Volume 24 Number 1 March 2000 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

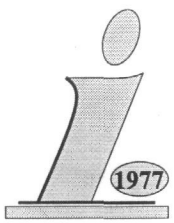
Special Issue:

Database, Web and Cooperative Systems

Guest Editors:

Y. Zhang, V.A. Fomichov, A.P. Železnikar

Informatica 24 (2000) Number 1, pp. 1–146



The Slovene Society Informatika, Ljubljana, Slovenia

Informatica

An International Journal of Computing and Informatics

Archive of abstracts may be accessed at USA: http://, Europe: http://ai.ijs.si/informatica, Asia: http://www.comp.nus.edu.sg/ liuh/Informatica/index.html.

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 2000 (Volume 24) is
– DEM 100 (US$ 70) for institutions,
– DEM 50 (US$ 34) for individuals, and
– DEM 20 (US$ 14) for students
plus the mail charge DEM 10 (US$ 7).
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

LaTeX Tech. Support: Borut Žnidar, Kranj, Slovenia.
Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.
Printed by Biro M, d.o.o., Žibertova 1, 1000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Jožef Stefan Institute: Tel (+386) 61 1773 900, Fax (+386) 61 219 385, or send checks or VISA card number or use the bank account number 900–27620–5159/4 Nova Ljubljanska Banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

Informatica is published 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 (Janez Peklenik)

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, Engineering Index, INSPEC, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik, Linguistics and Language Behaviour Abstracts, Cybernetica Newsletter

The issuing of the Informatica journal is financially supported by the Ministry for Science and Technology, Slovenska 50, 1000 Ljubljana, Slovenia.

Post tax payed at post 1102 Ljubljana. Slovenia taxe Percue.

Introduction: Special Issue on Database, Web and Cooperative Systems

Special Issue Editors:
Yanchun Zhang
Department of Mathematics and Computing
University of Southern Queensland
Toowoomba, Q4350, Australia
yan@usq.edu.au
http://www.sci.usq.edu.au/staff/yan

Vladimir A. Fomichov
Faculty of Applied Mathematics
Moscow State Institute of Electronics and Mathematics
109028 Moscow, Russia
and Department of Information Technologies
K. E. Tsiolkovsky Russian State Technological University -"MATI"
vladfom@yahoo.com
http://www.geocities.com/CapeCanaveral/Hall/3648

Anton P. Železnikar
Volariceva 8, Ljubljana, Slovenia, Europe
s51em@lea.hamradio.si

Recent advances in the Internet and the World Wide Web have made access to various databases and data resources much easier. A result of this is that there is now more research interest in how to efficiently support cooperative work over the Web. The purpose of the 1st International Symposium on Database, Web and Cooperative Systems (DWACOS'99), August 3-4, Baden-baden, Germany, was to provide a forum for exchanging research results on database systems, web data management, and cooperative systems. Due to DWACOS'99's great success, a new conference series for the 21st century, International Conferences on Web Information Systems Engineering (WISE), has been developed, and its first conference WISE'2000 will be held in Hong Kong in June 19-20, 2000 (http://www.cs.cityu.edu.hk/ wise2000).

This special issue contains 10 high quality papers which are published in the proceeding based on the referees' recommendations. They are extensively revised papers that appeared in DWACOS'99 proceedings. The papers have been selected in two steps. Firstly, 25 papers were chosen from 57 submissions to the DWACOS'99 Symposium. Secondly, 10 papers were selected for including their extended versions in this Special Issue.

The 10 papers are divided into 5 categories: the first 2 papers address Web-based database support for distance education and for cooperative work. Papers 3-5 cover distributed and cooperative database processing as well as parallel processing in object-oriented databases. Papers 6 -7 discuss query formulation and information retrieval. Papers 8-9 address electronic commerce and security issues, and the last paper discusses the design of a digital library.

Kambayashi et al presented a Distance Education System which uses action view and action history view. When combining video with action history view, it can realise high level functions for retrieving arbitrary portion of video data. Yun presented an effective architecture for Web-based teamwork automation support and its corresponding mechanisms for visual process modelling for teamwork managers and process enactment for team members in an asynchronous and synchronous manner. The prototype has been implemented in Java with an object-oriented and/or relational database as data repository.

Akiyama et al proposed a dynamic database migration approach, which detects the information access skew and relocates the database over the Internet based on the skew information. The effectiveness has been shown by means of the comparison with the conventional fixed database method. Liu et al developed a cooperative database system called Ozgateway for integrating heterogeneous existing information systems into an interoperable environment. It has provided a gateway for legacy information systems migration. Sasaki et al proposed a performance improvement on Thakore's algorithm for parallel processing in object-oriented databases. They modified the algorithm by introducing the speculative execution technique and adopting dynamic task scheduling in assignment between a class and a processing node. The evaluation has been conducted to show the effectiveness.

Fomichov introduced a new theoretical framework, the Universal Resources and Agents Framework (URAF), for electronic commerce and for developing a more semantically-structured Web. It is based on his original approach to formalizing conceptual structures of complicated natural language texts. Chang et al have developed a new

watermarking method, the threshold watermarking method (TW) for copyright protection of two color bitmaps. TW extracts characteristic values from the bit map first and then generates some information to protect the copyright of this bitmap. The robustness of this method has been shown through experimental work.

Oussalah and Seriai developed a framework for facilitating query formulation in a multi-user environment. A model is developed to store the query formulation skills, and these skills are made available to help users to formulate new queries. The users of a given business can cooperate transparently through their queries to design databases specific to their business. Kobayashi et al address the similarity issues in intelligent information retrieval systems. They proposed a new measure of word similarity based on the normalised information content of concepts in semantic networks. The experimental work shows that their new measure can achieve higher word similarity than other existing ones.

Baptista and Kemp address the design of a spatiotemporal digital library. They discussed the main requirements and issues involved in spatiotemporal digital libraries and proposed a hierarchical metamodel based on four layers of abstraction.

Finally we would like to express our gratitude to the authors and referees for their valuable time spent in preparing and reviewing the papers, and to the Contacting Editor, Prof Matjaz Gams, for fruitful cooperation.

Extensive Interaction Support in Distance Education Systems Utilizing Action History Views

Yahiko Kambayashi, Akihiro Hatanaka, Akira Okada, Madoka Yuriyama
Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto-shi, Kyoto-fu, Japan
Phone: +81 75 753 5375, Fax: +81 75 753 4970
E-mail: {yahiko, hatanaka, aokada, madoka}@isse.kuis.kyoto-u.ac.jp

Keywords: User Cooperation, CSCW, database view, video database, distance education

Edited by: Yanchun Zhang, Vladimir Fomichov, Anton P. Železnikar
Received: August 10,1999 **Revised:** November 12,1999 **Accepted:** November 15,1999

Action views and action history views are promising tool for CSCW (Computer Supported Cooperative Work) with security and privacy constraints. If we combine video with action history sequence, high-level functions for retrieving arbitrary portion of video data can be realized. How to use such functions for distance education is discussed in this paper. Four major topics are (1) recording functions for lecture, (2) retrieval functions for arbitrary portions of lecture video, (3) on-line quiz functions, and (4) discussion support for students.

1 Introduction

Due to the recent progress of data compression technology and rapid reduction of storage cost, it is rather easy to store various kinds of data in low cost. Until recently, data in the real world is selected before storage and stored in specific form due to the storage limitation. In near future it is expected to store everything first and selection is made during usage of data. The method gives great flexibility for data usage, but how to find required portions will become serious problems. We have been developing methods to store video data with various kinds of auxiliary data synchronously, so that required portions can be retrieved easily [6]. For sequence of operations performed by a user, we have defined Action View and Action History View, which will define a proper subset of operations in the operation sequence [7]. A user can observe only necessary operations together with corresponding part of video (if required), by such kind of views.

The functions are especially useful for users working cooperatively yet in different time. Usually systems for CSCW (Computer-Supported Cooperative Work) follow the WYSIWIS (What You See Is What I See) principle; the users share the identical display contents [2]. For many applications such a work mode is undesirable, especially for security/privacy reason. Depending on a user, the contents s/he can see may be different. Private email cannot be read by others. Thus it is necessary to restrict data to be displayed. Action view is defined for such a purpose. If action sequence is recorded, we can use the sequence later. Action history view can extract proper sequences from such a sequence.

In this paper we will discuss how these mechanisms are used for distance education systems [5]. We have developed prototypes for reuse of hypermedia usage, and lecture. Extended usage will be discussed in this paper.

2 Action Views and Action History Views

In this section, basic concepts on action views and action history views are summarized.

2.1 Action Views

For cooperation among distributed users, usually WYSIWIS principle is assumed and awareness tools are used. Under the WYSIWIS principle, all the users share the same contents, and using awareness tools, they can know other status information each other.

However a user sometimes wants to hide own information from others for privacy/security reasons, so in such case, WYSIWIS principle is not adequate. Action Views are extended awareness under non-WYSIWIS environments. If user A works for two projects α and β, and user B works only for project α, then the display contents and operations of user A which is related to project β should be hidden form user B. Note that action view is defined for not only data objects but also methods applied to them.

For displaying objects there are the following facilities.

- **Selection of objects to be displayed**
  Users can make objects hidden from others, because of security/privacy.
- **Partial information about objects to be displayed**
  Users can modify information about objects and show them to other users. A manager does not need to see

the all the details of work of people working for him, but he requires to know such work in abstract way.

For methods to displaying objects, there are the following facilities.

- **Hiding methods to hidden objects**
  If partial information of an object is displayed, abstract motion of method application can be shown.
- **Restrictions on applying methods to displayed objects**
  For example, some users can change the contents, while other users can only read the contents.

Finally, methods to display objects can be changed for preference of users. Shape of icons, color of windows, character size can be changed as far as some constraints are maintained.

Moreover, methods to display objects will be determined by characteristics of display monitors such as resolution and the number of colors. Low resolution will require overlapping of objects.

Note that even if we use only different display methods, it will violate the WYSIWIS principle.

2.2 Action History Views

If operation sequence of one user is recorded, thus that record are called Action History and the function to see a part of such sequence is called Action History Views.

Action History consists of a sequence of events related to particular objects (users). Moreover, those events consist of messages sent to objects and methods triggered by messages. Thus Action History has the following components.

Id an object identifier of an object, a receiver of a message;

M the message having the object identifier as the parameter;

A_{old} it represents attribute values of objects before the method determined by the message is executed;

A_{new} it represents attribute values of objects after the method is executed;

I user's input which triggered the operation;

T user's time when the execution of the method is finished;

S a name of a user who triggered the operation;

Action History Views are introduced to search and access parts of Action Histories, and the result of such operations consists of

- Action Views for each snapshot of Action History
- Snapshot sequence as parts of Action History

Let a history $h =< h_1, h_2, ..., h_k >$ be a sequence of snapshots h_i. The user can replay the whole history, or only such its snapshots that satisfy a given condition; in such cases the order of displayed snapshots is preserved. The user can replay from arbitrary point, example $h =< h_2, h_2, ..., h_k >$, and also can replay skipping some point, example $< h_3, h_5, h_9 >$. Moreover, the user can select one point snapshot and compare other point snapshot.

Besides such selection functions, another important function of Action History View is change of replay time based on priority. Some important sequences can be played in short time.

3 Distance Education System -VIEW Classroom-

In this section, we explain "VIEW Classroom" as one of the example applications using Action Views and Action History Views. VIEW Classroom is a distance education system that we have been developing.

3.1 Outline of VIEW Classroom

Figure 1 illustrates the conceptual image of VIEW Classroom. In VIEW Classroom teachers and students are suppose to be in distributed location and individually have personal computers connected via Internet. Using computers, they attend a virtual classroom through network. By CCD cameras and microphones, teachers and students communicate each other in VIEW Classroom. Besides online participation of the class, offline (after-class) participation is realized by Action History Views.

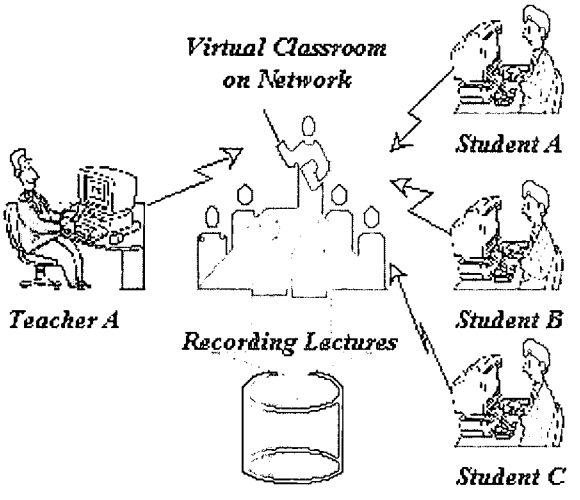


Figure 1: Concept of VIEW Classroom

Basically, teachers conduct classes the following using-facilities.

- **Hyper Media-based Texts and Hyper Media Browsers**

In VIEW Classroom, texts are Hyper Media-based. Teachers and students see the texts using hyper media browsers.

- **Notes**
  Teachers prepare notes before lectures and students can make notes of lectures. They select which others can see their notes or not.

- **Pointers and Markers**
  Because of attracting attentions, teachers use pointers and markers on hyper media browsers. Movements of teachers' pointers and markers are broadcasted in realtime. Students can use, too, however movements of their pointers and markers are broadcasted only when teachers' allowance are given.

3.2 Recording Functions of VIEW Classroom

In VIEW Classroom, class activities are recorded in database as Action History. One of the serious problems for using multimedia data is how to find a portion corresponding to a user's request. Although automatic scene analysis and speech recognition are very popular among researchers, it is still not easy to form reliable index in reasonable amount of time. So we approach the way cooperative works are recorded not only by videos but also by action histories, and realize flexible search functions using action histories. First we describe recording functions of VIEW Classroom.

Examples of recorded activities are

- Teachers' video and audio
- Movements of teachers' pointer and marker
- Transition of Text Pages

These activities are recorded using time-stamps because of synchronization.

Table 1 shows example of Action History in VIEW Classroom. Video and Audio are recorded with timestamps automatically, so other actions are recorded as Action History. Teacher Taro explains the slide whose identifier S1 using marker. Marker is activated by mouse_button1_down at 09:10:11, Taro writes on the slide S1 by dragging the mouse to the point x: 200 y: 456 at 09:10:12, x: 254 y: 456 at 09:10:13. Marker is deactivated by mouse_button1_release at 09:10:14.

3.3 Flexible Search Functions of VIEW Classroom

In this section, we will discuss flexible search functions for recorded action histories and videos. Figure 2 shows search functions user interface of VIEW Classroom. In VIEW Classroom users can search by following information.

- **Time**
  Users can replay videos from arbitrary point by using Fast Forward button and Rewind button of the text browser.

- **Slide Texts**
  Users can search slides by searching texts in slides. Slides that contain search keywords are replayed in order of slide id.

- **Notes Texts**
  Teachers sometimes edit on notes and open such notes to students. Texts in such notes can be object to search. Users can search not only public notes but also private notes of one.

- **Movement of Marker**
  Teachers and students can use markers and write on slides directly. Users can find slides that are much written on by searching action histories of markers.

- **Structures of Slides**
  Structures of slides are search object, too. Users can search slides that contain pictures, movies, links or items.

