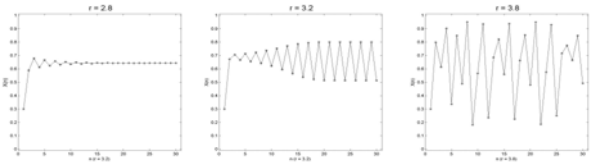
One of the simplest chaos functions that have been studied recently for cryptography applications is the logistic map. The logistic map function is expressed as:

$$X_{n+1} = rX_n(1-X_n) \quad (1)$$

Where X_n takes values in the interval [0,1]. It is one of the simplest models that present chaotic behavior [17].

The parameter r can be divided into three segments, which can be examined by experiments on following conditions: $X_0 = 0.3$. When $r \in [0,3]$ as shown in Fig. 1(a), the calculation results come to the same value after several iterations without any chaotic behaviour. When $r \in [3,3.57]$, the phase space concludes several points only, as showed in Fig. 1(b), the system appears periodicity. While $r \in [3.57,4]$, it becomes a chaotic system with periodicity disappeared as shown in Fig. 1(c). So we can draw the following conclusions:

(1) When $r \in [0,3.57]$, the points concentrate on several values and could not be used for image cryptosystem.
(2) For $r \in [3.57,4]$, the logistic map exhibits chaotic behavior, and hence the property of sensitive dependence [18]. So it can be used for image cryptosystem.



a) iteration property when r = 2.8 b) iteration property when r = 3.2 c) iteration property when r = 3.8

Figure 1: Analysis of Logistic Map

6 The Proposed ECBFSC

In this section, we discuss the step by step procedure of the proposed ECBFSC for encryption as well as decryption process. The proposed ECBFSC consists of two main modules, the encryption and decryption module.

6.1 The encryption module

An overview of ECBFSC encryption module is depicted in Fig. 2. the proposed ECBFSC is a simple block cipher with block size of 8-bit and 256-bit secret key. The key is used to generate a pad that is then merged with the plaintext a byte at a time.

1. For the encryption/decryption, we divide plaintext/ciphertext into blocks of 8-bits. Plaintext and ciphertext of i blocks can be represented as

$$P = P_1P_2P_3P_4.....P_i \quad (2)$$

$$C = C_1C_2C_3C_4.....C_i \quad (3)$$

2. The proposed image encryption process utilizes an external secret key of 256-bit long. Further, the secret key is divided into blocks of 8-bit each, referred as session keys.

$$K = K_1K_2K_3K_4.....K_{64} \text{ (in hexadecimal)} \quad (4)$$

here, K_i's are the alphanumeric characters (0–9 and A–F) and each group of two alphanumeric characters represents a session key. Alternatively, the secret key can be represented in ASCII mode as

$$K = K_1K_2K_3K_4.....K_{32} \text{ (in ASCII)} \quad (5)$$

here, each K_i represents one 8-bit block of the secret key i.e. session key.

3. The initial condition (X_0) for the chaotic map and the initial code C_0 are generated from the session keys as

$$R = \sum_{i=1}^{32}(M1[K_i]) \quad (6)$$

$$X_0 = R - \lfloor R \rfloor \quad (7)$$

$$C_0 = \left[\sum_{i=1}^{32}(K_i)\right] \bmod 256 \quad (8)$$

here K_i, $\lfloor\ \rfloor$, and M1 are, respectively, the decimal equivalent of the ith session key, the floor function, and mapping from the session, key space, all integers between 0 and 255, into the domain of the logistic map, all real numbers in the interval [0,1].

4. Read a byte from the image file (that represent a block of 8-bits) and load it as plainimage pixel P_i.
5. Encryption of each plainimage pixel P_i to produce its corresponding cipherimage pixel C_i can be expressed mathematically as:

$$C_i = \left(P_i + M2\left[\sum_{i=1}^{\#_i} rX_i(1-X_i)\right]\right) \bmod 256 \quad (9)$$

Where X_i represents the current input for logistic map and computed as:

$$X_i = M1[X_{i-1} + C_{i-1} + K_i] \quad (10)$$

$\#_i$ is the number of iteration of logistic map for its current input X_i and calculated as:

$$\#_i = K_{i+1} + C_{i-1} \quad (11)$$

And M2 maps the domain of the logistic map, [0,1], back into the interval [0,255].

6. Repeat steps 4-5 until the entire image file is exhausted.

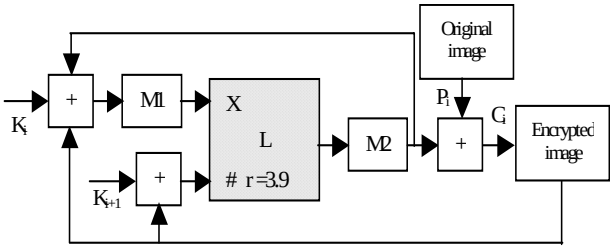


Figure 2: Diagram of Encryption Module

6.2 The decryption module

Decryption is very simple, the same pad is generated but this time un-merged with the ciphertext to retrieve the plaintext.

The diagram of ECBFSC decryption module is given in Fig. 3. The decryption module receives an encrypted image (cipherimage) and the 256-bit secret key and returns the original image (plainimage).

In particular, the decryption module works in the same way as the encryption module but now the output of the logistic map is subtracted from the corresponding cipherimage pixel C_i providing the plainimage pixel P_i . The output of the decryption module is the original image (plainimage).

Decryption of each cipherimage pixel C_i to produce its corresponding plainimage pixel P_i can be expressed mathematically as:

$$P_i = \left(C_i - M2\left[\sum_{i=1}^{\#_i} rX_i(1-X_i) \right] \right) \bmod 256 \qquad (12)$$

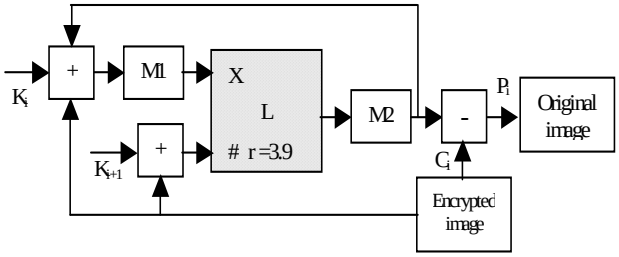


Figure 3: Diagram of Decryption Module

7 Design Principles of the Proposed ECBFSC

The basic concept is that the encryption of each part of the plainimage depends not only on the key, but also on the previous cipherimage.

The use of feedback mechanism has two desirable benefits. The first benefit is that there can be no simple periodicity in the encrypted image (cipherimage) because the encryption of each plainimage pixel depends not only on the encryption key, but also on the previous cipherimage pixel. The second benefit is that any changes in the plainimage are cascaded forward throughout the cipherimage, which means that two almost identical plainimages will encrypt to completely different cipherimages. This sensitivity to the plainimage is also a plus to the security of the proposed ECBFSC.

The proposed ECBFSC makes heavy use of data-dependent essentials. This appears for the current input of logistic map, which is data-dependent since it is computed as a function of the current session key K_i, previous computed cipher pixel C_{i-1} and previous logistic output. Also, the number of iterations # for the chaotic logistic map is data-dependent since it is computed as a function of current session key K_{i+1} and previous computed cipher pixel C_{i-1}.

As we encrypt each new block, i, the counter used to keep track of the current session key, is incremented. The output of the logistic map is then merged with the plaintext to give the ciphertext.

8 Test, Verification and Efficiency of ECBFSC

Results of some experiments are given to prove its efficiency of application to digital images.

We use the gray-scale images--Lena and Eiffel Tower, each of size 256 x 256, gray-scale (0-255) as the original images (plainimages) and the secret key "12345789012345678912345678901 2" (in ASCII) is used for encryption whose size is 256-bit. The encrypted images are depicted in Figs. 4(b)-5(b). As shown, the encrypted images (cipherimages) regions are totally invisible.

The decryption method takes as input the encrypted image (cipherimage), together with the same secret key "12345789012345678912345678901 23" (in ASCII). The decrypted images are shown in Figs. 4(c)-5(c).