- **Slides List and Timetable**
  VIEW Classroom has slides list that shows slides used in classes, and timetable that shows when and what operations are happen. Using slides list, users can replay arbitrary slide. Using timetable, users can know how many time each slides are used for explaining.

- **Use of Facilities**
  Not only teachers but also students use several functions. For example, users can specify the video location when students are talking.

3.4 Example Usage of Search Functions

In this section, we describe example usage of search functions. By thinking several scenarios, users can realize advanced searches.

(1) Time required for explaining each slide
Time used for explaining each slide reflects that

- The slide is important.
- The slide is difficult to understand.

In the latter case, either the explanation of the teacher is long or there are a lot of questions from the students. In both cases, the slide seems to be important if the time for explanation is long. Users can find such slides by using timetable. Timetable shows when a class is started and finished, and each of slides are changed. One possible exception is that the teacher talks about something else not related to the topic of the slide. To find such cases, we have coffee icon and whenever the teacher wants talk on something else s/he has to select the icon. The icon will be put

Table 1: Example of Action History

L_{id}	S_{id}	Act_{name}	Act_{time}	Act_{type}	x	y	$slide\_id$
$L1$	$S1$	$Taro$	09 : 10 : 11	$Mouse\_button1\_down$	123	456	
$L1$	$S1$	$Taro$	09 : 10 : 12	$Mouse\_move$	200	456	
$L1$	$S1$	$Taro$	09 : 10 : 13	$Mouse\_move$	254	456	
$L1$	$S1$	$Taro$	09 : 10 : 14	$Mouse\_button1\_release$	254	456	
$L1$	$S1$	$Taro$	09 : 15 : 20	$Next\_slide$			$S2$
$L1$	$S2$	$Taro$	09 : 22 : 30	$Mouse\_button2\_down$	456	100	

on the slide so that a student can retrieve slides with coffee icon.

(2) Movement of marker and editing shared notes

If there is a lot of writing on a slide and much editing public notes, we can assume that the slide is important. For teachers, the purpose of using marker and editing public notes is

- To explain in detail
- To discuss on the topic not in the slide
- To correct mistakes

As important concepts will be shown by underlines, a slide with many underlines may be also important. Such extracted words can be used to identify the characteristics of the slide. Such slides can be found by searching movement of marker or public notes.

While for students, the purpose of using marker and editing private notes is

- To show the places where s/he thinks important
- To add explanation since s/he does not know the details

Basically students' actions are private and cannot be searched. The location of one student's marking and the contents of private note should not be known and cannot be searched by teachers and other students. However they can search and know abstract information of such activities. Slide id of one student's marking and the amount of editing private notes can be searched and known by teachers and other students. By searching such abstract students information, teachers can find

- Which slides is lack of explanation.
- What topics students have interests in.

While, by searching abstract other students information, a student can find

1. Which there is oversight or not.
2. Other students' trend

Searching not only detail information but also abstract information is useful for users, however users must pay attentions to privacy problems.

4 On-Line Quiz Functions And Action History Views

4.1 On-line quiz functions

In education, interactions among the teacher and students are very important. In distance education, however, they are limited because of spatial dispersion among the teacher and students. Interaction support for distance education system is essential. In conventional distance education systems, only communication from the teacher to students is regarded as important. In education, however, communication from the students to teacher or among students is very important, too. In this section, we will discuss the former. In interaction from students to the teacher, most important point for the teacher is to know the level of understanding of the students. The teacher can use this information to improve the contents of the lecture and to improve teaching methods. There are the following methods to find student progress.

(a) Examine the contents of notebooks of students.

(b) Ask questions to (randomly selected) students.

(c) By defining buttons for "easy", "understand", "difficult", "not understandable" and at each point, the teacher asks to select the buttons. The resulting statistics is shown in the display of the teacher.

(d) Give quiz or examination to students at appropriate check points.

Since (a) may violate the privacy of the students and (b) is time-consuming, we have implemented (c) and (d). We will discuss (d), on-line quiz function here. Figure 3 is an example of on-line quiz. Giving on-line quiz in the lecture and seeing its results, the teacher can know students' level of understanding objectively.

4.2 The process of on-line quiz

For the teacher, the process of on-line quiz consists of the following four steps.

(1) Prepare problems before class.
The teacher prepares problems of on-line quiz based on his educational strategy before class.

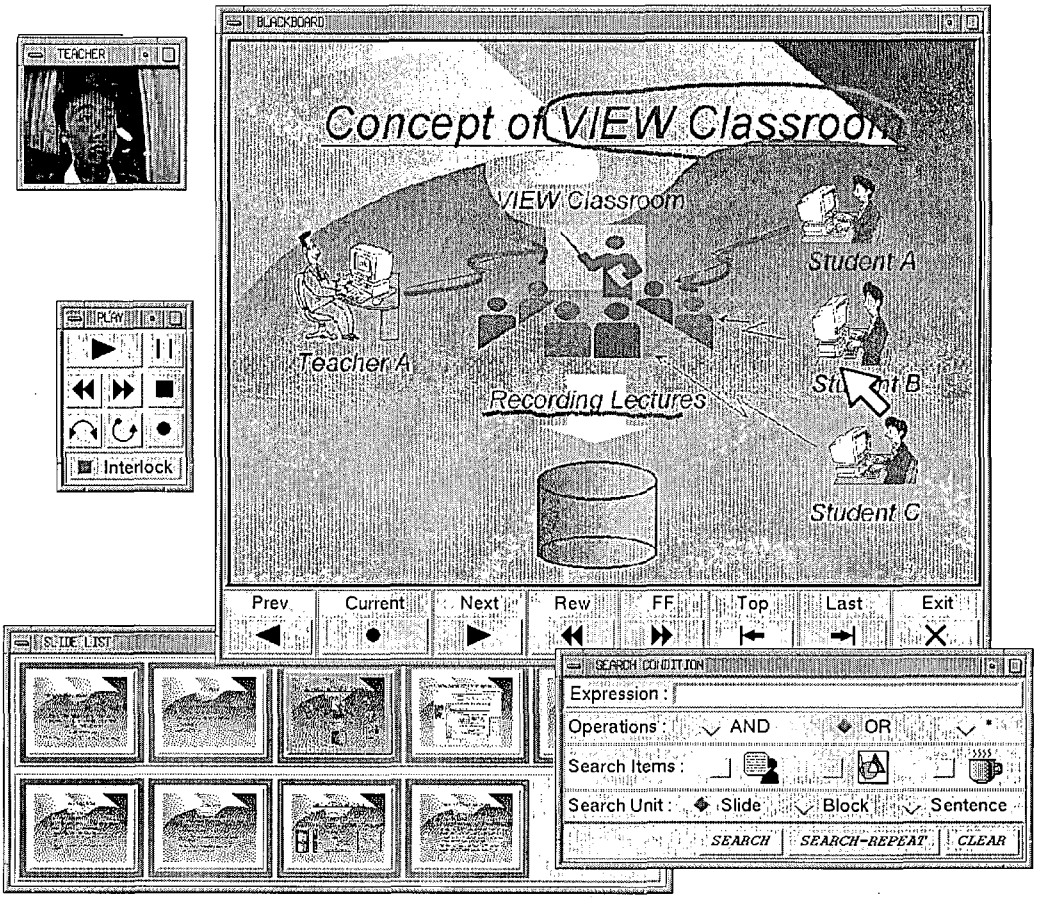


Figure 2: User Interface of VIEW Classroom

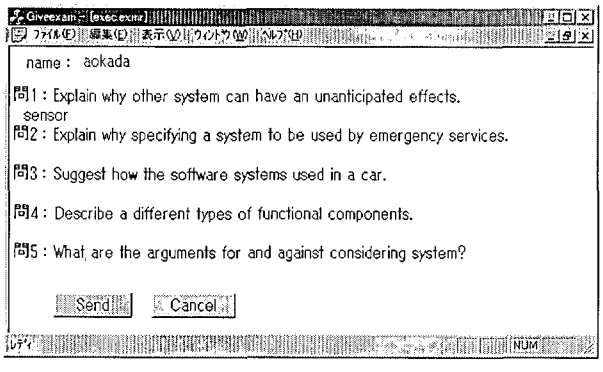


Figure 3: On-line Quiz Function

(2) Give a quiz in the lecture.
The teacher gives quiz prepared before the class to students at appropriate points of the lecture. On-line real time generation of problems is possible as well, by teaching a part of texts.

(3) Analyze the results of on-line quiz.
It is difficult for the teacher to know students' level of understanding by only seeing students' answers roughly. To know students' understanding level more preciously and deeply, the teacher analyzes various data obtained during quiz, not only answers.

(4) Guide students.
The teacher guides students based on the results of analysis. Through this step, students can understand contents of the lecture.

From a viewpoint of interactions among the teacher and students, (3) is very important. The teacher knows students' level of understanding through the step of (3). Therefore we put more emphasis on (3) in our development. We discuss how to utilize Action View and Action History View mechanisms for this function in the followings.

4.3 Actions for analysis of on-line quiz

The on-line quiz system equips the function of analyzing results of on-line quiz to support the teacher. This function uses the mechanisms of Action View and Action History View. One of features of on-line quiz is collection and analyzing of data using the network and database. These data are saved in the database as Actions, and our system can

generate various Views from these data for the teacher. The on-line quiz system collects the following Actions together with time required to perform these actions.

- To solve one problem.
- The order of solving the problems.
- The process of changing answers.

4.4 Action Views for on-line quiz

The Action Views are used as follows.

- **The situation of students in the quiz**
  The teacher can see the situation of students in real-time, such as what problem a student is thinking, time required for a student to answer a problem. For example, the teacher can give a hint to students if they have been trying to solve a problem for a long time.
- **The results of quiz of students in real-time**
  The teacher can see the students' answers and their exam results in real-time. Such information helps the teacher to improve the contents of the lecture dynamically. Although these are various kinds of views, one possible view is to show the dependency structure of the problems. Figure 4 shows an example. The progress of student un-derstanding is shown as a process through the vertices. This View can show the results in various forms. One example of this view is shown in Figure 4. This example presents students progression of understandings.

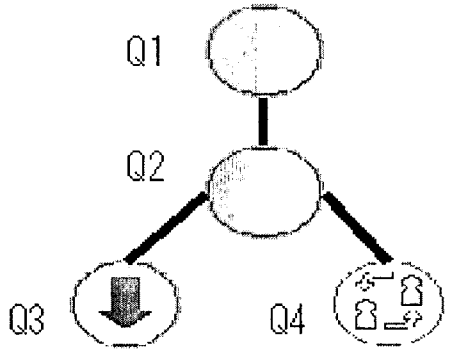


Figure 4: The Dependency Structure among Problems

4.5 Action History Views for on-line quiz

The Action History Views are used as follows.

- **Time series of the student's action**
  Such Views show time series of actions of the student to on-line quiz. The target actions are
  - To give an answer to a problem (and whether it is right)
  - To cancel an answer

  This view shows above actions in the style of two-dimensional graph (Figure 5). The x-axis presents time and the y-axis presents the number of problem. The sequence of a student's action is presented as a line graph. The apices present actions of the student. From this View, the teacher can know the action sequence of the student to on-line quiz with information of time. To analyze this, the teacher can get various knowledge about the student, such as the problems difficult for the student or weak points of the student.

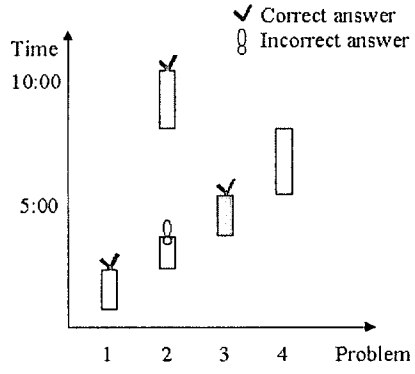


Figure 5: The View of Time Series of the student's action

- **The growth of students' understanding**
  The teacher can know the growth of students' understanding. For example, if quiz, consists of the same or similar problems is given repeatedly, the growth of results shows the growth of students' understanding. In other case, time required to answer becomes to be short, it seems that the level of students' understanding grows. Knowing this, the teacher can review his lecture objectively.

5 Support of Discussions among Students

Although in the real classrooms interaction among students are believed to be important, conventional distance education systems do not support such interactions. In this section we will discuss a system which mainly supports interaction among students. It can be used in the following situations.

1. During the lecture, the students can discuss on the topic being taught.
2. Using the previous lecture history views, students are studying (synchronous self-study).

3. After the class students will study some part of lecture again and they may ask questions to the friends which may cause discussion among students (asynchronous self-study).

There are the following differences between our system and conventional chat systems.

1. The teacher can observe the contents of discussion by the list of words used in the chat.
2. The teacher can control the discussion by interrupting it or by stopping it.
3. The rights given to the teacher can be different from the rights given to the students.
4. To analyze discussions useful functions including action history views are realized.

We have designed a discussion mechanism among students and teachers.

5.1 Interaction among students

The following system supports discussions among the students and teachers. Lecture notes or materials given by teachers are shown in the opinion screen (Figure 6), and students can discuss on the topic shown.

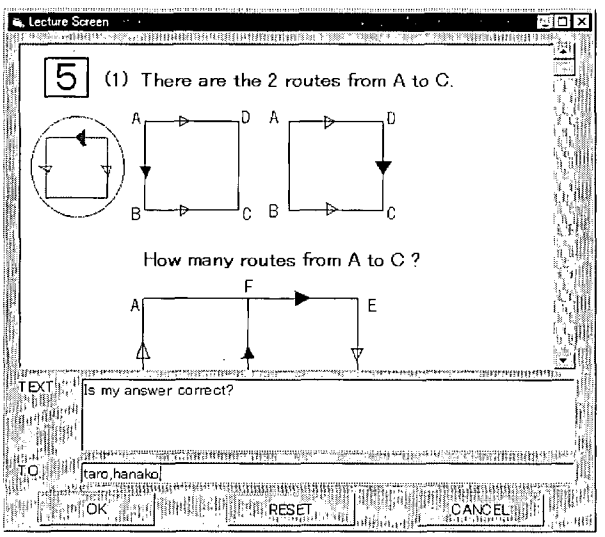


Figure 6: Opinion Screen

New discussion

In order to start a new discussion, the following steps are taken.

1. A student writes an opinion on the opinion screen (Figure 6) directly and sends an opinion to selected people. The opinion consists of text and writing data on the screen.
2. The receivers know the arrival of an opinion by the topic screen (Figure 7).
3. The receivers select the discussion which they want to join, write opinion on the opinion-send screen directly and send opinions (text and writing data on the screen).

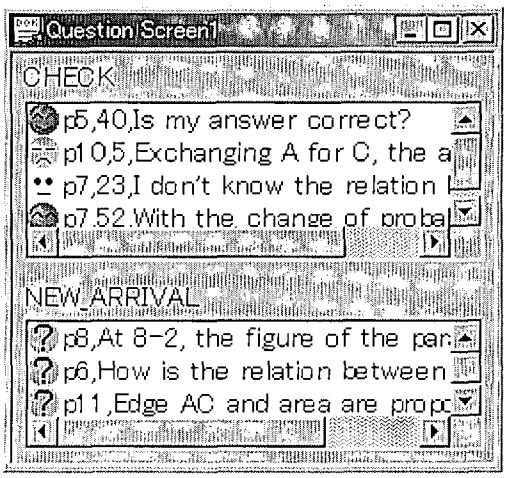


Figure 7: Topic Screen

Discussion

In order to join existing discussion the following steps are performed.