The visual inspection of Figs. (4-5) shows the possibility of applying the proposed ECBFSC successfully in both encryption and decryption. Also, it reveals its effectiveness in hiding the information contained in them.

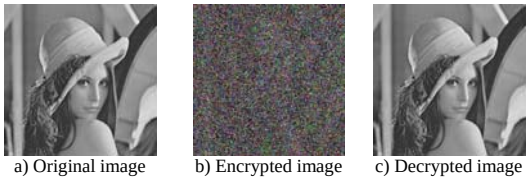

a) Original image b) Encrypted image c) Decrypted image

Figure 4: Application of ECBFSC to Lena Plainimage/Cipherimage


a) Original image b) Encrypted image c) Decrypted image

Figure 5: Application of ECBFSC to Eiffel Tower Plainimage/Cipherimage

9 Security Analysis and Test Results

A good encryption procedure should be robust against all kinds of cryptanalytic, statistical and brute-force attacks. In this section, we discuss the security

analysis of the proposed ECBFSC such as key space analysis, statistical analysis, and sensitivity analysis with respect to the key and plainimage to prove that the proposed cryptosystem is secure against the most common attacks [19-22].

9.1 Key space analysis

For a secure image cryptosystem, the key space should be large enough to make the brute force attack infeasible. The proposed ECBFSC has 2^{256} different combinations of the secret key. An image cipher with such a long key space is sufficient for reliable practical use.

In the proposed ECBFSC, a chaotic logistic map is employed which is sensitive on the initial condition. The initial condition for logistic map is calculated from the secret key.

Additionally the number of iterations supported by the logistic map module is between 0 and 767, as cipher pixels take values in the interval [0,512] and the session keys take values in the interval [0,255].

9.2 Statistical analysis

It is well known that many ciphers have been successfully analyzed with the help of statistical analysis and several statistical attacks have been devised on them. Therefore, an ideal cipher should be robust against any statistical attack. To prove the robustness of the proposed ECBFSC, we have performed statistical analysis by calculating the histograms and the correlations of two adjacent pixels in the plainimage/cipherimage.

9.2.1 Histograms analysis

To prevent the leakage of information to an opponent, it is also advantageous if the cipherimage bears little or no statistical similarity to the plainimage. An image-histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each color intensity level. We have calculated and analyzed the histograms of the several encrypted images as well as its original images that have widely different content.

One typical example among them is shown in Fig. 6(b). The histogram of a plainimage contains large spikes. These spikes correspond to color values that appear more often in the plainimage.

The histogram of the cipherimage as shown in Fig. 6(d), is more uniform, significantly different from that of the original image, and bears no statistical resemblance to the plainimage. It is clear that the histogram of the encrypted image is fairly uniform and significantly different from the respective histograms of the original image and hence does not provide any clue to employ any statistical attack on the proposed image encryption procedure.

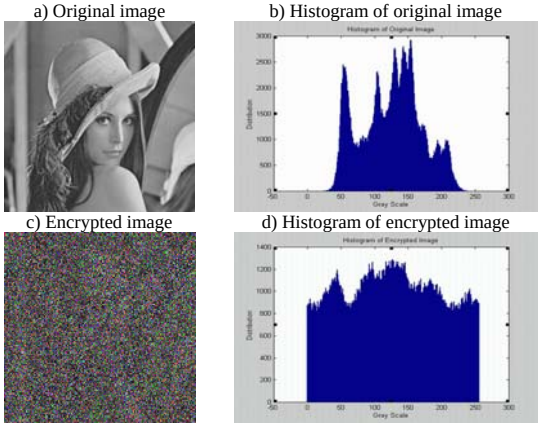


Figure 6: Histograms of the plainimage and the cipherimage

9.2.2 Correlation coefficient analysis

In addition to the histogram analysis, we have also analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels and two diagonally adjacent pixels in plainimage/cipherimage respectively. The procedure is as follows: First, randomly select 1000 pairs of two adjacent pixels from an image. Then, calculate their correlation coefficient using the following two formulas:

$$\mathrm{cov}(x,y) = E(x - E(x))(y - E(y)), \tag{13}$$

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{14}$$

Where x and y are the values of two adjacent pixels in the image. In numerical computations, the following discrete formulas were used:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{15}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2, \tag{16}$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)), \tag{17}$$

Fig. 7 shows the correlation distribution of two horizontally adjacent pixels in plainimage/cipherimage for the proposed ECBFSC. The correlation coefficients are 0.9905 and 0.0308 respectively for both plainimage/cipherimage. Similar results for diagonal and vertical directions are obtained as shown in Table 1. It is clear from the Fig. 7 and Table 1 that there is negligible correlation between the two adjacent pixels in the cipherimage. However, the two adjacent pixels in the plainimage are highly correlated.

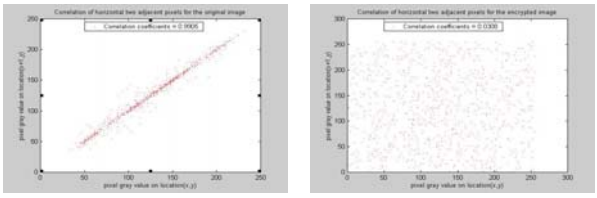


Figure 7: Two horizontally adjacent pixels Correlation in plainimage/cipherimage

Direction of Adjacent pixels	Plainimage	Cipherimage
Horizontal	0.9905	0.0308
Vertical	0.9787	0.0304
Diagonal	0.9695	0.0317

Table 1: Correlation coefficients in plainimage/ cipherimage

9.3 Sensitivity analysis

An ideal image encryption procedure should be sensitive with respect to both the secret key and plainimage. The change of a single bit in either the secret key or plainimage should produce a completely different encrypted image. To prove the robustness of the proposed ECBFSC, we will perform sensitivity analysis with respect to both key and plainimage.

9.3.1 Key sensitivity analysis

High key sensitivity is required by secure image cryptosystems, which means that the cipherimage cannot be decrypted correctly although there is only a slight difference between encryption or decryption keys. This guarantees the security of the proposed ECBFSC against brute-force attacks to some extent.

For testing the key sensitivity of the proposed image encryption procedure, we have performed the following steps:

(a) An original image in Fig. 8(a) is encrypted by using the secret key "12345789012345678912345678901 2" (in ASCII) and the resultant image is referred as encrypted image A as shown in Fig. 8(b).

(b) The same original image is encrypted by making the slight modification in the secret key i.e. "22345789012345678912345678901 2" (in ASCII) (the most significant bit is changed in the secret key) and the resultant image is referred as encrypted image B as shown in Fig. 8(c).

(c) Again, the same original image is encrypted by making the slight modification in the secret key i.e. secret key "12345789012345678912345678901 3" (in ASCII) (the least significant bit is changed in the secret key) and the resultant image is referred as encrypted image C as shown in Fig. 8(d).

(d) Finally, the three encrypted images A, B and C are compared.

In Fig. 8, we have shown the original image as well as the three encrypted images produced in the aforesaid steps. It is not easy to compare the encrypted images by simply observing these images. So for comparison, we have calculated the correlation between the corresponding pixels of the three encrypted images. For this calculation, we have used the same formula as given in Eq. (14) except that in this case x and y are the values of corresponding pixels in the two encrypted images to be compared. In Table 2, we have given the results of the correlation coefficients between the corresponding pixels of the three encrypted images A, B and C. It is clear from the table that no correlation exists among three encrypted images even though these have been produced by using slightly different secret keys.

Key sensitivity analysis shows that changing one bit in encryption key will result in a completely different cipherimage by more than 99% in terms of pixel gray scale values.

Moreover, in Fig. 9, we have shown the results of some attempts to decrypt an encrypted image with slightly different secret keys than the one used for the encryption of the original image. Particularly, in Fig. 9(a) and Fig. 9(b) respectively, the original image and the encrypted image produced using the secret key "12345789012345678912345678901 2" (in ASCII) are shown whereas in Fig. 9(c) and Fig. 9(d) respectively, the images after the decryption of the encrypted image (shown in Fig. 9(b)) with the secret keys "12345789012345678912345678901 2" (in ASCII) and "12345789012345678912345678901 1" (in ASCII). It is clear that the decryption with a slightly different key fails completely and hence the proposed image encryption procedure is highly key sensitive.