1. By the double click of the icon of the topic screen, the discussion tree is displayed (Figure 8.
2. They select the discussion which they want to join, write opinions on the opinion screen directly and send opinions (text and writing data on the screen).

There are many examples of known methods using text-base communication mechanisms such as chat. They are not sufficient means for communication because text-base discussion is limited in its expressive power. Our system is different in the following points.

1. Teachers can limit discussions of stu-dents.
2. There is a function to search discussion history from a database.

5.2 Use of Action Views

In our system action views are utilized as follows.

- **Situation of discussions and students who are in discussion**
  Teachers can know who are in discussion. Students who frequently participate in discussions may make idle talks or actively tackle discussions. On the other

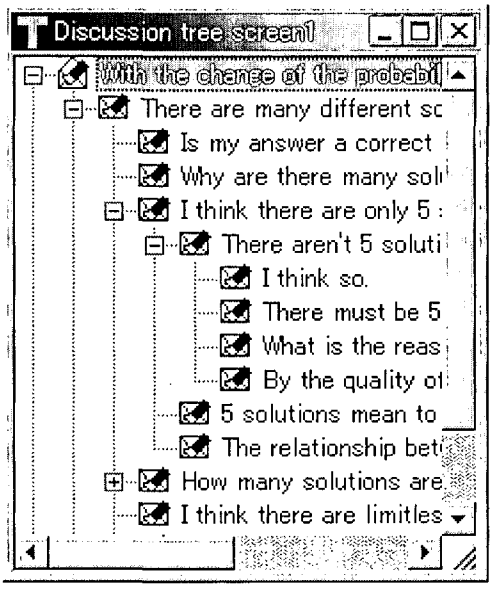


Figure 8: Discussion Tree Screen

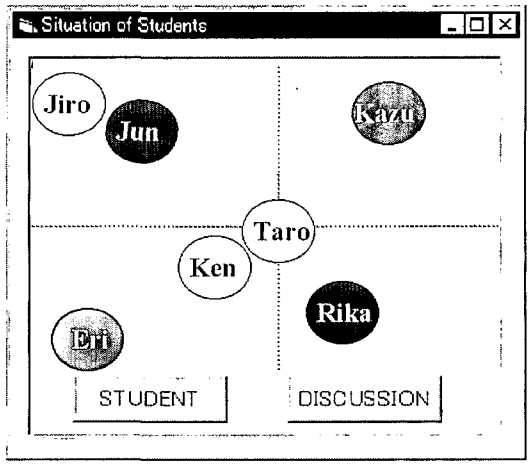


Figure 9: Situation of Students

hand students who don't participate in discussions may not be interested in the topic. If a lot of students are participating in the discussion, the teacher should participate in it. The system show not only activities of students but also relations among students. Students who participate in the same discussions or similar discussions are located closely (Figure 9). The distance between student A and student B is determined by the number of the same discussions or similar discussions which A and B join together. The similarity between discussion A and discussion B is defined by the keywords used in the discussion. The system shows teachers and students situation of discussions and relations among discussions, too. A discussion is defined as a circle. If more students participate in discussion, the circle of the discussion becomes bigger. Circles of which discussions are similar each other are closely placed possibly with overlap.

- **Topics of discussions**
  It is required to know what kinds of topics are discussed. It can be realized by taking keywords from sentences used in the discussion. If the topic is not related to the contents of the lecture note or the materials given by teachers, the teacher can order these students to stop the discussion.

- **Identification of positions where discussion is related**
  The position of the teaching material where the discussion is related may show the topics that students cannot understand or they are interested in. The system shows teachers the position in the lecture note or materials given by teachers where discussions are related. The positions are marked. If there are a lot of marks, the place should be examined.

- **Status of students**
  To observe status of students in abstract way, we can use an imaginary classroom. The seat of each student is fixed and the seat shows the status of the student. For example, a student is writing something on his/her notebook, the seat is shown by red. The seats of the students in the same discussion group are colored by the identical color.

5.3 Use of Action History Views

Action history views are used as follow.

- **Students' participation**
  It is required to know who frequently participate in discussions. Such information may be used for evaluation.

- **Important topics**
  Teachers require repeated topics and topics discussed for long time. Such discussion should be analyzed in detail.

- **Parts where a lot of students participate**
  When the lecture is replayed, the part where a lot of students participate in discussion may be important portion. The amount of opinions and keywords change as time goes. This change from the start to the end is shown. (Figure 10) This graph consists of two parameters. One is the number of opinions in all discussions. The other is keywords and the number of opinions in the selected discussion.

6 Concluding Remarks

In this paper, we describe Action Views, Action History Views and their application to distance education system

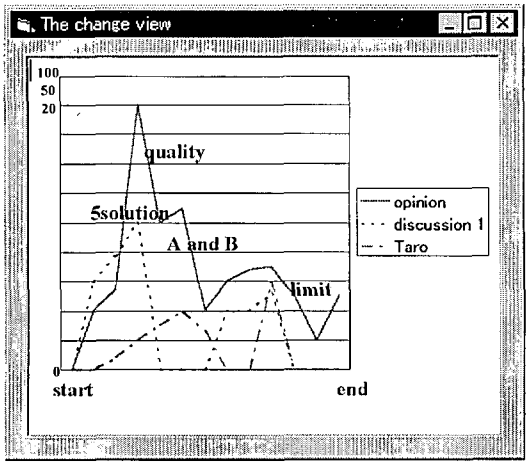


Figure 10: The Change of Discussion

"VIEW Classroom". The prototype was developed using SGI workstations. Currently, we are developing second version of VIEW Classroom using Windows. In the future, there are two directions for VIEW Classroom.

1. First is to support real-time distance lecture on Internet. The on-line quiz function, the discussion function and the other functions using Action View mechanisms are used. Using on-line quiz function, the teacher can guide students in various methods, such as group guidance according to their understanding found by the results of quiz. Using the discussion function, the teacher can change the lecture as the situation of students in discussions.

2. Second direction is to support distance education using digital lecture library. We think that the digital lecture library is one of the digital libraries and museums and keeps lecture records using Action History View mechanisms. Users who want to learn something enter this library, and study by themselves reviewing lecture records, editing those records and communicating each other. Digital lecture library is thought to be one of the most reasonable systems supporting life-long learning.

7 Acknowledgements

The authors would like to thank to Pr. K. Subieta (Polish - Japanese Institute of Information Technologies), Pr. O. Kagawa (Hiroshima Kokusai Gakuin University), Pr. H. Tarumi (Kyoto University) and Mr. K. Katayama (Kyoto University) for their discussion on the topic discussed here.

References

[1] Berlage, T. and Spenke, M. (1992) The GINA Interaction Recorder. *Engineering for Human-Computer Interaction*, p.69-78

[2] Ellis, C.A., Gibbs, S. J. and Rein, G. L. (1991) GROUPWARE: Some Issues and Experiences. *Communications of the ACM*, vol.34, No.1, p.39-58

[3] Greif, I. and Sarin, S. (1987) Data Sharing in Group Work. *ACM Transactions on Office Information Systems*, Vol.5, No.2, p.187-211

[4] Iwamoto, H., Ito, C. and Kambayashi, Y. (1998) Design and Implementation of Action History View Mechanisms for Hypermedia Systems. *Proceeding of Computer Software and Applications Conference (COMPSAC)*, p.412-420

[5] Kagawa, O. and Kambayashi, Y. (1997) Advanced Database Functions for Distance Education System: VIEW Classroom. *Proceedings of the 1997 International Database Engineering and Applications Symposium (IDEAS97)*, p.231-239

[6] Kambayashi, Y., Katayama, K. Kakimoto, T. and Iwamoto, H. (1998) Flexible Search Functions for Multimedia Data with Text Data. *ACM Symposium on Applied Computing*, p.498-504

[7] Kambayashi, Y., Subieta, K. and Fujita, K. (1998) Action View Mechanisms for Cooperative Work Environments. *Proceedings of the 12th International Conference on Information Networking*, p.501-504

[8] Kambayashi, Y. and Peng, Z. (1996) An Object Deputy Model for Realization of Flexible and Powerful Object-bases. *Journal of System Integration, Kluwer Academic Publishers*, vol.6, No.4, p.329-363

[9] Konomi, S., Yokota, Y., Sakata, K. and Kambayashi, Y. (1997) Cooperative View Mechanisms in Distributed Multiuser Hypermedia Environments. *Proceedings of the 2nd IFCIS Conference on Cooperative Information Systems (CoopIS-97)*, p.15-24

[10] Takada, H. and Kambayashi, Y. (1993) An Object-Oriented Office Space Description Model and an Office View Mechanisms for Distributed Office Environment, *Proceedings of 4th International Conference on Foundation of Data Organization and Algorithms*, p.362-377

[11] Peng, Z. and Kambayashi, Y. (1998) Realization of Computer Supported Cooperative Work Environments Using the Object Deputy Model. *Proceedings of the International Database Engineering and Application Symposium'98*, p.276-285

An Architecture and the Related Mechanisms for Web-based Global Cooperative Teamwork Support

Yun Yang
School of Information Technology, Swinburne University of Technology, Hawthorn, Australia 3122
Email: yun@it.swin.edu.au

Keywords: teamwork, CSCW, processes, Java, Web, visualisation, databases

Edited by: Yanchun Zhang, Vladimir A. Fomichov and Anton P. Železnikar
Received: August 15, 1999 **Revised:** November 8, 1999 **Accepted:** November 15, 1999

Given the exposure of the Internet and the Web, there is a significant impact on Web-based cooperative teamwork support which can be beneficial to many teamwork managers and normal team members who may be either computing or non-computing professionals. In this paper, we focus on our research into a more effective architecture for Web-based teamwork automation support and its corresponding innovative mechanisms for various perspectives including visual process modelling for teamwork managers and process enactment for team members in an asynchronous and synchronous manner. Our research prototype is implemented in Java and the data repository used can be an object-oriented or relational database.

1 Introduction

Teamwork is a key feature in any workplace organisation. In this computing era, a process/project is usually carried out by a cooperating team who may be physically dispersed by using various (software) tools. Systems for computer-mediated teamwork, groupware, workflow or CSCW (computer-supported cooperative work) offer various automatic support mechanisms for team cooperation to improve productivity. Generally speaking, a process is normally composed of tasks which are partially ordered (Feiler & Humphrey 1993). How to manage these tasks is the key issue for completion of the entire process. With software support, team members are coordinated by a system which is clearly more effective than being managed manually by a human being. In addition, team members may, for example, reside in Asia-Pacific, Europe and North America. With about 8-hour time differences among locations, a 24 hour a day working mode can be potentially facilitated (Gorton & Motwani 1996).

Nowadays, there is a growing interest in supporting cooperative work over the Internet (or Intranet) and the Web. The emergence and wide-spread adoption of the Web offers a great deal of potential for the development of collaborative technologies as an enabling infrastructure (Oreizy & Kaiser 1997). In addition, the Java programming language, which has the capabilities of delivering applets over the Web as well as the slogan of "write once and run anywhere", i.e. platform independence, has encouraged us to prototype our work in Java and based on the Web environment. In this case, no particular software needs to be installed for team members regarding teamwork coordination since Java applets can be downloaded on the fly and then run directly. Furthermore, using combination of Web/Java seems better than using Web/CGI (common gateway interface) (Evans & Rogers 1997) in terms of performance and control/data granularity. Therefore, we have treated the Web and Java as an excellent, if not ideal, vehicle to prototype our teamwork support mechanisms in a global distributed environment.

In this paper, we start with related work and then we focus on major issues involved in teamwork support. The topics we cover in this paper for supporting Web-based global teamwork include a dedicated architecture, visual teamwork modelling, enactment in an asynchronous fashion, which are enhanced by process evolution and dynamic resource management, as well as synchronous collaboration. Finally, the conclusions are drawn and the future directions are pointed out.

2 Related work

In a Web-based environment, there exist quite a few systems for teamwork support. The prototype developed by Scacchi and Noll (Scacchi & Noll 1997) takes an approach of using HTML forms and associated CGI scripts for process support. The workflow system investigated by Groiss and Eder (Groiss & Eder 1997) is based on using the standard email and HTTP to process information sent as HTML pages. Both of them are coarse-grained compared to the Web/Java approach. BSCW (Bentley et. al. 1995) is a Web-based system to primarily support a shared workplace in an asynchronous manner. The work implemented in Java by Ly (Ly 1997) is mainly for project management. In Serendipity-II, Grundy et. al. use a decentralised architecture for process modelling and enactment (Grundy et. al. 1998). In general, on one hand, most (process-centred) teamwork environments including workflow systems focus

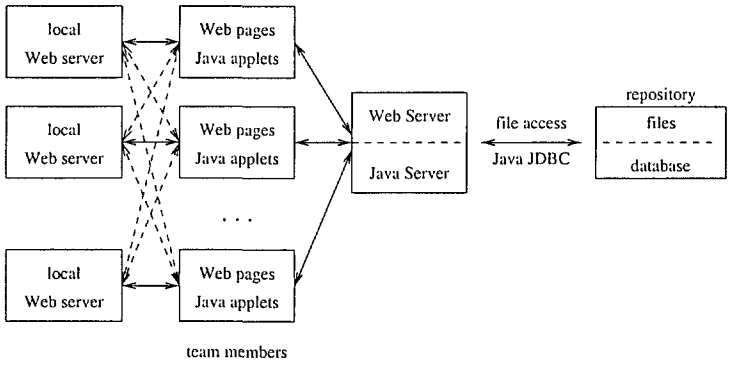


Figure 1: Architecture for supporting teamwork processes

on coordination support among partially ordered individual activities. On the other hand, most CSCW systems focus on collaboration support among the team members for some individual activities in the process.

Research into teamwork process support has been carried out for more than a decade and many fruitful outcomes have been achieved. However, there are still many open issues to be solved in the long run (Ambriola et. al. 1997, Fuggetta & Ghezzi 1994, Sheth 1997). Our work is unique as follows. Firstly, we have proposed an effective semi-centralised multi-tiered client-server architecture for teamwork support as explained in detail in this paper. Secondly, we have covered most inter-related areas to better reflect real world teamwork such as Web-based visualised global teamwork for asynchronous coordination with strong support of process evolution, resource management, and synchronous collaboration. Finally, we have offered many innovative mechanisms for teamwork support as addressed in individual sections in this paper when appropriate.

3 Teamwork support architecture

There exist various system architectures that support teamwork such as centralised and decentralised. We have chosen the semi-centralised multi-tiered client-server architecture for Web-based teamwork process support as depicted in Figure 1 (Yang 1998). This architecture includes (1) clients as front-ends using local Web servers and tools, (2) centralised servers with tools, and (3) supporting tools such as databases and file systems as back-ends. The advantages of this kind of architecture are addressed below.

The centralised server site plays the role of managing the teamwork processes (or projects) based on a process engine, and provides some centralised tools such as synchronous cooperative editors. This kind of centralisation can reduce the teamwork coordination inconsistency in a Web-based environment dramatically. The process information that resulted from modelling is stored in the database repository. Please note that the database repository is a general concept which can include various databases such as relational and object-oriented databases. During process enactment, information such as documents can be stored locally at the client sites or at the server site and accessed by team members based on the Web support which implies that information can be distributed rather than only centralised. This architecture also offers the flexibility of supporting teamwork in an offline (mobile) mode in additions to the normal online mode. Basically, for process coordination, at the client site, only an appropriate Web browser is required and no other software needs to be installed since Web pages and Java applets can be downloaded on-the-fly. Certainly, local tools can still be used for carrying out tasks.

For data repository, we have experimented with two types of databases (Yang et. al. 1999b): the Oracle relational database and the ObjectStore object-oriented database. The experiment results are in favour of deploying an object-oriented database to support process-centred teamwork. With a Java interface, such as that in ObjectStore, we only need to handle objects in Java directly without concern for mapping between the objects in Java and tables in the (Oracle) relational database. In fact, it is more natural to carry out a process in the object-oriented manner which is another important reason why we are in favour of using an object-oriented database as data repository.

4 Supporting teamwork modelling

Over the last decade, process modelling, such as the rule based paradigm, has been investigated intensively which is assessed comprehensively in (Ambriola et. al. 1997). We view the teamwork process very much as a reactive system. Reactive systems are characterised as owing much of their complexity to the intricate nature of reactions to discrete occurrences and the common notion imposed is the reactive behaviour (Harel et. al. 1990). Extending (Harel et. al. 1990, Zhou & Eide 1998), we use the reactive system concepts to model three layers of teamwork process coordination control. At the bottom, the policy layer for making decisions relies on the middle mechanism layer for sensing and actuating application objects, which are at the top application layer. With this paradigm, if a top layer policy is changed, it may have no impact on related middle layer mechanisms and vice versa. We illustrate next that how

the reactive system paradigm can effectively coordinate the process.

As indicated earlier that a process is composed of partially ordered tasks. In the normal sense, the partial ordering implies that a task should and can only start (including bypass etc.) when all its previous tasks (i.e. the AND condition) have been completely finished (i.e. 100% completion rate). However, in reality, it may not be the case. For example, a task may start when one of the two previous tasks is finished (i.e. the OR condition). As another example, a task may also start when the previous tasks reach a certain threshold, say a 80% completion rate. Certainly, there could be other (complex) conditions for invoking the execution of a task. In other words, the coordination should be able to be supported by some fine-grained policies instead of the very coarse-grained ones in most, if not all, existing process modelling paradigms. With our innovative reactive system paradigm, for example, if the decision making condition based on two sensors was "AND" and is now "OR", and even the values for the sensor thresholds are changed, the sensors can still be used without any changes required, i.e. the mechanism layer can remain unchanged. These features offer the flexibility most other paradigms lack. Due to the space limit, readers are referred to (Chen et. al. 1999) for the details of our reactive system paradigm for process coordination.

Modelling of a teamwork process using a computer modelling language is a time consuming and difficult task. Given the exposure of graphical user interfaces oriented environments, it is now expected that modelling support provided should be a visual editing system (Gruhn & Urbainczyk 1998). However, many teamwork support environments do not offer this critical feature. There is no doubt that visual modelling would provide a quicker, easier and less-time consuming process modelling support regardless whether it is used by computing or non-computing professionals.

Figure 2 depicts the main layout of the Java applet for visual process modelling (Yang & Wojcieszak 1999a]. This layout of the interface features grids that form rows and columns. The numbers represented at the bottom of the grids are in fact the hours, days, months or years depending on the "mode" selected. The horizontal scrollbar is used to scroll over the next consecutive rows of hours, days, months or years. The vertical scrollbar allows for parallel tasks within the process. The idea behind the scrollbars is to ensure that process modelling is not limited in any way in order to provide a full view of the process which is not restricted by the time or size constraints.

The teamwork manager can model the process visually by creating new tasks. The oval shaped objects located in between the grids are the process tasks themselves which can be labelled and linked to specify ordering and so forth. The "zoom" button permits the manager to zoom in or out of the interface. This allows for the manager to concentrate on the lower-level details of the process to "divide and conquer". When facilitating the system, for example,

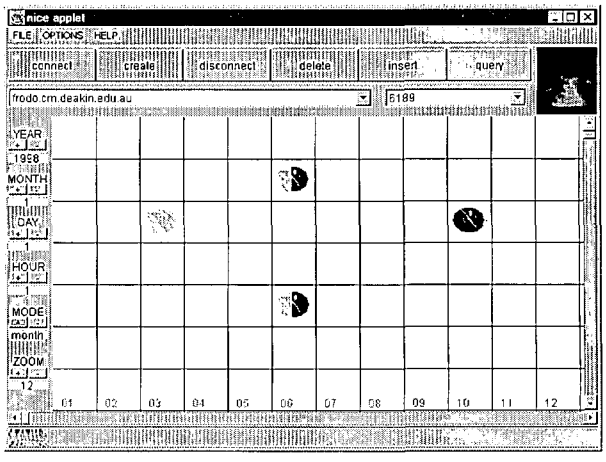


Figure 2: The main layout for teamwork process modelling

if the manager clicks on the button that handles the starting/ending dates and other artefacts for a particular task, information of the selected dates and the other artefacts will be mapped onto a database object and stored within the database. Similarly if the manager decides to link two tasks, the ordering information in the database will be updated accordingly.

5 Supporting teamwork enactment

After a process is modelled by the teamwork manager, it is ready to launch the process for teamwork coordination. For teamwork coordination in our environment, once the process is started, the most essential facility is that each team member is provided with a dynamic up-to-date to-do list. For example, as depicted in Figure 3, the "integration" task is on the to-do list for that particular person. There are practically two basic strategies for the to-do list notification: active and passive, which are all used. The active notification strategy is to send emails via JavaMail to appropriate team members to notify them of the new tasks. The passive way is to get the to-do lists refreshed on demand by team members. We note that the layout of the process in Figure 3 is slightly different to that in Figure 2 because we are testing different visual presentations to see which one is more user friendly. With notification, there could be other information to be passed on such as instructions/messages for the work to be done and sensitive indicators for deadlines. In general, team members normally do not rely much on the centralised server because they mainly work locally on the client side to carry out the tasks assigned. This creates the opportunity for us to explore teamwork support in an offline (mobile) mode, such as working at home, in addition to the normal online mode.

Team members can use local tools, or tools available in the online teamwork environment, to carry out tasks. Sometimes, tools can and need to be specified in the process. For instance, some tasks may involve several team members cooperating at the same time, hence a centralised

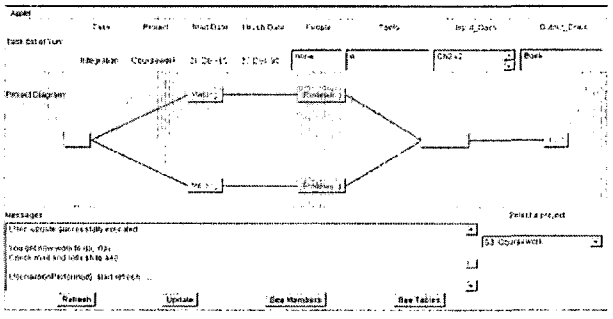


Figure 3: Process coordination user interface

Web-based cooperative editing tool as described in the next section may be better automatically invoked to allow team members to use it as a shared workspace. Even some local tools such as a single-user editor for individual team members can also be indicated to enable automatic tool invocation. From the information/data exchange point of view, data can be stored either locally or at the server side, which can then be easily accessed across the Internet with some simple and extensible standards, such as HTTP, based on the Web support. The richness of data/object types, such as multimedia, can also be achieved. To manage data exchange in a teamwork process, most data types such as documents are specified during the process modelling. In addition, messages from team members during process enactment can be recorded and forwarded to other team members for fine-tuning effective data exchange.

When a certain task is finished or a value of the sensor based on team member's input is changed, a notification is sent to notify the process support environment at the server side. The process engine (a daemon) of the environment will utilise the decision making policy to generate new to-do lists for all affected team members. For a team member working in a team environment, it is very useful to have a global view of the process in a visualised fashion in order to create a better teamwork atmosphere as shown in Figure 3. This is important from the psychological point of view when a person works in a computer-mediated teamwork environment. Different colours are used for the status of each task to indicate whether the task is enacted, enacting or unenacted which are completed, currently ongoing and not yet invoked respectively. The global view of the process is adjusted automatically whenever the status of any task is changed.

6 Supporting synchronous teamwork

Process coordination as described in the preceding section has involved various mechanisms for supporting shared workspaces in an asynchronous manner. It is common that most teamwork activities are undertaken by team members individually. That means that most tasks are carried out asynchronously, but interdependent, i.e. the outcome of a task of one team member is often the input to the tasks of other team members. However, some tasks are shared, i.e. they involve team members working together synchronously to complete the activity such as cooperative editing and brain storming. For example, the integration activity in Figure 3 of the previous section may involve allowing two team members to edit synchronously in order to merge two separate parts of the document into one single piece.

Real-time distributed cooperative editing systems allow physically dispersed people to view and edit shared documents at the same time. They are very useful facilities in the rapidly expanding area of groupware and CSCW applications. Research into cooperative editors has been a popular topic in the CSCW community since the mid-80s and many papers have been published in various CSCW related conference proceedings and journals (Ressel et. al. 1996, Sun et. al. 1998).

The goal of our Web-based REDUCE (REal-time Distributed Unconstrained Cooperative Editing) research has been to investigate the principles and techniques underlying the construction of the REDUCE system with the following features (Yang et. al. 2000): (1) real-time - the response to local user actions should be quick (without noticeable delay) and the latency for remote user actions should be low; (2) distributed - cooperating users may reside on different machines connected by the Internet; and (3) unconstrained - multiple users may concurrently and freely edit any part of the document at any time, in order to facilitate free and natural information flow among cooperating users. Our novel underlying technology for maintaining consistency across different sites for unconstrained real-time cooperative editing is very complicated. This has been comprehensively investigated by us in a text editing context (Sun et. al. 1998). In this section, we only illustrate the functionality of the REDUCE prototype which can be easily integrated with our teamwork process environment in order to provide a shared workspace for synchronous collaboration among team members.

The screen snapshot in Figure 4 depicts a synchronous cooperation in action, again as a Java applet. The graphics canvas at the bottom plays the role of a whiteboard which enables team members to draw free style graphical objects, select and draw pre-defined shapes with optional fillings, or input text strings. The text editing panel on the top allows team members to edit the document cooperatively without any constrains, i.e. edit at any position of the text at any time.

7 Supporting process evolution and resource management

Teamwork processes are dynamic entities that need to evolve to take into account changes in technology, in the goals and requirements of the organisation, in the market place, and customers needs. Teamwork is often difficult

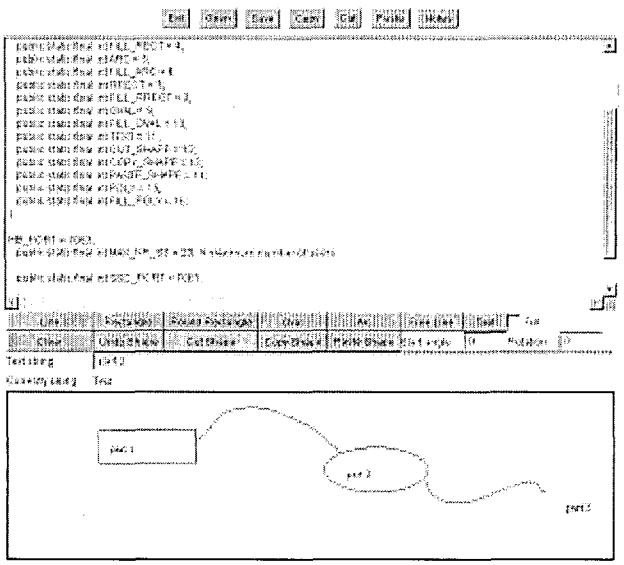


Figure 4: Cooperative editing supported by a whiteboard

to be planned completely in advance. On many occasions, process evolution or change on the fly is very critical to the usability of the process-centred environments for teamwork because it is directly related to teamwork modelling and enactment as we addressed earlier. Some interesting work has been done such as (Kaba & Derniame 1996) and we have investigated this issue intensively (Yang & Zhang 1997b). In general, process evolution may be requested for any process tasks. When a change is requested, depending on its status (enacted, enacting, unenacted), corresponding actions need to be taken to accommodate the change appropriately.

Based on the process repository using a database system, we illustrate our effective process evolution mechanism by an example. For instance, at the current time, if some error is detected which is rooted to task A, it is necessary to roll back to the start point of task A to un-do and then re-execute/re-do it in the new context. To enable correct rollback, it is necessary to maintain a process write-log to keep track of what has been done. In addition, task A may have some impact on the follow-on tasks. For those following-on enacted and enacting tasks, depending on the circumstances, if they are affected by task A or in another word, dependent upon task A, they need to be rolled back and re-executed. However, we can re-use unaffected tasks in order to achieve incrementality because roll-back and re-execution are expensive, especially when human resources are involved. To support such an incremental behaviour, it is essential to carry out a dependency analysis for each task. Clearly it is unnecessary to worry about unenacted tasks which will be executed based on the up-to-date process context eventually. Therefore, the change of unenacted tasks can be directly carried out individually.

Resource management is another important issue for teamwork modelling and enactment, given the nature of global teamwork in which resources including team members, documents, and hardware/software are so dynamic. To better facilitate management of resources in an automatic or semi-automatic fashion, we have proposed to use a trader which can handle dynamic resource attributes effectively (Yang & Ni 1997a). Our mechanism is based on the following reasonable assumptions: 1) there is a dynamic resource pool which includes all types of resources; 2) all resources have registered with a trader, i.e., exported to the trader as service offers; and 3) there are a number of system tools from which the dynamic attribute values can be obtained.

Given the space limit of the paper, we only address our mechanism briefly. In Sections 3 and 5, we mentioned the process engine which is the core part of a process-centred environment to coordinate and manage the process execution. For example, normally the process engine inputs a teamwork process description and allocates required resources to tasks according to the executing order defined in the task description. After completion of a task, the process engine releases the resources, resets any changed resource status, and then starts the next task(s). The process engine interacts with the trader to find available resources, i.e., importing. It is also responsible for updating the information held in the trader, for instance, modifying or withdrawing a service offer.

8 Conclusions and future work

In this paper, we have described an effective semi-centralised multi-tiered architecture for Web-based global teamwork support and indicated some of our prototypical work for carrying out a teamwork process in a computer-mediated automatic fashion. The issues involved include visualised teamwork modelling and enactement for asynchronous coordination with dynamic process evolution and resource management, as well as for synchronous collaboration. In this paper, we have also illustrated corresponding innovative mechanisms to implement the proposed effective architecture for teamwork support. With the fine-grained Web/Java approach, we can dramatically reduce the costs and delays associated with distributed information; provide up-to-date (Internet-based cooperation) software tools without local installation; and offer a simple, extensible and standard (platform-independent) environment.

In the future, we need to further improve Web-based visualised teamwork support mechanisms and the reactive system paradigm as the underlying technology. We also need to further evaluate our current outcomes so as to improve the architecture adopted. For the teamwork support environment in general, many things can be investigated such as better mobility, interoperability, security and tool integration.

Acknowledgement

The work reported here is carried out primarily when the author was working at Deakin University. This project has been supported partially by several ARC (Australian Research Council) grants and seeding grants from School of Computing and Mathematics, Deakin University since 1996. We are grateful for some implementation support from P. Wojcieszak, D. Zhang, P. Jeffers and D. Brain.