Figure 8: Key sensitive test result 1 with ECBFSC

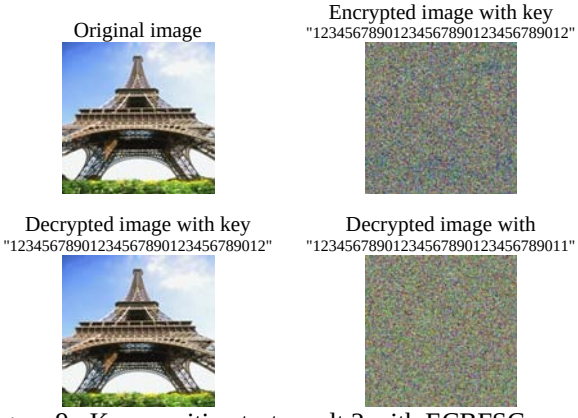


Figure 9: Key sensitive test result 2 with ECBFSC

Image 1	Image 2	Correlation coefficient
Encrypted image A Fig. 8(b)	Encrypted image B Fig. 8(c)	0.0326
Encrypted image B Fig. 8(c)	Encrypted image C Fig. 8(d)	0.0370
Encrypted image C Fig. 8(d)	Encrypted image A Fig. 8(b)	0.0369

Table 2: Correlation coefficients between the corresponding pixels of the three different encrypted images obtained by using slightly different secret key of an image shown in Fig. 8.

9.3.2 Plainimage sensitivity analysis

A desirable property for the proposed ECBFSC is that it is highly sensitive to small change in the plainimage (single bit change in plainimage).

To test the influence of one-pixel change on the plainimage, encrypted by the proposed ECBFSC, two common measures may be used: Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI). Let two ciphered images, whose corresponding plainimages have only one pixel difference, be denoted by C1 and C2. Label the gray-scale values of the pixels at grid (i,j) in C1 and C2 by C1(i,j) and C2(i,j), respectively. Define a bipolar array, D, with the same size as images C1 and C2. Then, D(i,j) is determined by C1(i,j) and C2(i,j), namely, if C1(i,j) = C2(i,j) then D(i,j) = 1; otherwise, D(i,j) = 0.

The NPCR is defined as

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%, \qquad (18)$$

Where W and H are the width and height of C1 or C2. The NPCR measures the percentage of different pixel numbers between plainimage and cipherimage.

The UACI is defined as

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{C_1(i,j) - C_2(i,j)}{255}\right] \times 100\% \qquad (19)$$

Which measures the average intensity of differences between the two images. One performed test is on the one-pixel change influence on a 256 grey-level Lena image of size 256 x 256.

With respect to NPCR estimation. NPCR is obtained using the proposed ECBFSC and estimated to be over 99.62% showing thereby that the encryption scheme is very sensitive with respect to small changes in the plainimage.

With respect to UACI estimation. UACI is calculated to be %80.51 indicating that the rate of influence due to one pixel change is very large. Generally, these obtained results for NPCR and UACI show that the proposed ECBFSC is very sensitive with respect plainimage (plainimages have only one pixel difference).

10 Performance Evaluation

Apart from the security consideration, some other issues on image encryption are also important. These include the running speed for real-time image encryption/decryption. The simulator for the proposed ECBFSC is implemented using the compiler in Borland C++ Development Suite 5.0. Performance was measured on a 2.4 GHz Pentium IV with 256 Mbytes of RAM running Windows XP. In addition, to improve the accuracy of our timing measurements, each set of the timing tests shown in Table 3 was executed 10 times, and we report the average of the times thereby obtained.

Image size (in pixels)	Colors	Encryption in Sec.	Decryption in Sec.
256 x 256	2	< 0.0010	< 0.0010
256 x 256	16	< 0.0010	< 0.0010
256 x 256	256	0.0030	0.0040
256 x 256	16777216	0.0267	0.0360
512 x 512	2	< 0.0010	< 0.0010
512 x 512	16	0.0090	0.0108
512 x 512	256	0.0305	0.0358
512 x 512	16777216	0.1108	0.1306
1024 x 1024	2	0.0150	0.0163
1024 x 1024	16	0.0716	0.0832
1024 x 1024	256	0.1618	0.1744
1024 x 1024	16777216	0.4690	0.587
2048 x 2048	2	0.0666	0.0954
2048 x 2048	16	0.2942	0.3802
2048 x 2048	256	0.6322	0.7604
2048 x 2048	16777216	1.6085	1.8100

Table 3: Enciphering/deciphering speed test results of the proposed ECBFSC

Table 3 summarizes the encryption/decryption speeds for the proposed ECBFSC on images of different sizes. The results emphasis that proposed ECBFSC. Simulation results show that the average encryption/decryption speed is 7.46 MB/Sec for encryption and 6.63 MB/Sec for decryption. The peak speed can reach up to 7.6 MB/Sec for encryption and 6.7 MB/Sec for decryption.

11 Conclusion

In this paper, a new way of image encryption scheme have been proposed which utilizes a chaos-based feedback cryptographic scheme using the logistic map and an external secret key of 256-bit.

The robustness of the proposed ECBFSC is further reinforced by a feedback mechanism, which leads the cipher to a cyclic behavior so that the encryption of each plain pixel depends on the key, the value of the previous cipher pixel and the output of the logistic map.

We have carried out key space analysis, statistical analysis, and key sensitivity analysis to demonstrate the security of the new image encryption procedure. According to the results of our security analysis, we conclude that the proposed ECBFSC is expected to be useful for real-time image encryption and transmission applications.

Furthermore, we have also discussed the characteristics of image cryptosystems, chaos and cryptography including characteristics and analysis of chaotic logistic map, and research issues related to image cryptosystems.

12 References

[1] W. Stallings., "Cryptography and Network Security: Principles and Practice," Prentice-Hall, New Jersey, 1999.

[2] Bruce Schneier, "Applied Cryptography – Protocols, algorithms, and source code in C," John Wiley & Sons, Inc., New York, second edition, 1996.

[3] N. Bourbakis and C. Alexopoulos , Picture data encryption using SCAN patterns. Pattern Recognition 25 6 (1992), pp. 567–581.
[4] Alexopoulos, C., 1989. SCAN, A language for 2-D sequential data accessing. Ph.D. Thesis, University of Patras, Greece.
[5] C.J. Kuo , Novel image encryption technique and its application in progressive transmission. J. Electron. Imaging 24 (1993), pp. 345–351.
[6] Chang, H.K., Liou, J.L., 1994. An image encryption scheme based on quadtree compression scheme. In: Proceedings of the International Computer Symposium, Taiwan, pp. 230–237.
[7] Chin-Chen Chang, Min-Shian Hwang, Tung-Shou Chen, “A new encryption algorithm for image cryptosystems”, The Journal of Systems and Software 58 (2001), 83-91
[8] Fridrich Jiri, Symmetric ciphers based on two dimensional chaotic maps, Int. J. Bifurcat Chaos 8 (1998) (6), pp. 1259–1284.
[9] J. Scharinger, Fast encryption of image data using chaotic Kolmogrov flow, J. Electronic Eng 7 (1998) (2), pp. 318–325.
[10] J.C. Yen, J.I. Guo, A new image encryption algorithm and its VLSI architecture, in: Proceedings of the IEEE workshop signal processing systems, 1999, pp. 430–437.
[11] J.C. Yen, J.I. Guo, A new chaotic key based design for image encryption and decryption, Proceedings of the IEEE International Symposium Circuits and Systems, vol. 4, 2000, pp. 49–52.
[12] S. Li, X. Zheng, X. Mou, Y. Cai, Chaotic encryption scheme for real time digital video, Proceedings of the SPIE on electronic imaging, San Jose, CA, USA, 2002.
[13] G. Chen, Y. Mao and C.K. Chui, A symmetric image encryption based on 3D chaotic maps, Chaos Solitons Fractals 21 (2004), pp. 749–761.
[14] W. Diffie and M.E. Hellman , New directions in cryptography. IEEE Trans. Inf. Theory 22 (1976), pp. 644–654.
[15] M. S. Baptista, "Cryptography with chaos". Phys. Lett. A, vol.240, pp.50-54,1998.
[16] G. Jakimoski and L. Kocarev, "Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps," IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 48, no. 2, February 2001.
[17] R. Devaney, "An Introduction to Chaotic Dynamical Systems," 2nd ed. Redwood City, CA: Addison-Wesley, 1989.
[18] Steven Henry Strogatz, "Nonlinear dynamics and chaos: With applications to physics, biologym chemistry, and engineering," first ed., Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.
[19] Shujun Li, Guanrong Chen and Xuan Zheng, "Chaos-based encryption for digital images and videos," chapter 4 in Multimedia Security Handbook, February 2004.
[20] Yaobin Mao and Guanrong Chen, "Chaos-based image encryption," in Eduardo Bayro-Corrochano, editor, Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neural Computing and Robotics. Springer-Verlag, Heidelberg, April 2004.
[21] Yaobin Mao, Guanrong Chen, and Charles K. Chui, "A novel fast image encryption scheme based on 3D chaotic Baker maps," Int. J. Bifurcation and Chaos in June 2003.
[22] Yaobin Mao, Guanrong Chen, and Shiguo Lian, "A symmetric image encryption scheme based on 3D chaotic Cat maps," Chaos, Solitons and Fractals 21, pages 749-761, 2004.

Novel Broadcasting Algorithm of the Complete Recursive Network

Jywe-Fei Fang
National Taichung University, Department of Digital Content and Technology
140 Ming-Shen Rd., Taichung 403, Taiwan, R. O. C.
E-mail1: jffang@mail.ntcu.edu.tw

Wen-Yew Liang
National Taipei University of Technology, Department of Computer Science and Information Engineering

Hong-Ren Chen
National Taichung University, Department of Digital Content and Technology

Ka-Lok Ng
Asia University, Department of Biotechnology and Bioinformatics

Keywords: complete WK-Recursive networks, interconnection networks, broadcasting algorithms

Received: November 20, 2005

The interconnection network considered in this paper is the complete WK-Recursive network that demonstrates many attractive properties, such as high degree of regularity, symmetry and efficient communication. Chen and Duh have proposed a distributed stack-base broadcasting algorithm for the complete WK-Recursive networks [Networks, 24 (1994) 303-317]. To perform this algorithm, a stack of O(log N) elements, where N is the number of nodes, to keep the labels of links is included in each message. Moreover, as a node receives the message, a series of O(log N) pop and push operations on the stack is required. In this paper, we present a novel broadcasting algorithm for the complete WK-Recursive network, which is much simpler and requires only constant data included in each message and constant time to determine the neighbors to forward the message.

Povzetek: Opisan je nov algoritem razpošiljanja v rekurzivni mreži.

1 Introduction

In massively parallel MIMD systems, the topology plays a crucial role in issues such as communication performance, hardware cost, potentialities for efficient applications and fault tolerance capabilities. A topology named *complete WK-Recursive network* has been proposed by Vecchia and Sanges under CAPRI(Concurrent Architecture and Programming environment for highly Integrated systems) project supported by the Strategic Program on Parallel Computing of the National Research Council of Italy [11]. A complete WK-Recursive network with amplitude W and level L is denoted by WK(W, L), where $W \geq 2$. The topology has many attractive properties, such as high degree of regularity, symmetry and efficient communication. Particularly, for any specified number of degree, it can be expanded to arbitrary size level without reconfiguring the links. The complete WK-Recursive networks have received considerable attention. Researchers have devoted themselves to various issues of complete WK-Recursive networks. A VLSI implementation and a simple routing algorithm of complete WK-Recursive networks have been developed [11]. Verdoscia and Vaccaro proposed an adaptive routing algorithm on the complete WK-Recursive networks [12]. The topological properties of complete WK-Recursive networks are studied [7]. The subnetwork allocation of complete WK-Recursive networks has been discussed [4, 13]. On the other hand, various variations of the complete WK-Recursive networks have been proposed. *Three-dimensional WK-Recursive networks* are defined; and a performance comparison of standard WK-Recursive networks and three-dimensional WK-Recursive networks is given [3]. *Hierarchical WK-Recursive networks* and *Pyramid WK-Recursive networks* have been proposed and studied [5]. The *incomplete WK-Recursive networks* have been defined; and the shortest routing algorithm has been devised [9].

It is widely recognized that interprocessor communication is one of the most important issues for interconnection networks because the communication problem is the key issue to many parallel algorithms [8]. *Broadcasting* which is a primitive communication problem is to distribute the same message from a source node to all other nodes without redundancy. The common approach to implement broadcasting algorithm is to embed the broadcasting tree that is a spanning tree with the source node as the root [6]. Numerous applications employ a broadcasting algorithm as a basic function. For example, it is applied in the applications such as matrix operations (e.g., matrix multiplication, factorization, inversion, transposition), database

operations (e.g., polling, master-slave operation), transitive closure algorithms, distributed fault diagnosis, distributed agreement and distributed election [6, 14]. The interconnection network must facilitate efficient broadcasting algorithm to achieve high performance during execution of the various applications. A broadcasting algorithm is *distributed* if it is distributed among all the nodes in the interconnection networks; and each node is responsible for deciding the neighbors to forward the broadcasting message by its own decisions except a centralized controlling node.

Chen and Duh have proposed a distributed stack-base broadcasting algorithm for the complete WK-Recursive networks [1]. Moreover, the algorithm has been generalized for other relative networks [2]. To perform this algorithm, a stack of length L+1, which is used to keep the labels of links, is included in the broadcasting message. It is forbidden to further transmit the message through the links whose labels appear in the stack. Initially, the source node pushes the label L into the stack and transmits the message through all its incident links except the free link. Each node, after it has received the message through a link labeled i, where $1 \le i \le L$-1, performs the following: (1) pop the elements of the stack until the current top element is greater than i; (2) push i into the stack and (3) transmits the message through the links whose labels do not appear in the stack. Clearly, it requires L operations to decide the neighbors to transmit at most, and L is O(log N) where N is the number of nodes. Recall that the length of the stack is also O(L) = O(log N). Thus the distributed stack-base broadcasting algorithm requires O(log N) element in the message and O(log N) time to decide the neighbors to transmit. Because the broadcasting is a primitive problem on the interconnection networks, its performance reveals particular importance. For example, the distributed s broadcasting algorithm on the well-known hypercube requires only constant time to determine the neighbors to forward the message [8]. In this paper, we present a novel broadcasting algorithm for the complete WK-Recursive networks. Our algorithm which is much simpler requires only constant data included in each message and constant time to determine the neighbors to forward the message.

2 Notations and background

A *complete graph* with n nodes, denoted by K_n, is a graph in which every two distinct nodes are adjacent. A WK(W, L), where $W \ge 2$, can be recursively constructed as: WK(W, 0) is a node with W free links that are not incident to other nodes yet. WK(W, 1) is a K_W in which each node has one free link and W-1 links that are used for connecting to other nodes. Clearly, WK(W, 1) has W nodes and W free links. WK(W, C) consists of W copies of WK(W, C-1) as supernodes and the W supernodes are connected as a K_W, where $2 \le C \le L$. By induction, it is easy to see that WK(W, L) has W^L nodes and W free links. Consequently, for any specified number of degree W, the complete WK-Recursive networks can be expanded to arbitrary level L without reconfiguring the links. In Figure 1 and Figure 2, the structures of the WK(4, 0), the WK(4, 1), the WK(4, 2) and the WK(4, 3) are shown.