References

[1] Ambriola V., Conradi R. & Fuggetta A. (1997) Assessing process-centred software engineering environments. *ACM Transactions on Software Engineering and Methodology*, 6, 3, p. 283-328.

[2] Bentley R., Horstmann T., Sikkel K. & Trevor J. (1995) Supporting collaborative information sharing with the World Wide Web: the BSCW shared workplace system. *Proc. of the 4th WWW Conference*, www.w3.org/Conferences/WWW4/Papers/151/.

[3] Chen C., Zhou W. & Yang Y. (1999) Coordination mechanisms for the teamwork support. *Proc. of 1999 Asia Pacific Decision Sciences Institute Conference*, Shanghai, China, p. 389-391.

[4] Evans E. & Rogers D. (1997) Using Java applets and CORBA for multi-user distributed applications. *IEEE Internet Computing*, 1, 3, p. 43-55.

[5] Feiler P. H. & Humphrey W. S. (1993) Software development and enactment: concepts and definitions. *Proc. of the 2nd Int. Conf. on Software Processes*, Berlin, Germany, p. 28-40.

[6] Fuggetta A. & Ghezzi C. (1994) State of the art and open issues in process-centred software engineering environments. *The Journal of Systems and Software*, 26, p. 53-60.

[7] Gorton I. & Motwani S. S. (1996) Issues in cooperative software engineering using globally distributed teams. *Information and software technology Journal*, 38, 10, p. 647-655.

[8] Groiss H. & Eder J. (1997) Workflow systems for inter-organisational business processes. *ACM SIGGROUP Bulletin*, 18, 3, p.23-26.

[9] Gruhn V. & Urbainczyk J. (1998) Software modelling and enactment: an experience report related to problem tracking in an industrial project. *Proc. of the 20th Int. Conf. on Software Engineering*, Kyoto, Japan, p. 13-21.

[10] Grundy J. C., Apperley M., Hosking J. G. & Mugridge W. B. (1998) A decentralised architecture for software process modelling and enactment. *IEEE Internet Computing*, 2, 5, p. 53-62.

[11] Harel D., Lachover H., Naamad A., Pnueli A., Politi M., Sherman R. & Shtul-Trauring A. (1990) STATEMATE: a working environment for the development of complex reactive systems. *IEEE Transactions on Software Engineering*, 16, 4, p. 403-414.

[12] Kaba A. B. & Derniame J-C (1996) Modelling processes for change: basic mechanisms for evolving process fragments. *Software Process Technology*, Lecture Notes in Computer Science, 1149, p. 99-107.

[13] Ly E. (1997) Distributed Java applets for project management on the Web. *IEEE Internet Computing*, 1, 3, p. 21-26.

[14] Oreizy P. & Kaiser G. (1997) The Web as enabling technology for software development and distribution, *IEEE Internet Computing*, 1, 6, p. 84-87.

[15] Ressel M., Nitsche-Ruhland D. & Gunzenhauser R. (1996) An integrating, transformation oriented approach to concurrency control and undo in group editors. *Proc. of ACM Conference on CSCW*, Boston, USA, p. 288-297.

[16] Scacchi W. & Noll J. (1997) Process-driven Intranets: life-cycle support for process reengineering. *IEEE Internet Computing*, 1, 5, p. 42-49.

[17] Sheth A. (1997) Workflow and process automation in information systems: state-of-the-art and future directions. *ACM SIGGROUP Bulletin*, 18, 1, p. 23-24.

[18] Sun C., Jia X., Zhang Y., Yang Y. & Chen D. (1998) Achieving convergence, causality-preservation, and intention-preservation in real-time cooperative editing systems. *ACM Transactions on Computer-Human Interaction*, 5, 1, p. 63-108.

[19] Yang Y. & Ni Y. (1997a) Resource management with trader for distributed software processes. *Proc. of Int. Symp. on Future Software Technology*, Xiamen, China, p. 63-68.

[20] Yang Y. & Zhang Y. (1997b) A process evolution mechanism using prevalent databases as process repository. *Proc. of IASTED Software Engineering Conf.*, San Francisco, USA, p. 40-44.

[21] Yang Y. (1998) Issues on supporting distributed software processes. *Software Process Technology*, Lecture Notes in Computer Science, 1487, p. 243-247.

[22] Yang Y. & Wojcieszak P. (1999a) Visual programming support for coordination of Web-based process modelling. *Proc. of the 11th Int. Conf. on Software Engineering and Knowledge Engineering*, Kaiserslautern, Germany, p. 257-261.

[23] Yang Y., Zhang D. & Wojcieszak P. (1999b) Coordination management with two types of databases in a

Web-based cooperative system for teamwork. *Proc. of the 1st International Symposium on Database, Web and Cooperative Systems*, Baden-Baden, Germany, p. 47-52.

[24] Yang Y., Sun C., Zhang Y. & Jia X. (2000) REDUCE approach to achieving high responsiveness in Internet-based cooperative systems. *IEEE Internet Computing*, to appear.

[25] Zhou W. & Eide E. (1998) Java sensors and their applications. *Australian Computer Science Communications*, 20, 1, p. 345-356.

Access Skew Detection for Dynamic Database Relocation

Toyokazu Akiyama, Takahiro Hara, Kaname Harumoto, Masahiko Tsukamoto and Shojiro Nishio
Dept. of Information Systems Engineering, Graduate School of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan
TEL: +81-6-6879-7820, FAX: +81-6-6879-7815
E-mail: {akiyama,hara,harumoto,tuka,nishio}@ise.eng.osaka-u.ac.jp

Keywords: DB-migration, distributed database management, transaction processing, access pattern

Edited by: Yanchun Zhang, Vladimir Fomichov and Anton P. Železnikar
Received: August 12, 1999 **Revised:** November 6, 1999 **Accepted:** November 16, 1999

Due to the recent development of network technologies, broader channel bandwidth is becoming available everywhere in the world-wide networks. Based on this fact, we have proposed a new technology that makes good use of such broad bandwidth by dynamically relocating the databases through networks, which we call database migration. *In the method proposed previously, it was assumed that we know the sequence of accesses to the system, this is used for database relocation. However, in order to use database migration in a practical environment, it is necessary to detect the access skew. In this paper, we propose a database relocation method which detects access skew from the access information. Moreover, we examine the effectiveness of the proposed method by comparing it with the conventional database-fixed method.*

1 Introduction

Due to the recent development of network technologies including fiber optic cables and switching technologies, very high-speed data transmission in the order of gigabits/second is becoming available. The revolution in broadband networks affects the design of database management systems. The most important issue for performance improvement of a network-wide database system is how to use the network bandwidth effectively. This is contrary to conventional systems where minimization of the volume of data transmitted in (narrowband) networks has been considered as the primary factor for performance improvement.

Here, the question is, how we can make efficient use of the bandwidth available in broadband networks? A feasible answer is to make databases migrate from one site to another site through networks, which we call *DB-migration*. DB-migration can be performed quickly in broadband networks. For example, if the available bandwidth is one gigabit/second, a database which is 100 megabytes in size can be transferred only in 8.0×10^{-1} seconds. Therefore, the dynamic relocation of databases employing DB-migration can be practically used for several purposes such as *transaction processing*.

In the conventional distributed database environment, each database is fixed at a particular site and a typical database operation is performed through several *operation request messages*. Let us refer to such a fixed-database method as the *fixed-processing method*. On the other hand, if we use DB-migration, there is no need for exchanging messages since the transaction initiation site can have the necessary remote databases migrate to the site. Let us refer to such a method based on DB-migration as the *migration-processing method*.

Based on the idea mentioned above, we have proposed transaction processing methods, which take advantage of DB-migration in broadband networks(Hara et al.,1998a). The proposed methods choose the most efficient of the fixed-processing method and the migration-processing method. We have performed simulation study and confirmed the effectiveness of the proposed methods. Furthermore, we have proposed a distributed database management system based on these methods(Hara et al., 1998b).

However, the proposed methods only give an outline of the system functionality, the methods in (Hara et al., 1998a) were based on some unrealistic assumptions such as that the access pattern of the future transactions is known. In a practical environment, the access pattern of the future transactions is hardly ever known. Thus, in order to use database migration in a practical environment, it is necessary to detect the access skew. In this paper, we propose a method which detects the access skew by using the *access information*. Here, the access information means the information of the successive transactions which were most recently finished. Furthermore, we examine the proposed method by simulation experiments.

2 Previous methods

The methods proposed in (Hara et al., 1998a) are the *simple method* and the *log-statistics method*, which adaptively choose either the fixed-processing method or the migration-processing method. In this section, we explain

the outlines of these two methods and then describe their problems.

2.1 Simple method

In the simple method, first, the communication time of the fixed-processing method and the migration-processing method are estimated. Then, the method which gives the shortest communication time is chosen for executing the transaction.

2.2 Log-statistics method

Although the simple method chooses the most efficient method to process a transaction, this choice may be inefficient in the long run. For example, even if the fixed-processing method is estimated to be efficient for processing a single transaction; it is considered to be more efficient to use the migration-processing method at the beginning of transaction processing if the transaction initiation site continuously initiates transactions which use the same databases. In contrast, even if the migration-processing method is chosen, it is considered more efficient to employ the fixed-processing method if the databases involved in the transaction will be used continuously by the site where the databases currently reside.

The above consideration led us to propose the log-statistics method, where the following two points were considered:

- Each database should reside in the site which uses the database most frequently (the most recent use of the database should be given high priority when deciding its location).
- If it is known that a site S_I uses the database D_j continuously, the priority of S_I to have D_j should be increased.

The log-statistics method chooses the processing method based on both (i) the communication time, estimated in the same way as in the simple method and (ii) the transaction access pattern given by using the *continuous-use declaration*. This is associated with a database when the transaction initiation site uses the database continuously in the next transaction.

2.3 Problems of the previous methods

When we use these methods in a practical environment, we are facing the following problems:

- In the simple method and the log-statistics method, the processing method is chosen at the beginning of the transaction. However, in a practical system, queries included in a transaction cannot be completely known apriori because the transaction may include some conditional branches. Therefore, it is difficult to estimate the communication time of all queries in the transaction at the beginning of the transaction.
- In the log-statistics method, the continuous-use declaration is associated with a database when the transaction initiation site uses the database continuously in the next transaction. However, it is usually unknown whether the next transaction uses the same database or not.

Owing to these problems, both methods work well only when the access pattern of the transaction is known in advance. Furthermore, in these methods, all databases involved in a transaction migrate to the transaction initiation site when the migration-processing method is chosen. This may cause an inefficient migration, i.e., a migration of a database which is rarely used by the transaction initiation site.

3 Access skew detection method

In this section, to resolve the problems described in section 2.3, we propose a database relocation method, which we call the *access skew detection (ASD) method*. This method chooses the processing method using the access information of the most recently committed transactions. Moreover, this method decides the migration of each database individually, thus avoiding inefficient migrations.

In the ASD method, access information is prepared for each database and recorded at the site which holds the database. If access skew is detected from the access information, the system expects that future successive transactions will be initiated at the same site, and makes the database migrate to the site. When a database migrates to the other site, its access information is also transferred. In the following, we describe the detail of the access information and its maintenance process.

3.1 Record of access information

In this subsection, we describe how to record the access information, which will be used to detect access skew. The access information is recorded for each database. The attributes of the access information are as follows:

S: the site that initiates the most recent transaction.

P_A: the total number of pages accessed by the successive transactions.

Q: the total number of queries included in the successive transactions.

At the end of the currently processing transaction, the information of the current transaction is compared with the access information which is recorded for each database involved in the current transaction. If S in the access information is equal to the transaction initiation site, the number

of pages accessed by the current transaction is added to P_A and the number of queries included in the transaction is added to Q. If S in the access information is not the transaction initiation site, we replace the old access information with the information of current transaction as new access information.

3.2 Decision of DB-migration

Similar to the previous methods in (Hara et al., 1998a), at the beginning of a transaction, it is decided whether DB-migration is performed or not. While, in the previous methods, all databases involved in the transaction migrate to the transaction initiation site when the migration-processing method is chosen, in this method, the migration of each database is decided individually. This enables the system to perform effective migrations when the access patterns of databases are different each other.

The decision of DB-migration is executed in the following two steps. Here, these steps are applied to each database which is involved in the current transaction and does not reside in the transaction initiation site.

(1) In the case that the current transaction is initiated from the same site which is recorded in the access information as S, we estimate the communication time of the previous successive transactions from S. The estimation is done both using the fixed-processing method (T_{fix}) and the migration-processing method (T_{DB}), respectively.

Here, T_{fix} and T_{DB} are estimated as in the following equations:

$$T_{fix} = P_A \cdot D_T + 2Q \cdot D_P$$

$$T_{DB} = P_{DB} \cdot D_T + 3D_P$$

P_A : the number of pages accessed by the successive transactions (the attribute in the access information).
P_{DB} : the number of pages composing the target database.
D_T : the delay for sending a page out to the network (transmission delay).
D_P : the delay for transferring data to the remote site (propagation delay).

The first term of T_{fix} formulates the transmission delay of request/reply messages for processing the queries and the second term formulates the propagation delay of the messages. The first term and second term of T_{DB} formulate the transmission delay and the propagation delay of the migration-processing method, respectively. Here, the migration-processing method includes three data transmissions; (i) the request message transmission, (ii) the database transmission, and (iii) the completion notification message transmission. The second term is dominant in T_{fix}, and the first term is dominant in T_{DB}.

(2) If T_{fix} is larger than T_{DB}, the target database migrates to the transaction initiation site. This indicates that when the total communication time of the previous successive transactions which were initiated from the same site becomes shorter by processing them using the migration-processing method, it is concluded that there exists an access skew and further transactions will be initiated from the same site. Thus it is decided to perform the migration of the database.

4 Evaluation

In this section, we present simulation results regarding the performance evaluation of the proposed ASD method. We compare the ASD method with the conventional fixed-processing method and the migration-processing method. We use the queueing model in the simulation experiments. The effectiveness of applying the queueing model to the analysis of the distributed database is described in (Jenq et al., 1988) and (Sheikh et al., 1997).

4.1 Simulation environment

We compare the average response time of the three methods. During DB-migration, we can concurrently access the database using the concurrency control method proposed in (Hara et al., 1998a). Since the cost of managing access information is small, we neglect the cost in the simulation experiments. The cost of the completion notification of DB-migration is calculated as the broadcasting cost in the network. For the purpose of simplicity, we set the cost equal to the point-to-point communication cost.

Now, we explain the three methods which we compare in the simulation experiments.

MIG: The method where all transactions are processed using the migration-processing method. At the beginning of a transaction, all databases involved in the transaction migrate to the transaction initiation site.

FIX: The conventional method that all transactions are processed using the fixed-processing method.

ASD: The proposed ASD method that adaptively chooses either the fixed-processing method or the migration-processing method.

Since the methods proposed in (Hara et al., 1998a) assume that the access pattern of transactions is known in advance which is different from the assumption set in this paper, we do not compare the ASD method with those methods.

When a target database resides in a remote site, it is assumed that only subqueries, which access the target database are sent to the site where the database exists. In the simulation experiments, we assume the *main memory database* (DeWitt et al., 1984), (Garcia-Molina et al., 1984), (Hara et al., 1997). A read operation only accesses main memory and a write operation accesses a main memory and a disk for recording an update log. The number of sites is fixed to 3 since the performance of the ASD method does not depend on it. For the same reason, the number of databases is set to 1. We define the unit of I/O as a page, and 1 page is equal to 8192 bytes. The size of the databases is shown in Table 1.

The access patterns of transactions are represented by parameters in Table 1. Clients in each site initiate transactions at the arrival interval based on the exponential distribution. The average arrival interval is changed in a specific period to create the access skew. Since the intensive accessing site is determined randomly when the interval is changed, the total number of accesses from each site becomes almost the same in the long run. Therefore, the performance of FIX does not depend on the location of the database. In this simulation, the initial location of the database is fixed in every method. The arrival interval ratio represents the ratio of the arrival interval from the intensive accessing site and the other sites. The access page ratio is the ratio of the pages accessed by a query and the total pages in the database. The number of pages accessed in each query is selected randomly from 1 to a specific number. A write operation is initiated every 3 transactions. The access page ratio of a write operation is fixed to 1/50.

4.2 Experiments and results

The relation between the bandwidth and the average response time of each method is shown in Fig. 1. The size of the database is set to 100 Mbytes (12220 pages). Since the response time of MIG at 10Mbps is very large (about 58 sec), it is omitted in Fig. 1. From these simulation results, ASD gives good performance for any bandwidth. When the value of the bandwidth is set to more than 100Mbps, ASD gives the best performance. Under narrow bandwidth, even if transactions are successively initiated, since DB-migration takes long time, the migration-processing method is never chosen. Therefore, ASD and FIX give almost the same performance. However, the average response time of ASD is slightly larger than that of FIX because extreme successive transactions which are incidently initiated, cause an inefficient choice of the migration-processing method and degrades the performance. When the bandwidth is larger than 600 Mbps, MIG gives better performance than FIX.

The relation between the size of the database and the average response time of each method is shown in Fig. 2. The bandwidth is set to 400 Mbps (164 μsec/page). In this simulation experiment, ASD also gives good performance for any database size. When the size of the database

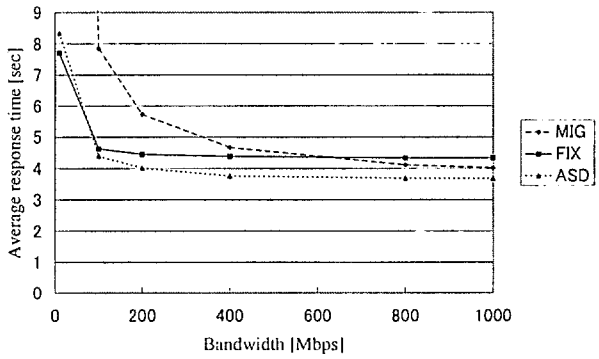


Figure 1: Average response time and bandwidth

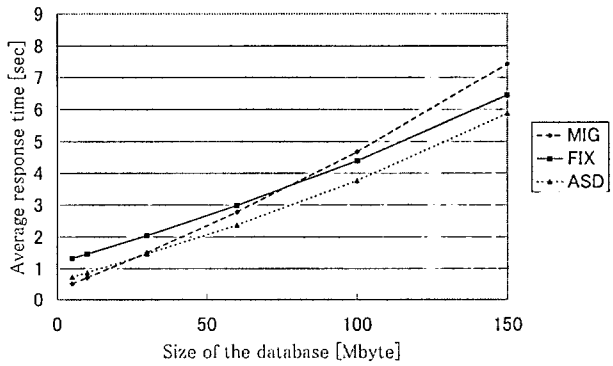


Figure 2: Average response time and size of the database

is very small, the time for DB-migration becomes almost the same as the time for transmitting request/reply of a query. Therefore, MIG, in which the database always migrates, gives better performance than ASD. When the size of the database is large, MIG gives worse performance than FIX. Although we omit the results where the size is larger than 150 Mbytes, when the size of the database becomes about 300 Mbytes, the response time of FIX becomes almost the same as that of ASD. This is because ASD never chooses the migration-processing method when the size of the database is very large.

These results show that the ASD method enables the system to choose the method which has the shorter response time between FIX and MIG. However, the ASD method could not minimize communication delays when successive transactions are initiated concurrently from different sites accessing the same database, since the ASD method could only detect successive transactions from a single site. Therefore, some extensions should be considered to our proposed method in order to deal with concurrent accesses from different sites.

5 Conclusion

In this paper, we have proposed a database relocation method which adaptively decides whether to perform DB-

Parameters	Values
number of sites	3
number of databases	1
size of the database	611 $\sim$ 18311 pages (5 $\sim$ 150 Mbytes)
bandwidth	1G $\sim$ 10 Mbps
(transmission delay (D_T))	(66 $\sim$ 6554 μsec/page)
propagation delay (D_P)	200 msec
memory access speed	10 nsec/page
disk access speed	30 msec/page
average arrival interval	20 sec (intensive access)
	120 sec (normal access)
period to change the arriving interval	400 sec
arrival interval ratio	6
number of queries in a transaction	1 $\sim$ 20
access page ratio	1/20

Table 1: Parameters for the experiments

migration or not by using the access information of transactions. We have also evaluated the effectiveness of the proposed ASD method by simulation experiments. The simulation results show that the proposed method enables the system to perform effective DB-migrations even if the access pattern of the transactions is unknown.

As part of our future work, we are planning to implement our proposed method on the existing system and evaluate it in various aspects. Furthermore, we are planning to extend our proposed method to deal with concurrent accesses from different sites.

References

[1] DeWitt, D., Katz, R., Olken, F., Shapiro, L., Stonebraker, M., & Wood, D.(1984): Implementation Techniques for Main Memory Database Systems. *Proceedings of ACM SIGMOD*, 1-8.

[2] Garcia-Molina, H., Lipton, R.J., & Valdes, J.(1984): A Massive Memory Machine. *IEEE Transaction on Computing*, Vol.C-33, 391-399.

[3] Hara, T., Harumoto, K., Tsukamoto, M., & Nishio, S.(1997): Main Memory Database for Supporting Database Migration. *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (IEEE PACRIM '97)*, Vol.1, 231-234.

[4] Hara, T., Harumoto, K., Tsukamoto, M., & Nishio, S.(1998a): Database Migration: A New Architecture for Transaction Processing in Broadband Networks. *IEEE Transaction on Knowledge and Data Engineering*, Vol.10, No.5, 839-854.

[5] Hara, T., Harumoto, K., Tsukamoto, M., & Nishio, S.(1998b): DB-MAN: A Distributed Database System Based on Database Migration in ATM Networks. *Proceedings of 14th International Conference on Data Engineering*, 522-531.

[6] Jenq, B., Kohler, W. H., Towsley, D.(1988): A Queueing Network Model for a Distributed Database Testbed System. *IEEE Transaction on Software Engineering*, Vol.14, No.7, 908-921.

[7] Sheikh, F., Woodside, M.(1997): Layered Analytic Performance Modeling of a Distributed Database System. *Proceedings of 17th International Conference on Distributed Computing Systems*, 482-490.

On Incremental Global Update Support in Cooperative Database Systems

Chengfei Liu
School of Computer and Information Science, Uni. of South Australia, Adelaide,SA 5095 Australia
Phone: +61 8 8302 3287, Fax: +61 8 8302 3381
E-mail: chengfei.liu@unisa.edu.au
AND
Xiaofang Zhou
Dept. of Computer Sci. & Electrical Engineering, Uni. of Queensland, Brisbane QLD4072 Australia
Phone: +61 7 3365 3248, Fax: +61 7 3365 4999
E-mail: zxf@csee.uq.edu.au
AND
Jinli Cao
Dept. of Mathematics & Computing, Uni. of Southern Queensland, Toowoomba QLD4350 Australia
Phone: +61 7 4631 1619, Fax: +61 7 4631 1775
E-mail: cao@usq.edu.au
AND
Xuemin Lin
School of Computer Sci. & Engineering, Uni. of New South Wales, Sydney NSW2052 Australia
Phone: +61 2 9385 6493, Fax: +61 2 9385 5995
E-mail: lxue@cse.unsw.edu.au

Keywords: Multidatabases, Global Transaction Management, Transaction Classification

Edited by: Yanchun Zhang, Vladimir Fomichov, and Anton P. Železnikar
Received: August 15, 1999 **Revised:** November 10, 1999 **Accepted:** November 15, 1999

OzGateway is a cooperative database system designed for integrating heterogeneous existing information systems into an interoperable environment. It also aims to provide a gateway for legacy information system migration. This paper summarises the problems and results of multidatabase transaction management research. In supporting global updates in OzGateway in an evolutionary way, we introduce a classification of multidatabase transactions and discuss the problems in each category. The architecture of OzGateway and the design of the global transaction manager and servers are presented.

1 Introduction

Data management technology has been evolving rapidly. While modern database management systems (DBMSs) and client-server based distributed computing environments are available now, most large organisations are still using out of fashion technologies (e.g., COBOL, pre-relational DBMSs) and mainframe computers to manage their data. It is currently a high priority task to find methodology of integrating existing information systems to meet the application requirements in an interoperable environment, and the ways for making legacy information systems (ISs) reusable by converting them into target ISs which use new database technologies [Brodie and Stonebraker, 1995].

A multidatabase system (MDBS) is a software interface to provide users uniform access to multiple, heterogeneous database systems. The component systems of an MDBS may include DBMSs of different designs and models, and possibly some file systems. In spite of different and possibly redundant and conflicting representations of objects using different models at different locations, a user can access information as if he/she is using a single centralised DBMS at the multidatabase level. Major issues in current multidatabase research include schema integration and global transaction management [Hurson et al., 1994].

OzGateway is a cooperative database system which aims to provide solutions for integrating heterogeneous existing information systems into an interoperable environment. It also aims to be a gateway system for legacy information system migration. In this paper, we discuss global update support in OzGateway. We know that supporting global update in multidatabases is very difficult. However, in many multidatabase applications there are no global constraints at all, since each site was developed independently and may wish to remain independent. Because of this, we introduce a classification of multidatabase transactions and discuss the problems for each catagory of transactions.

The rest of the paper is organised as follows: In section 2, problems and results in current multidatabase transaction management are summarised. The classification of multidatabase transactions is introduced in Section 3. In section

4, the architecture of OzGateway and a preliminary design of the GTM and servers are presented. Section 5 concludes the paper.

2 Multidatabase Transaction Management

In an MDBS, global transactions are executed under the control of the MDBS, and at the same time local transactions are executed under the control of the local DBMSs. Each local DBMS may employ a different transaction management scheme (or even no transaction management at all). In addition, each local DBMS has complete control over all transactions (both global and local) executing at that site, including the ability to abort any transaction at any point at its site. Typically, no design or internal DBMS structure changes are allowed in order to accommodate the MDBS. Furthermore, the local DBMSs may not be aware of each other and, as a consequence, cannot coordinate their actions. These issues make the transaction management in MDBS very difficult.

A local DBMS offers a set of operations, which can be classified into two classes: one deals with transaction operations (such as *transaction_begin*, *transaction_end*, *abort*, *commit*, *prepare_to_commit*), another deals with transaction status information (such as *get_wait_for_graph*, *get_serialization_order* and *get_transaction_status*). In general, a local DBMS does not export its *wait_for* graph or *serialisation* order when participating the MDBS. If a local DBMS only allows a database user (the MDBS transaction manager is nothing more than a local DBMS user to the LDBMS) to submit a transaction as a whole (i.e., from *transaction_begin* to *commit*), the MDBS has no control as to when it is executed.

In multidatabase transaction management three problems remain to guarantee [Breitbart et al., 1992]: *global serialisability*, *atomicity of global transactions*, and *deadlock-free executions of global transactions*. The presence of local transactions could make a serial execution of global transactions not satisfy global serialisability as some possible invisible relationship among global transactions could be introduced by local transactions. Global serialisability can be achieved by forcing otherwise invisible conflicts by letting transaction T_1 write some objects on site s and letting T_2 read these objects if T_1 proceeds T_2 (denoted as $T_1 \rightarrow T_2$) in the global serialisability graph and site s is involved in the execution of both transactions [Georgakopolous et al., 1991]. Many other methods have also been proposed for global serialisability: *strongly serialisable scheduling* [Alonso et al., 1987] given all local systems use the basic timestamping order, *serialisation-point* [Pu, 1988] (e.g., the first operation in the timestamping scheme, the first lock release in 2PL, the last operation *commit* in *strongly recoverable scheduling* [Breitbart et al., 1991]), scheduling based on *rigorous* local DBMSs [Breitbart et al., 1991], $\epsilon - serialisability$, *two level serialisability* [Mehrotra et al., 1991], etc.

Atomicity of global transactions is difficult to support as a local DBMS does not have any obligation to the global transaction execution coordinated by the GTM, i.e., it does not usually export the *prepare_to_commit* operation, it can abort its local branch of a global transaction unilaterally at any time before commit, therefore the 2PC protocol can not be used. A *server*, when the local DBMS does not support the *prepare_to_commit* operation, can be used to participate in the global 2PC protocol on behalf of the local system. When a server votes to commit a global transaction but the LDBMS aborts the global subtransaction, the server has to *redo* or *retry* the global subtransaction. Another approach is the *Compensate* approach, which allows the GTM to semantically undo any committed global subtransactions.

Similarly, deadlock-free executions of global transactions are hard as the GTM cannot access the *wait_for* graph for local transactions. Based on communication ordering and time-out, an approximation global *wait_for* graph is used by the GTM to detect all deadlocks but also some false deadlocks.

All the above methods may result in poor performance or bring in some restrictions in use. It is very difficult, if not impossible, to give satisfactory solutions to these global transaction management problems without sacrificing local autonomy.

3 A Classification of Transactions

There are two types of transactions in an MDBS: global and local transactions. The fundamental distinction is by the data they access. A data item is global if it can be seen at the global level (i.e., that data item has been exported by the local system and been imported by the global system); otherwise it is local. A transaction is local to an LDBMS if it uses local data only, or uses local data and global data exported from the local system. A transaction is global if it uses only global data. Note that it is not a valid multidatabase transaction if it access a mixture of local data and global data from other local systems. The MDBS is not aware of any local transactions, but a local transaction can use global data which is exported from the local system. This is where most problems of multidatabase transaction management arise from.

From our discussion in the above section, one can see that it is very hard for the GTM to support global updates if local sites do not provide at their interface level sufficient transaction control commands or internal transaction status information. Most global transaction management solutions are based some assumptions about local systems. If any of the local sites cannot meet the requirement, the global transaction management has to sacrifice performance to use a less efficient method to compromise with that local system. Given that a multidatabase system

consists of a large number component systems, which may join or withdraw from the multidatabase system on their own merits, an MDBMS may have to always be based on the weakest assumption about local systems. This leads to poor performance.

Another problem of global update comes from schema integration. A multidatabase relation is often integrated from several local relations. To maintain local autonomy, such an integration is usually a view integration (as opposed to database integration which physically merges relations into a single global relation). In other words, such a global relation is a view defined from joining several relations; thus, it cannot always be updated. Another example of not being able to update a global relation comes from the common practice in multidatabase integration of applying some transformation rules on the local data items. For example, in order to integrate some component systems which store a piece of land using its geometry data, and some other component systems which store land areas, the global data can use areas. A simple transformation rule can be applied to those systems using geometry data. It is clear that the update of area in the global relation is impossible as such an update cannot be done in those component systems using geometry data.