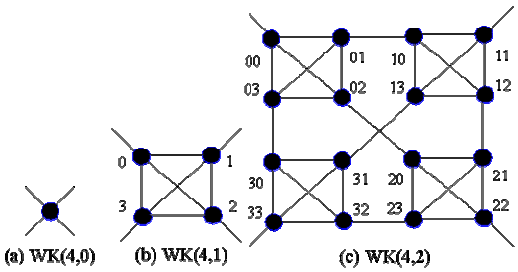


Figure 1: The structures of the WK(4, 0), the WK(4, 1) and the WK(4, 2).

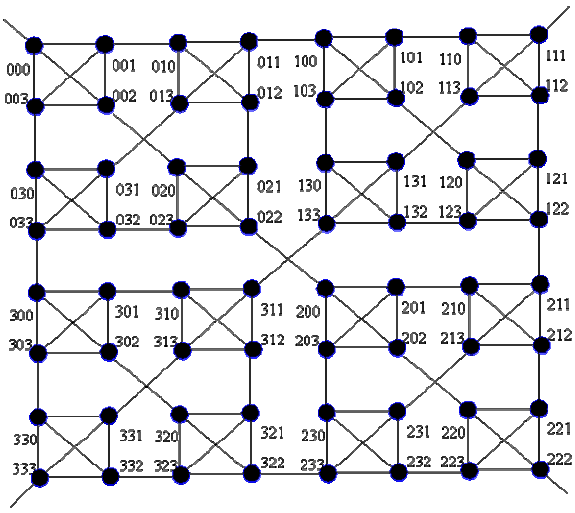


Figure 2: The structures of the WK(4, 3).

In order to utilize the full bandwidth, we assume that each node can transmit and receive messages along different incident links simultaneously (i.e., *all port* assumption). This assumption is quite reasonable because the time required for local computation is negligible. In fact, if each node can transmit and receive messages along only one incident link at any one time (i.e., *one port* assumption), the bandwidth of any interconnection network is the same as the bandwidth of a ring [8].

The rest of this paper is organized as follows. In Section 2, we present some notations and background that will be used throughout this paper. In Section 3, we propose our broadcasting algorithm for the complete WK-Recursive networks. The paper is concluded in Section 4.

The following addressing scheme for WK(W, L) is described in [10]. After fixing an origin and an orientation (i.e., clockwise or counterclockwise), each node within a WK(W, 1) subnetwork is labeled with an index digit d_1 from 0 to W-1. Similarly, each WK(W, C-1) subnetwork within a WK(W, C) subnetwork is

labelled with an index d_C from 0 to W-1, where $2 \le C \le L$. Hence, each node of WK(W, L) is labeled with an unique address $d_L d_{L-1} \ldots d_2 d_1$ as illustrated in Figure 1 and Figure 2. Likewise, a subnetwork of WK(W, L) can be represented by a string $d_L d_{L-1} \ldots d_{C+1}(*)^C$ over set $\{0,1,\ldots,W\text{-}1\} \cup \{*\}$, where * is a "don't care" symbol and $(*)^C$ represents C consecutive *'s. For example, in WK(4, 3), 0** is the subnetwork $\{0 d_2 d_1 \mid 0 \le d_2 \le 3$ and $0 \le d_1 \le 3\}$.

For a subnetwork $d_L d_{L-1} \ldots d_{C+1}(*)^C$ in WK(W, L), a node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$ is called a *corner node* of $d_L d_{L-1} \ldots d_{C+1}(*)^C$. For example, in WK(4, 3), 000, 011, 022 and 033 are corner nodes of 0**. Specifically, the node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$ is called the d_C-*corner* of $d_L d_{L-1} \ldots d_{C+1}(*)^C$. For example, in WK(4, 3), the nodes 000, 011, 022 and 033 are 0-corner, 1-corner, 2-corner and 3-corner of 0**, respectively. Note that node 000 is also the 0-corner of 00* and WK(4, 3).

Definition 1. The *corner identifier* of a node $v = d_L d_{L-1} \ldots d_2 d_1$, denoted by *cor-id*($v$), is defined as d_1.

Definition 2. The *corner level* of a node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, where $d_{C+1} \ne d_C$, denoted by *cor-level*(v) is defined as C.

For example, the corner identifiers of nodes 000, 011, 022 and 033 are 0, 1, 2 and 3, respectively. In fact, a node is the *d-corner* of a subnetwork if and only if the corner identifier of the node is d. The corner levels of nodes 000, 011, 012 are 3, 2, and 1, respectively. Note that each node has a corner identifier between 0 and W-1 and a corner level between 1 and L in WK(W, L).

In this paper, a link within a WK(W, 1) subnetwork is called an *inner-cluster link*.

Definition 3. The inner-cluster links of node $d_L d_{L-1} \ldots d_2 d_1$ are defined as $(d_L d_{L-1} \ldots d_2 d_1, d_L d_{L-1} \ldots d_2 h)$, where $0 \le h \le W$-1 and $d_1 \ne h$.

For example, in WK(4, 3), (002, 000), (002, 001), (002, 003) are inner-cluster links of node 002. Clearly, each node has W-1 inner-cluster links in WK(W, L). A link connecting two WK(W, C) subnetworks, where $1 \le C \le L$-1, is called an *inter-cluster link* and specifically a *C-level link*.

Definition 4. The C-level inter-cluster link of node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, where $d_{C+1} \ne d_C$, is defined as $(d_L d_{L-1} \ldots d_{C+1}(d_C)^C,\ d_L d_{L-1} \ldots d_C (d_{C+1})^C)$. The *inter-cluster neighbor* of a node $d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, where $d_{C+1} \ne d_C$, is $d_L d_{L-1} \ldots d_C (d_{C+1})^C$. The *flipping corner identifier* of a node $v = d_L d_{L-1} \ldots d_{C+1}(d_C)^C$, denoted by *flip-cor-id*(v), is the corner identifier of inter-cluster neighbor of v.

For example, in WK(4, 3), (022, 200) is a 2-level link and (012, 021) is a 1-level link. Note that each node except the corner nodes $(d_L)^L$, where $0 \le d_L \le W$-1, has exactly one inter-cluster link in WK(W, L). Each corner node $(d_L)^L$ of WK(W, L) has no inter-cluster link but a free link. In fact, a node is incident to a C-level inter-cluster link, where $1 \le C \le L$-1, if and only if the corner level of the node is C. In this paper, a node v is said to be connected to a subnetwork U if there exists a node $u \in U$ such that v is adjacent to u. Nodes 133 and 311 are inter-cluster neighbors of each other. Thus, *flip-cor-id*(133) = 1 and *flip-cor-id*(311) = 3.

Definition 5. The *outline graph* of a WK(W, L), denoted by an OG(WK(W, L)), is to take each WK(W, 1) subnetwork as a supervertex.

Recall that a WK(W, L) can be constructed recursively. If each WK(W, 1) subnetwork of a WK(W, L) is taken as a supervertex, the WK(W, L) will be transformed to a WK(W, L-1). Moreover, each original level-1 inter-cluster link will be an inner-cluster link in the OG(WK(W, L)); and each original level-J inter-cluster link will be a level-(J-1) inter-cluster link in the OG(WK(W, L)), where L-1 $\ge J \ge$ 2. We have the following proposition.

Proposition 1. *An* OG(WK(W, L)) is a WK(W, L-1).

3 The broadcasting algorithm

In this section, we present a new broadcasting algorithm for the complete WK-Recursive networks. Suppose that each node is associated with its own node address, corner identifier, corner level and flipping corner identifier; and each node has kept the system size level L. This algorithm is based on the idea as follows: Let A be a broadcasting algorithm that works for WK(W, L). According to Proposition 1, the outline graph of WK(W, L+1) is WK(W, L). If we apply A to the outline graph of WK(W, L+1), a broadcasting algorithm between these WK(W, 1) subnetworks of WK(W, L+1) is obtained. When a node receives the broadcasting message from an inter-cluster link, it broadcasts the message to the other nodes in the same WK(W, 1) subnetwork. As a consequence, the message can be broadcast to each node of WK(W, L+1) correctly.