We have seen that global update in an MDBS is difficult, and is not always possible. Now we discuss from a positive perspective: is this necessary? Given that the GTM can become very simple and much more efficient if global update is not to be supported or partly supported, it should be carefully decided whether it is worthwhile to support global update. We introduce a classification of transaction here. Which transactions an MDBMS should support, and how they are supported will be discussed later.

Transactions in an MDBS can be classified into the following categories:

1. *SWSR (Single-Write, Single-Read) transactions:* They read and update data from the same local system. Such transactions can be either global or local.

2. *SWMR (Single-Write, Multiple-Read) transactions:* They read data from multiple sites, but only update data from one site. They are global transactions.

3. *MWSR (Multiple-Write, Single-Read) transactions:* They read data from one single site, but update multiple local systems. They are global transactions.

4. *MWMR (Multiple-Write, Multiple-Read) transactions:* They read as well as update data from multiple sites. They are global transactions.

From the transaction management point of view, there is no difference between MWSR transactions and MWMR transactions. Thus we only discuss the MWMR transactions.

Many applications only need to share information among sites with the agreement that data can only be updated by the owner. Recent research on providing an integrated view of a variety of legacy data also falls into this category [Roth and Schwarz, 1997, Haas et al., 1997, Levy et al., 1996]. There are two ways to facilitate updating of data by its owner. The simplest way is to *withdraw* data from MDBS for update [Ahmed et al., 1991]. Therefore, this data cannot be seen at the global level during the update, and the update transaction is a local SWSR transaction. The updated data can join the multidatabase system later. By assuming that each local DBMS can maintain local consistency and handle local deadlock, it is obvious that there will be no need to consider these issues at the global level. However, besides the problem of data availability during the update, a major problem of this approach is that it can be very costly, particularly when the data to be updated are in a very large relation and the local and global data formats are different.

If a local transaction updates local data as well as its exported data, it is still a local SWSR transaction. This is most often the case as old local applications should be able to run without change even if some of its data have been exported to the global system. The GTM may not know the existence of these local SWSR transactions. Therefore, possible effects caused by the presence of such transactions should be considered in the design of GTM. Two global serialisability problems will still occur even without other types of update transactions:
(1) a global retrieval transaction reads dirty data from an aborted local update transaction;
(2) inconsistent reads between two global retrieval transactions due to local update transactions.
The first problem can be solved if each local system employs strict 2PL. The second problem can be compromised if global consistency is not considered. Fortunately, there are no global deadlock and atomicity problems.

While an SWSR transaction can always be regarded as a local transaction, it is also possible to consider an SWSR transaction as global if it accesses only exported the data of a local system. All SWSR transactions can be "globalised" by temporally "globalising" the local data they use. This does not violate local autonomy, because only an interface shell needs to be added to the local system.

Now consider a common case where an application needs to read data from multiple sites to make a decision about how to update its own data. The user may wish to keep the data to be updated in the global system to make their application program simpler. Therefore, an MDBMS should support the global update transactions which read data from multiple sites, but update only the data on their own local systems. From the transaction management point of view, it has no much difference whether the update is on local site or not, as long as only one site is involved for each transaction to update. So this kind of transactions are SWMR transactions. The Oracle Procedural Gateway only supports SWMR transactions [Sandrolini, 1994]. To support this category of transaction, there is no need to implement the 2PC protocol; and atomicity is easy to maintain. However, global deadlock is possible.

The hard applications are those MWMR transactions,

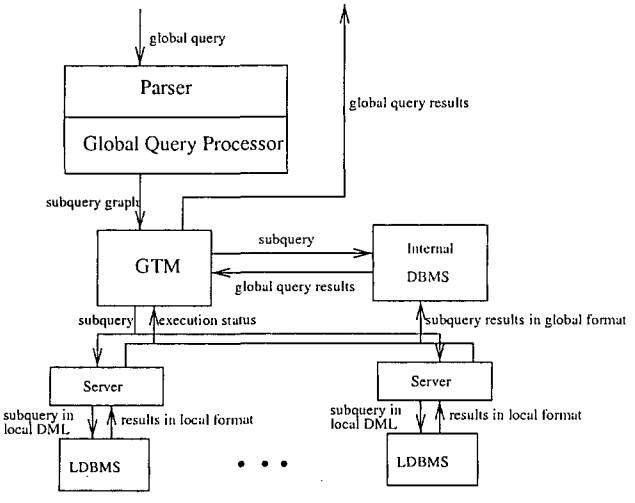


Figure 1: Architecture of OzGateway

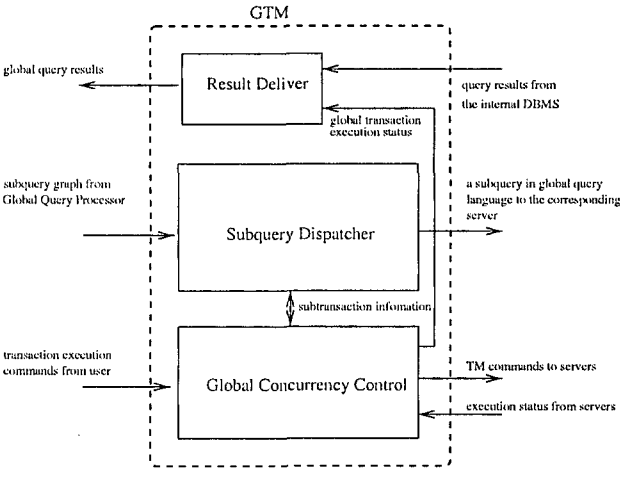


Figure 2: Architecture of GTM

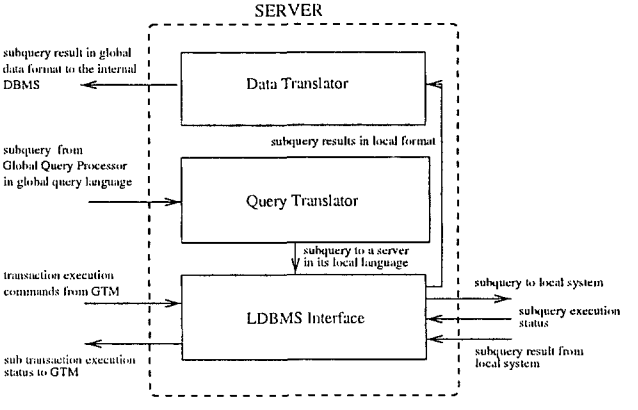


Figure 3: Architecture of a Server

which read and update data managed by multiple DBMSs. All three problems remain in this category of transactions. We do not consider this at the moment as we think this is not often, and global consistency is not important. For example, a travel agent to prepare a travel for a client may need to book air-tickets from multiple airlines, book accommodation from different hotels and book cars from some companies. These bookings are related to each other as some or all bookings may have to be cancelled if other bookings cannot be made. This application needs to update the databases of several airlines, hotels and car rental companies. However, there is no global integrety constraint, as long as each local system is consistent.

4 Design of OzGateway

OzGateway is a cooperative database system which aims to provide solutions for integrating heterogeneous existing information systems into an interoperable environment. It also aims to be a gateway system for legacy information system migration. Figure 1 is the general architecture of OzGateway. A global user issues a global query using a global query language against a global schema. The Global Query Processor decomposes the global query into a set of single-site subqueries, which are organised as a query graph according to data dependency and query processing cost. OzGateway supports applications which may consist of part of legacy systems and part of target systems. One of the subqueries is to be executed at the OzGateway internal DBMS to merge the intermediate results from other subqueries.

The GTM dispatches these subqueries to the servers of the corresponding local systems. Global concurrency control may be considered depending on the category of transactions being supported. A server translates the subquery received into local query language and then passes the translated query to the local system as an ordinary local query. It reports execution status to the GTM, and passes the subquery results, if the execution succeeds, to the internal DBMS after translating the subquery results into the OzGateway common data format. The GTM reports execution failure to the user and aborts all other subqueries on receiving a failure report from any server, or delivers the global query results to the user when the internal DBMS finishes data merging if none of the subqueries fail. The architectures of the GTM and a server are shown in Figure 2 and Figure 3, respectively.

OzGateway supports global updates along an evolutionay path. At the first stage, we are keen to set up a multidatabase environment to enable the user to share information which can otherwise not be shared. Therefore, global update is not allowed (all updates are through local SWSR transactions). At the second stage, we will support SWMR transactions, which are sufficient to support legacy system migration. In time, MWMR transactions may need to be supported; however, this is not considered in our current design. To enable OzGateway to evolve from stage 1 to stage 2 smoothly, we should avoid a total restart when stage 2 begins. This is realised by using the open GTM architecture. We believe that the next stage only needs to substitute a limited number of components.

Sitting between the GTM and a local DBMS, a server

passes the subtransaction and data to the local DBMS, and reports the execution status of the subtransaction and returns result back to the GTM after necessary translation.

To support legacy ISs, a server serves as a gateway which translates requests at the global level to legacy ISs. Functions which cannot be performed by legacy ISs at the local level are either performed by the OzGateway internal DBMS or by the server. In the later case, a server also serves as a wrapper which supports a set of functions required at the global level in terms of local systems, especially for legacy ISs. For instance, it is possible for the server to enhance the local system such that a better GTM can be expected. This is particularly important when the local system is a file system, or a legacy IS. In supporting 2PC, a server can also simulate a *prepare_to_commit* status by redoing or resubmitting the subtransactions aborted by the local DBMS after the server votes to commit.

5 Conclusion

OzGateway is designed as a vehicle to facilitate research in multidatabase systems and legacy information system migration. In this paper, we re-examined the problems in a MDBS environment. To minimize the tasks of the GTM, a classification of multidatabase transactions was introduced and problems in each category were discussed. The general architecture, the GTM, and the servers of OzGateway were presented. Currently, we are investigating various approaches to problems in each transaction category.

References

[Ahmed et al., 1991] Ahmed, R., Smedt, P. D., Du, W., Kent, W., Ketabchi, M., Litwin, W. A., Rafii, A., and Shan, M.-C. (1991). The pegasus heterogeneous multidatabase system. *IEEE Computer*, 24(12).

[Alonso et al., 1987] Alonso, R., Garcia-Molina, H., and Salem, K. (1987). Concurrency control and recovery for global procedures in federated database systems. *Data Engineering*, 10(3):5–11.

[Breitbart et al., 1992] Breitbart, Y., Garcia-Molina, H., and Silberschatz, A. (1992). Overview of multidatabase transaction management. *VLDB J.*, 2:181–239.

[Breitbart et al., 1991] Breitbart, Y., Georgakopolous, D., Rusinkiewicz, M., and Silberschatz, A. (1991). On rigorous transaction schduling. *IEEE Traansaction on Software Engeering*, 17(9).

[Brodie and Stonebraker, 1995] Brodie, M. and Stonebraker, M. (1995). *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach.* Morgan Kaufmann.

[Georgakopolous et al., 1991] Georgakopolous, D., Rusinkiewicz, M., and Sheth, A. (1991). On serializability of multidatabase transaction through forced local conflicts. In *Proceedings of the 7th International Conference on Data Enginerring.*

[Haas et al., 1997] Haas, L., Kossmann, D., Wimmers, E., and Yang, J. (1997). Optimizing queries across diverse data sources. In *Proceedings of the 23rd VLDB Conference.*

[Hurson et al., 1994] Hurson, A. R., Bright, M. W., and Pakzad, S. H. (1994). *Multidatabase Systems: an advanced solution for global information sharing.* IEEE Computer Society.

[Levy et al., 1996] Levy, A., Rajaraman, A., and Ordille, J. (1996). Querying heterogeneous information sources using source description. In *Proceedings of the 22nd VLDB Conference.*

[Mehrotra et al., 1991] Mehrotra, S., Rastogi, R., Korth, H. F., and Silberschatz, A. (1991). Non-serializable execution in heterogeneous distributed database systems. In *Proceedings of the first internatinal conference on parallel and distributed database systems.*

[Pu, 1988] Pu, C. (1988). Superdatabases for composition of heterogeneous databases. In *Proceedings of the 4th International Conference on Data Engeering.*

[Roth and Schwarz, 1997] Roth, M. and Schwarz, P. (1997). Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *Proceedings of the 23rd VLDB Conference.*

[Sandrolini, 1994] Sandrolini, L. (Summar 1994). Remote procedure access to lagacy datastores. *Oracle Magazine*, pages 95–97.

Performance Improvements of Thakore's Algorithm with Speculative Execution Technique and Dynamic Task Scheduling

Takahiro Sasaki, Tetsuo Hironaka and Seiji Fujino
Graduate School of Information Sciences, Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima, 731-3194, Japan
E-mail: {sasaki, hironaka, fujino}@csys.ce.hiroshima-cu.ac.jp
AND
Tsuyoshi Takayama
Faculty of Software and Information Science, Iwate Prefectural University
152-52 Takizawa-aza-sugo, Takizawa, Iwate, 020-0173, Japan
E-mail: takayama@soft.iwate-pu.ac.jp

Keywords: Object-oriented database, query processing, parallel processing, speculative execution, performance improvement, performance evaluation.

Edited by: Yanchun Zhang, Vladimir Fomichov and Anton P. Železnikar
Received: August 15, 1999 **Revised:** November 10, 1999 **Accepted:** November 15, 1999

This paper proposes an approach to improving the performance of Thakore's algorithm with the speculative execution technique and dynamic task scheduling. Recently, object-oriented databases are being used in many applications, and are becoming larger and more complex. As a result, their response time is becoming longer. In order to reduce the response time, many parallel query processing approaches are proposed. The algorithm presented by Thakore et al. in 1995 is one of the representatives of such parallel approaches, but has a problem on load balancing. In order to reduce the problem, we modify this algorithm in two points: (i) introduce the speculative execution technique, and (ii) adopt dynamic task scheduling in assignment between a class and a processing node. These two modifications lead to a performance improvement of the original algorithm. Its effectiveness is shown with some evaluations.

1 Introduction

Recently, object-oriented(OO) databases(DBs) are being used in many applications and are becoming larger and more complex. This causes their response time to become longer. In order to reduce the response time, many parallel query processing approaches are proposed.

The parallel query processing algorithm by Thakore *et al.* (Thakore *et al.* 1995) is one of the representives of such approaches. Hereafter, we call this algorithm "Thakore's algorithm" or "original". Thakore's algorithm has a significant problem on load balancing. It makes it impossible to use a parallel environment effectively.

In order to reduce this problem, we modify the algorithm in two points. These modifications are effective and lead to performance improvement.

The remainder of this paper is organized as follows. In the next section, we summarize some related work. In Section 3, we point out a problem in Thakore's algorithm. In Section 4, we propose two modifications. Section 5 shows its effectiveness with some evaluations. Section 6 concludes our paper and shows some future research directions.

2 Related Work

2.1 Parallel Approaches in OODB

We can classify parallel approaches in OODBs into two categories. One is mainly focused on object placement, and the other mainly focused on query processing algorithm.

Grandeharizadeh *et al.*(Grandeharizadeh *et al.* 1994) argue various object placement with some evaluations. Kim(Kim 1990) points out that following three parallelisms are applicable for query processing in an OODB:

Path parallelism: All different paths in a query graph are processed in parallel.

Node parallelism: All nodes in a query graph, which corresponds to each class, can be processed in parallel.

Class hierarchy parallelism: Instances of different classes in a class hierarchy can be processed in parallel.