3.1 Corner Broadcasting Algorithm

First, we devise an algorithm, *corner broadcasting algorithm*, to deal with the case in which the source node is a corner node $(p)^L$, where $0 \le p \le W$-1, of WK(W, L). Observe that in each subnetwork $c(*)^{L-1}$, where $0 \le c \le W$-1 and $c \ne p$, the node connected to $p(*)^{L-1}$ is its p-corner = $c(p)^{L-1}$. For example, in subnetworks 1*, 2* and 3* of WK(4, 2), the nodes connected to 0* are nodes 10, 20 and 30, respectively; in subnetworks 1**, 2** and 3** of WK(4, 3), the nodes connected to 0** are nodes 100, 200 and 300, respectively. With the aid of this observation, a broadcasting algorithm for source node = $(p)^L$ (i.e., corner broadcasting algorithm) can be developed.

The broadcasting message is included by a label *source-corner-identifier* that records the corner identifier p of the source node. Initially, the source node $(p)^L$ disseminates the message labeled p to all other nodes in the same WK(W, 1) subnetwork by its inner-cluster links. Each node when it receives the message labeled p performs by the following conditions:

(1.1) if it receives the message from its inter-cluster link, broadcasts this message to all other nodes in the same WK(W, 1) subnetwork by its inner-cluster links.

(1.2) if it *receives* the message from its inner-cluster link,

(1.2.1) if *its* corner level is L, terminates transmitting.

(1.2.2) if its corner level is not L,

(1.2.2.1) if its flipping corner identifier is p, transmits the message to its inter-cluster neighbor by its inter-cluster link.

(1.2.2.2) if its flipping corner identifier is not p, terminates transmitting.

It is clearly that this algorithm works correctly for WK(W, 1) and WK(W, 2). For example, supposed that node 11 is the source node of broadcasting in WK(4, 2). Initially, node 11 broadcasts the message labelled 1 to nodes 10, 12 and 13. Then, according to Condition (1.2.2.1), the nodes 10, 12 and 13 transmit the message labelled 1 to nodes 01, 21 and 31, respectively. According to Condition (1.1), the nodes 01, 21 and 31 broadcast the message labelled 1 to all other nodes in 0*, 2* and 3*, respectively. Because the corner level of the nodes 00, 22 and 33 is 2, according to Condition (1.2.1), they terminate transmitting. Because the corner identifier of the nodes 02, 03, 20, 23, 30 and 32 is not 1, according to Condition (1.2.2.2), they terminate transmitting and duplicate message between 0*, 2* and 3* can be avoided.

Theorem 1. Using corner broadcasting algorithm, starting from a corner node $(p)^L$, a message can be transmitted to each node of WK(W, L) exactly once with 2^L-1 time steps.

Proof. We will prove the lemma by induction on L.

Clearly, it is true for L =1, 2.

Hypothesis: Assume that it is true for $L = k$.

Induction Step: Suppose that the source node is $(p)^{k+1}$. By hypothesis, the message labelled p can be transmitted to each node of $p(*)^k$ exactly once with 2^k-1 time steps. Then, according to Condition (1.2.2.1), each c-corner of $p(*)^k$ (i.e., $p(c)^k$), where $0 \le c \le W$-1 and $c \ne p$, will transmit the message labelled p to their inter-cluster neighbors, $c(p)^k$, respectively; because the flipping corner identifier of $p(c)^k$ is p. By hypothesis, in each $c(*)^k$, starting from the $c(p)^k$, the message labeled p can be transmitted to each node of $c(*)^k$ exactly once with 2^k-1 time steps. Thus, total time steps for this broadcasting is $2(2^k\text{-}1)+1 = 2^{k+1}\text{-}1$. This extends the induction and completes the proof. Q. E. D.

3.2 General Broadcasting Algorithm

Based on the corner broadcasting algorithm, we propose a *general broadcasting algorithm* to deal with general cases of the broadcasting problem for the complete WK-Recursive networks. To express the idea of this algorithm, the node set of WK(W, L) is partitioned into subsets according to where the source node s resides in. We define S_i, where $1 \le i \le L$, to be {v| if v and s reside in the same WK(W, i) subnetwork but distinct WK(W, i-1) subnetworks}. Particularly, we define $S_0 = \{s\}$. It is easy to see that S_i, where $1 \le i \le L$, consists of W-1 copies of WK(W, i-1) subnetworks. For example, suppose that the source node is node 201 in WK(4, 3). S_0 = {201}, S_1 = {200, 202, 203}, S_2 = {21*, 22*, 23*}, S_3 = {0**, 1**, 3**}.

In stage i, where $1 \le i \le L$, the message has been broadcast to S_{i-1} (i.e., the WK(W, i-1) subnetwork where source node s resides in) in previous stage; and it will be broadcast to S_i (i.e., the other W-1 copies of WK(W, i-1) subnetworks of the WK(W, i) subnetwork where source node s resides in) in this stage. The general broadcasting algorithm is based on the idea as follows: in stage i, first, let the message be transmitted to a corner node of each WK(W, i-1) subnetwork. Second, in each WK(W, i-1) subnetwork, starting from the corner node, broadcasting the message to all other nodes in the same subnetwork like applying the corner broadcasting algorithm. For example, suppose that starting from node 201 in WK(4, 3). In stage 1, node 201 broadcasts the message to nodes 200, 202 and 203. In stage 2, nodes 201, 202 and 203 transmit the message to nodes 210, 220 and 230, respectively; and then they broadcast the message to all other nodes in 21*, 22* and 23*, respectively. In stage 3, nodes 200, 211 and 233 transmit the message to nodes 022, 122 and 322, respectively; and then they broadcast the message to all other nodes in 0**, 1** and 3** like applying the corner broadcasting algorithm, respectively.

The broadcasting message is included by a label (m, t), where m records the max level of link that it has ever passed and t records the corner identifier of the WK(W, i-1) subnetwork in each stage i, where $1 \le i \le L$. Clearly, in stage i, the m included in each message is i-1. Initially, the source node s broadcasts the message labelled (0, *cor-id*(s)) (i.e., m = 0 and t = *cor-id*(s)) to the other nodes in the same WK(W, 1) subnetwork (i.e., S_1); and if its inter-cluster link exists, transmits the message labelled (*cor-level*(s), *cor-id*(s)) to its inter-cluster neighbor. Each node v when it receives the message labelled (m, t) performs by the following conditions:

(2.1) If it receives the message from its inter-cluster link, it broadcasts the message labelled (m, t) to all other nodes in the same WK(W, 1) subnetwork by its inner-cluster links.

(2.2) If it receives the message from its inner-cluster link,

(2.2.1) if cor-level(v) = L, terminates transmitting.

(2.2.2) if L > *cor*-level(v) > m, transmits the message labelled (*cor-level*(v), *flip-cor-id*(v)) to its inter-cluster neighbor by its inter-cluster link.

(2.2.3) if cor-level(v) = m, terminates transmitting.

(2.2.4) if $m > cor$-level(v),

(2.2.4.1) if flip-cor-id(v) = t, transmits the message labeled (m, t) to its inter-cluster neighbor by its inter-cluster link.

(2.2.4.2) if flip-cor-id(v) ≠ t, terminates transmitting.

Condition (2.1) means that if a node of a WK(W, 1) subnetwork has received the message, all other nodes in the same WK(W, 1) subnetwork will receive the message. Condition (2.2) means that if a node receives the message from its inner-cluster link, only its inter-cluster neighbor is under consideration to transmit the message. Thus, no duplicate message transmitting in a WK(W, 1) subnetwork can be guaranteed. Recall that in stage i, where $1 \le i \le L$, the message has been broadcast to S_{i-1} (i.e., the WK(W, i-1) subnetwork where the source node s resides in) in previous stage. First, let the message be transmitted to the corner nodes of other WK(W, i-1) subnetworks. Clearly, m included in the message is i-1. Condition (2.2.4) means that if the m included in the message is less than i-1, it performs like corner broadcasting algorithm. Since there exists no link level greater than or equal to i-1 in a WK(W, i-1) subnetwork, Condition (2.2.4) guarantees that the message can be transmitted to each node in a WK(W, i-1) subnetwork exactly once. Observe that the link level of links which connect two WK(W, i-1) subnetworks is i-1. Recall that in stage i, the m included in the message is also i-1. Condition (2.2.3) means that it terminates transmitting if the corner level equals to m. Thus, duplicate message transmitted between the WK(W, i-1) subnetworks can be avoided. Condition (2.2.2) means that if the corner level, cor-$level$(v), is greater than m, the message labelled by (cor-$level$(v), $flip$-cor-id(v)) should be transmitted to its inter-cluster neighbor in stage cor-$level$(v)+1. Condition (2.2.1) means that it terminates transmitting when a corner node of WK(W, L) receives the message.

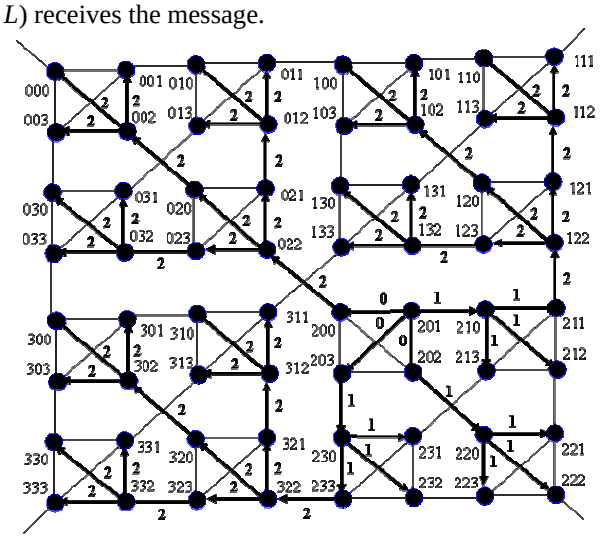


Figure 3: Starting from node 201, broadcasting in WK(4, 3). The label associated with an edge is the m included in the message passed through the edge.

As illustrated in Figure 3, we show an example of broadcasting by applying general broadcasting algorithm. Suppose that node 201 is the source node of broadcasting in WK(4, 3). For the readability of this paper, we describe the broadcasting stage by stage. Initially, in stage 1, node 201 broadcasts the message labelled (0, 1) to nodes 200, 202 and 203. In stage 2, according to Condition (2.2.2), because the corner level of the nodes 201, 202 and 203 is 1, they transmit the message labelled (1, 0) to nodes 210, 220 and 230. Then, according to Condition (2.1), the nodes 210, 220 and 230 broadcast the message labelled (1, 0) to all other nodes of 21*, 22* and 23*, respectively. The corner level of 212, 213, 221, 223, 231, 232 is 1. According to Condition (2.2.3), they terminate transmitting the message. Thus, transmitting duplicate messages between 21*, 22* and 23* is avoided. The corner level of 222 is 3. According to Condition (2.2.1), it terminates transmitting the message. In stage 3, because the corner level of nodes 200, 211 and 233 is 2, according to Condition (2.2.2), they transmit the message labelled (2, 2) to nodes 022, 122 and 322, respectively. In what follows, we only describe the broadcasting starting from 022 in 0**. According to Condition (2.1), the node 022 broadcasts the message labelled (2, 2) to other nodes in 02*. According to Condition (2.2.4.1), because the flip corner identifiers of nodes 020, 021 and 023 are 2, they transmit the message labelled (2, 2) to nodes 002, 012 and 032, respectively. According to Condition (2.1), 002, 012 and 032 transmit the message labelled by (2, 2) to all other nodes in 00*, 01* and 03*, respectively. The corner levels of 001, 003, 010, 013, 030, 031 are 1 and their flipping corner identifiers are not 2. According to Condition (2.2.4.2), they terminate transmitting; and duplicate messages between 00*, 01* and 03* can be avoided. The corner levels of 011, 033 are 2, according to Condition (2.2.3), they terminate transmitting and avoid duplicate messages sent to 1** and 3**. Since corner level of node 000 is 3, according to Condition (2.2.1), it terminates transmitting. Similarly, nodes 122 and 322 broadcast the message to all other nodes of 1** and 3**, respectively.

For readability of this paper, we have described general broadcasting algorithm stage by stage. However, we emphasize that it can also be executed in an asynchronous environment because it can be correctly implemented in each node as local computation and communication.

Theorem 2. By applying general broadcasting algorithm, starting from an arbitrary node, a message can be transmitted to each node of WK(W, L) exactly once with 2^L-1 time steps.

Proof. We will prove the theorem by induction on L.

Clearly, it is true for L =1.

Hypothesis: Assume that it is true for $L = k$.

Induction Step: Suppose that the source node is $s = s_{k+1}s_k \ldots s_2s_1$. By hypothesis, the message can be

transmitted to each node of $s_{k+1}(*)^k$ exactly once with 2^k-1 time steps. According to Condition (2.2.2), each c-corner of $s_{k+1}(*)^k$ (i.e., $s_{k+1}(c)^k$), where $0 \le c \le W$-1 and $c \ne s_{k+1}$, will transmit the message labelled (k, s_{k+1}) to their inter-cluster neighbors, $c(s_{k+1})^k$, respectively. According to Condition (2.2.4), the message can be broadcast in each $c(*)^k$ WK(W, k) subnetwork such that each node receives the message exactly once like applying the corner broadcasting algorithm. Moreover, according to Condition (2.2.3), as the message reaches the corner nodes of these WK(W, k) subnetworks, if the corner node is incident to another WK(W, k) subnetwork, they terminate transmitting and duplicate messages between these WK(W, k) subnetworks can be avoided. By hypothesis, time required for broadcasting in these WK(W, k) subnetworks is also 2^k-1 steps. Thus, total time steps for this broadcasting is $2(2^k\text{-}1)+1 = 2^{k+1}\text{-}1$. This extends the induction and completes the proof. Q. E. D.

4 Conclusions

In this paper, the author presents a novel broadcasting algorithm for the complete WK-Recursive networks. It gains many advantages as follows. This algorithm can guarantee that each node receives the message exactly once within 2^L-1 time steps, which is the diameter of WK(W, L). It is very simple and easy to be implemented. In fact, it requires only extra one integer included in each message and constant time to decide the neighbors to broadcast in an asynchronous environment.

Acknowledgement

This work was supported by the National Science Council of the Republic of China under the contract number: NSC95-2221-E-142-003.

References

[1] G. H. Chen and D. R. Duh (1994), Topological properties, communication, and computation on WK-Recursive networks, *Networks*, 24 303-317.

[2] G. H. Chen, S. C. Hwang, H. L. Huang, M. Y. Su and D. R. Duh (2001), A general broadcasting scheme for recursive networks with complete connection, *Parallel Computing*, 27(9) 1273-1278.

[3] R. Fernandes (1992), Recursive interconnection networks for multicomputer networks, *Proceed. Int. Conf. Parallel Process.*, 1 76-79.

[4] R. *Fernandes* and A. Kanevsky (1993), Substructure allocation in recursive interconnection networks, *Proceed. Int. Conf. Parallel* Process., 1 319-322.

[5] R. *Fernandes* and A. Kanevsky, Hierarchical WK-Recursive topologies for multicomputer systems (1993), *Proceed. Int. Conf. Parallel Process.*, 1 315-318.

[6] S. L. *Johnsson* and C. T. Ho (1989), Optimum broadcasting and personalized communication in hypercubes, *IEEE Transaction on Computers*, 38(9) 1249-1268.

[7] A. I. *Mahdaly*, H. T. Mouftah and N. N. Hanna (1990), Topological properties of WK-Recursive networks, *Proceed. Second* IEEE *Workshop on Future Trends of Distributed Computing Systems*, 374-380.

[8] Y. Saad and M. H. Schultz (1989), Data communication in hypercubes, *Journal of Parallel and Distributed Computing*, 6 115-135.

[9] M. Y. *Su*, G. H. Chen and D. R. Duh (1997), A shortest-path routing algorithm for incomplete WK-Recursive networks, *IEEE* Transactions *on Parallel and Distributed Systems*, 8(4) 367-379.

[10] G. D. Vecchia and C. Sanges (1987), Recursively scalable network for message passing architecture, *Proceed. Int. Conf. Parallel Processing and Applications*, 1 33-40.

[11] G. D. Vecchia and C. Sanges (1988), A recursively scalable network VLSI implementation, *Future Generat. Comput. Syst.* 4(3) 235-243.

[12] L. Verdoscia and R. Vaccaro (1999), An adaptive routing algorithm for WK-Recursive topologies, *Computing*, 63(2) 171-184.

[13] F. Wu and C. C. Hsu (2002), A generalized processor allocation scheme for recursively decomposable interconnection networks, IEICE *Transaction on Information and Systems*, E85D(4) 694-713.

[14] J. Wu and E. B. Fernandez (1993), Fault-tolerant distributed broadcast algorithm for cube-connected-cycles, *Computer Systems Science and Engineering*, 4 224-233.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 800 staff, has 600 researchers, about 250 of whom are postgraduates, nearly 400 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of S**love**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jođef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of Economic Relations and Development, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit an email with the manuscript to one of the editors from the Editorial Board or to the Managing Editor. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica LaTeX format and figures in .eps format. Style and examples of papers can be obtained from http://www.informatica.si. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

QUESTIONNAIRE

☐ Send Informatica free of charge

☐ Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1111 Ljubljana, Slovenia.

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than ten years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: .....

Title and Profession (optional): .....

.....

Home Address and Telephone (optional): .....

.....

Office Address and Telephone (optional): .....

.....

E-mail Address (optional): .....

Signature and Date: .....

Informatica WWW:

http://www.informatica.si/

Referees:

Witold Abramowicz, David Abramson, Adel Adi, Kenneth Aizawa, Suad Alagić, Mohamad Alam, Dia Ali, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Iván Araujo, Vladimir Bajič, Michel Barbeau, Grzegorz Bartoszewicz, Catriel Beeri, Daniel Beech, Fevzi Belli, Simon Beloglavec, Sondes Bennasri, Francesco Bergadano, Istvan Berkeley, Azer Bestavros, Andraž Bežek, Balaji Bharadwaj, Ralph Bisland, Jacek Blazewicz, Laszlo Boeszoermenyi, Damjan Bojadžijev, Jeff Bone, Ivan Bratko, Pavel Brazdil, Bostjan Brumen, Jerzy Brzezinski, Marian Bubak, Davide Bugali, Troy Bull, Sabin Corneliu Buraga, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Rajkumar Bvyya, Giacomo Cabri, Netiva Caftori, Particia Carando, Robert Cattral, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, Mel Ó Cinnéide, David Cliff, Maria Cobb, Jean-Pierre Corriveau, Travis Craig, Noel Craske, Matthew Crocker, Tadeusz Czachorski, Milan Češka, Honghua Dai, Bart de Decker, Deborah Dent, Andrej Dobnikar, Sait Dogru, Peter Dolog, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Marjan Družovec, Jozo Dujmović, Pavol Ďuriš, Amnon Eden, Johann Eder, Hesham El-Rewini, Darrell Ferguson, Warren Fergusson, David Flater, Pierre Flener, Wojciech Fliegner, Vladimir A. Fomichov, Terrence Forgarty, Hans Fraaije, Stan Franklin, Violetta Galant, Hugo de Garis, Eugeniusz Gatnar, Grant Gayed, James Geller, Michael Georgiopolus, Michael Gertz, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Jozsef Gyorkos, Marten Haglind, Abdelwahab Hamou-Lhadj, Inman Harvey, Jaak Henno, Marjan Hericko, Henry Hexmoor, Elke Hochmueller, Jack Hodges, John-Paul Hosom, Doug Howe, Rod Howell, Tomáš Hruška, Don Huch, Simone Fischer-Huebner, Zbigniew Huzar, Alexey Ippa, Hannu Jaakkola, Sushil Jajodia, Ryszard Jakubowski, Piotr Jedrzejowicz, A. Milton Jenkins, Eric Johnson, Polina Jordanova, Djani Juričič, Marko Juvancic, Sabhash Kak, Li-Shan Kang, Ivan Kapustøk, Orlando Karam, Roland Kaschek, Jacek Kierzenka, Jan Kniat, Stavros Kokkotos, Fabio Kon, Kevin Korb, Gilad Koren, Andrej Krajnc, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Sofiane Labidi, Les Labuschagne, Ivan Lah, Phil Laplante, Bud Lawson, Herbert Leitold, Ulrike Leopold-Wildburger, Timothy C. Lethbridge, Joseph Y-T. Leung, Barry Levine, Xuefeng Li, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Vincenzo Loia, Matija Lokar, Jason Lowder, Kim Teng Lua, Ann Macintosh, Bernardo Magnini, Andrzej Małachowski, Peter Marcer, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Daniel Memmi, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Armin R. Mikler, Gautam Mitra, Roland Mittermeir, Madhav Moganti, Reinhard Moller, Tadeusz Morzy, Daniel Mossé, John Mueller, Jari Multisilta, Hari Narayanan, Jerzy Nawrocki, Rance Necaise, Elzbieta Niedzielska, Marian Niedq'zwiedziński, Jaroslav Nieplocha, Oscar Nierstrasz, Roumen Nikolov, Mark Nissen, Jerzy Nogieć, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Daniel Olejar, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczyslaw Owoc, Tadeusz Pankowski, Jens Penberg, William C. Perkins, Warren Persons, Mitja Peruš, Fred Petry, Stephen Pike, Niki Pissinou, Aleksander Pivk, Ullin Place, Peter Planinšec, Gabika Polčicová, Gustav Pomberger, James Pomykalski, Tomas E. Potok, Dimithu Prasanna, Gary Preckshot, Dejan Rakovič, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Vojislav D. Radonjic, Luc de Raedt, Ewaryst Rafajlowicz, Sita Ramakrishnan, Kai Rannenberg, Wolf Rauch, Peter Rechenberg, Felix Redmill, James Edward Ries, David Robertson, Marko Robnik, Colette Rolland, Wilhelm Rossak, Ingrid Russel, A.S.M. Sajeev, Kimmo Salmenjoki, Pierangela Samarati, Bo Sanden, P. G. Sarang, Vivek Sarin, Iztok Savnik, Ichiro Satoh, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Dennis Sewer, Zhongzhi Shi, Mária Smolárová, Carine Souveyet, William Spears, Hartmut Stadtler, Stanislaw Stanek, Olivero Stock, Janusz Stokłosa, Przemysław Stpiczyński, Andrej Stritar, Maciej Stroinski, Leon Strous, Ron Sun, Tomasz Szmuc, Zdzislaw Szyjewski, Jure Šilc, Metod Škarja, Jiří Šlechta, Chew Lim Tan, Zahir Tari, Jurij Tasič, Gheorge Tecuci, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Drago Torkar, Vladimir Tosic, Wieslaw Traczyk, Denis Trček, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Romana Vajde Horvat, Elisabeth Valentine, Kanonkluk Vanapipat, Alexander P. Vazhenin, Jan Verschuren, Zygmunt Vetulani, Olivier de Vel, Didier Vojtisek, Valentino Vranić, Jozef Vyskoc, Eugene Wallingford, Matthew Warren, John Weckert, Michael Weiss, Tatjana Welzer, Lee White, Gerhard Widmer, Stefan Wrobel, Stanislaw Wrycza, Tatyana Yakhno, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Ales Zivkovic, Zonling Zhou, Robert Zorc, Anton P. Železnikar

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.
The subscription rate for 2007 (Volume 31) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printing: Dikplast Kregar Ivan s.p., Kotna ulica 5, 3000 Celje.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Igor Grabec)
ACM Slovenia (Dunja Mladenič)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, ACM Digital Library, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Cybernetica Newsletter, DBLP Computer Science Bibliography, Engineering Index, INSPEC, Linguistics and Language Behaviour Abstracts, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik

The issuing of the Informatica journal is financially supported by the Ministry of Higher Education, Science and Technology, Trg OF 13, 1000 Ljubljana, Slovenia.

Volume 31 Number 1 March 2007 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **31** (2007) Number 1, pp. 1–137