All of these parallelisms are relatively coarse grain. Comparing with them, Thakore *et al.*(Thakore *et al.* 1995) propose a finer grained parallel query processing algorithm.

2.2 Thakore's Algorithm

2.2.1 Features

We summarize some features of Thakore's algorithm:

Static assignment of a single class to a single processing node:
A single class is assigned to a single processing node respectively. A processing node retrieves only the objects belonging to the assigned class. Message passing is used to communicate with other classes.

Two types of parallelism: A query is processed in parallel on several processing nodes. Furthermore, a process in a processing node is also parallelized with the thread technique.

Closure property: A query result has a part of structure on the entire DB and a further query can be operated on one or more subdatabases and produces a new subdatabase. Thakore *et al.* are the pioneer in introducing this property into parallel OODB. That's why their work has a significant role in this field.

2.2.2 Outline of Algorithm

The algorithm is composed of the following two phases.

Identification Phase: Each processing node retrieves only the objects belonging to the assigned class. The objects that satisfy some conditions through multiple classes, like the AND or OR operations, are selected by message passing to the other classes.

The algorithm avoids the generation of large amounts of temporary files between the identification phase and the generation phase by marking the selected objects.

Generation Phase: This phase generates a query result from the marked objects in the identification phase, and returns a result to the DB user. In order to satisfy the closure property, the marked objects are sent from a root node to leaf nodes through some edges.

3 A Problem with Thakore's Algorithm

Thakore's algorithm has a significant problem in load balancing. The feature "static assignment" described in Subsection 2.2.1 leads to the following fact: "in the case of DB users' interests concentrate on certain classes, load is apt to concentrate on the corresponding processing nodes". This is not a rare case seen in many DB applications. In such cases, we can say that Thakore's algorithm has a problem in order to use a parallel environment effectively.

4 Solutions

Now we are in a position to present methods of modifying the algorithm. We try two modifications:

1. introduce the speculative execution technique, and
2. adopt dynamic task scheduling in assignment between a class and a processing node.

As mentioned in (Thakore *et al.* 1995), we concentrate on the discussion of the retrieval operation and avoid the discussion for update or lock operation.

4.1 Introduction of Speculative Execution

In parallel environments, "speculative execution" is known as an effective technique(Yamana *et al.* 1995). It executes a sequential process which includes a conditional branch earlier than its corresponding condition is determined.

Examples of introduction of the speculative execution technique into a DB field can be found in (Bestavros & Braoudakis 1995), (Reddy & Kitsuregawa 1998), and so forth. These papers introduce speculation into concurrency control in some DB technologies. Our paper introduces the speculative execution technique not into concurrent control but into query processing.

Before we describe it, we define the term "user-entrusted time (*UET*)" as the sum of the following three times:

1. a DB user reads some guidance sentences for query input,
2. he/she selects one query in his/her mind, and
3. he/she inputs the query.

Using this *UET*, our approach predicts some query conditions which have a relatively high possibility of being inputted, and starts to process the queries speculatively. If one of these queries matches the query he/she actually inputs, the DB system returns the result of the corresponding speculative query processing. The preceding time leads to a performance improvement. If the prediction fails, the DB system starts to process the query in the ordinary manner. In general, since users interests concentrate on a small part of an entire DB, the above-mentioned prediction is not unreal.

Now we proceed to an implementation issue. We consider the following two approaches to process creation for speculative query processing.

(A)Dynamic Execution: Create the processes after a transaction arises. Processes which have done an unsuccessful speculation are stopped after the DB user inputs his/her query.

(B)Static Execution: Create the processes before a transaction arises. These processes stay alive permanently. When a transaction arises, the DB system performs

the following three works; (i)predicts the query conditions which he/she inputs, (ii)reserves some idle processes for speculative query processing, and (iii)sends each predicted query to a reserved process. The process receives the query and starts to process it speculatively. After the process finishes the query processing, it becomes idle again and waits for the next requests.

4.2 Adoption of Dynamic Task Scheduling to Assign a Processing Node

The original algorithm assigns a single class to a single processing node statically and fixedly. It has possibility of causing load unbalancing. We adopt another approach; a single class is dynamically assigned to a single processing node when a transaction arises. It is not a fixed assignment. We call each method:

(a)Static Placement: It is the approach used in Thakore's algorithm, and

(b)Dynamic Placement: It is another approach we adopt in this paper, respectively.

4.3 Four Methods for Implementation

For the combinations of (A), (B) described in Subsection 4.1, and (a), (b) described in Subsection 4.2, we can consider four approaches:

1. *DESP*(Dynamic Execution / Static Placement): ((A)-(a)),
2. *DEDP*(Dynamic Execution / Dynamic Placement): ((A)-(b)),
3. *SESP*(Static Execution / Static Placement): ((B)-(a)), and
4. *SEDP*(Static Execution / Dynamic Placement): ((B)-(b)).

All of them are possible to be implemented, and it is not easy to predict which has the best performance. In the next section, we evaluate them through some experiments.

5 Evaluations

5.1 Methods

Thakore *et al.*(Thakore *et al.* 1995) carry out the evaluations only under a single transaction environment. From a practical point of view, we use multi-transactions environments. Figure 1 shows a DB schema for evaluations, and Figure 2 shows some query graphs. In order to observe the effects based on the differences of query sequences, we use five types of multi-transactions (Figure 3). Each character, '5', 'V', 'X' and 'Y', means a query type in Figure 2. The order of transaction is from left to right. For example, in the case of *Pattern 3*, *Type 5* is the first query and *Type V* is the tenth and last query. We adopt the allocation with random numbers as one of the dynamic task scheduling approaches.

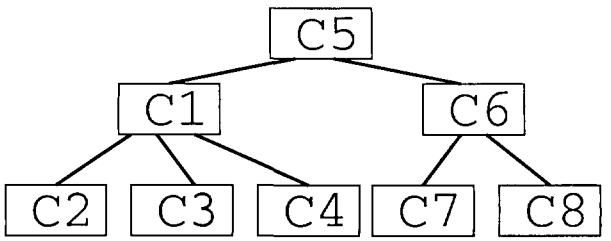


Figure 1: DB schema for evaluations.

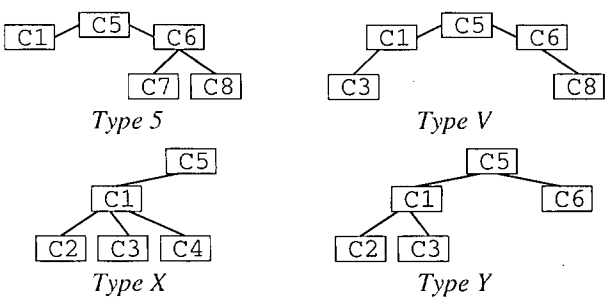


Figure 2: Query graph for evaluations.

Pattern 1:	5, 5, 5, 5, 5, 5, 5, 5, 5, 5
Pattern 2:	5, 5, X, X, V, V, Y, Y, 5, 5
Pattern 3:	5, V, X, Y, 5, V, X, Y, 5, V
Pattern 4:	5, X, Y, 5, Y, V, 5, V, Y, X
Pattern 5:	5, Y, Y, X, 5, 5, X, V, X, X

Figure 3: Five types of multi-transactions.

We carry out the following two experiments:

(1) *UET* vs. "average response time (*ART*)", and

(2) "success ratio of speculation (*SRS*)" vs. *ART*.

5.2 Environment

Figure 4 shows the network topology of the parallel environment for evaluations. A single segment consists of eight

Table 1: Environment.

Number of workstations used as a processing node	30
Workstation	Sun SS-5 110MHz
Operating System	Solaris 2.5.1
Main memory of each workstation	64MBytes
Programming language	C language
Message passing library	PVM 3.3.11

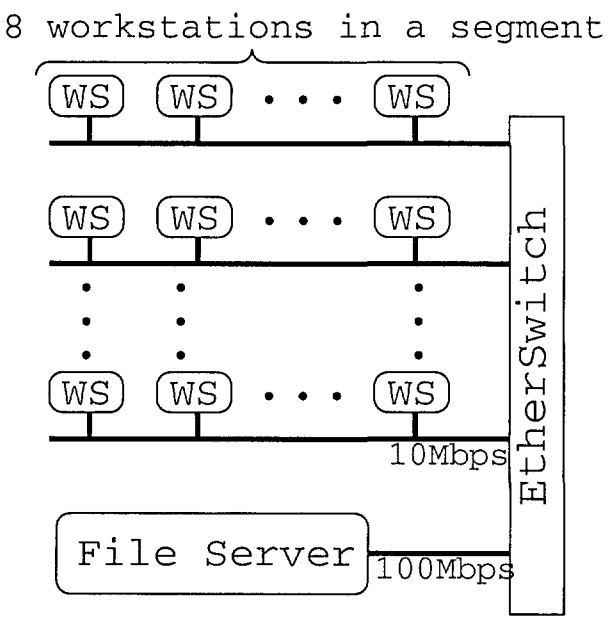


Figure 4: Workstation cluster for evaluations.

Table 2: Parameters and its default values.

Parameter	Default value
Number of objects(*NOO*)	10,000 objects/class
Object identifier size	4Bytes
Attribute size	4Bytes
Selectivity factor*1	0.1
Base data conectivity*2	10
Transaction frequency(*TFR*)	10sec
User-entrusted time(*UET*)	5sec
Success ratio of speculation(*SRS*)	80%

*1 The ratio of the number of objects selected due to the selection conditions and the total number of a given class.
*2 Average number of an associated class related with an object of a given class.

workstations connected via 10Mbps Ethernet. These segments are connected by an EtherSwitch. A file server is connected to the EtherSwitch via 100Mbps Ethernet. The DB is located on the file server, and it connects with each workstation via the EtherSwitch.

We use a simulator which we have implemented, instead of using a real DB management system. This is the same manner as (Thakore *et al.* 1995). The simulator is written in C language and PVM3 as a message passing library. Table 1 summarizes the environment for evaluations. Table 2 shows some evaluation parameters and its default values.

5.3 Results

***UET* vs. *ART* (Figure 5):** We show only the graphs of Pattern 1 and 3, but other patterns also have the following common tendencies. First, in all the time, *DEDP* is the fastest in all cases. Second, all approaches except for the original decreases *ART* depending upon *UET*. Third, *DEDP* is faster than the original even if the *UET* closes to zero. We can guess that since *DEDP* assigns a single class to a single processing node dynamically, it can use more processing nodes and it is easier for it to distribute the load than the original.

***SRS* vs. *ART* (Figure 6):** Although the original has no correlation to *SRS*, we plot it in the figure for reference. We show only the graphs of pattern 1 and 3, but other patterns also have the following common tendencies. First, *DEDP* is the fastest again. Second, in all approaches except for the original, ART decreases depending upon *SRS*. Third, even if *SRS* eqauls 0%, *DEDP* is faster than the original. Here, we note that, *SRS* equals 0 doesn't mean that speculative query processing is not used.

In order to investigate the third nature in detail, we carry out one additional experiment. For original and *DEDP*, we observed the behaviour of *ART* when *SRS* was fixed to zero. In this experiment, we used 180 cases, all possible combinations[1] of the following parameters:

NOO(object/class) = (3000, 5000, 7000, 10000, 30000, 50000),

Multi-transactions types (*MTT*) = (*Pattern 1-5*),

TFR(sec) = (10, 15), and

UET(sec) = (1, 5, 10).

This experiment clarifies the following facts. First, the most sensitive parameter to *ART* is *NOO*. Second, about each parameter, we can observe the following rules:

NOO → **Larger:**
Processing time per a transaction → Longer
Overlap in time of each query processing → Longer
A gain by dynamic load balancing → Larger
Superiority of *DEDP* → Larger

[1] 6 x 5 x 2 x 3 = 180

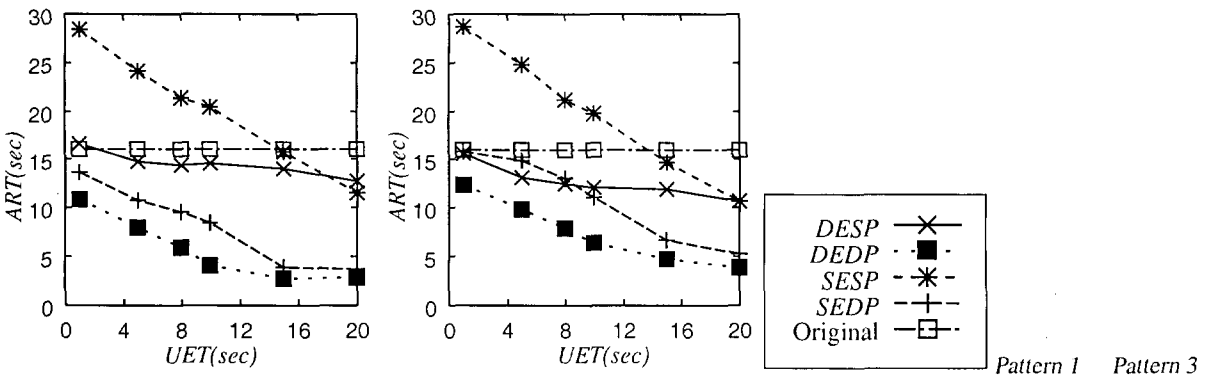


Figure 5: "User-entrusted time (*UET*)" vs. "average response time (*ART*)".

Queries in Multi-transactoin → Biased:
Load → Unbalanced.(The original)
Load → Balanced.(*DEDP*)
Superiority of *DEDP* → Larger

TFR → **Smaller:**
Overlap in time for the processing of each query → Longer
A gain by dynamic load balancing → Larger
Superiority of *DEDP* → Larger

UET → **Larger:**
Overhead of failed speculation → Larger
Superiority of *DEDP* → Smaller

The above results show us that, in many cases, *DEDP* is superior to the original. More detailed analysis of this boundary condition is described in (Sasaki *et al* 1999). The larger *NOO* becomes or the smaller *TFR* becomes, the larger the superiority of *DEDP* becomes than the original. In other words, when a DB becomes much larger or much busier, the effectiveness of *DEDP* becomes much larger.

6 Conclusion and Future Research Directions

We propose a performance improvement method for Thakore's algorithm. The larger or busier a DB becomes, the more superior our *DEDP* approach is to Thakore's algorithm. According to our evaluations, we can achieve about the twice more better performance of the original with not always unrealistic condition; for example, *UET* is 8 seconds, success ratio of speculation is 80%, and only 8 processing nodes. Here, as described in Subsection 4.1, *UET* is the sum of the following three time; a DB user reads some candidate queries, selects his/her query in their mind, and actually inputs it.

Future research directions are: (i)a prediction method to improve the success ratio of speculation, and (ii)a method to reduce penalty on speculation failure.

References

[1] Bestavros, A. & Braoudakis, S. (1995) Value-cognizant Speculative Concurrency Control. *The 21th International Conference on Very Large Data Bases*, p.122-133.

[2] Grandeharizadeh, S. *et al.* (1994) Object Placement in Parallel Object-Oriented Database Systems. *Tenth International Conference on Data Engineering*, p.253-262.

[3] Kim, K.C. (1990) Parallelism in Object-Oriented Query Processing. *Sixth International Conference on Data Engineering*, p.209-217.

[4] McBryan, O.A. (1994) An Overview of Message Passing Environments. *Parallel Computing*, Vol.20, No.9, p.417-444.

[5] Reddy, P. K. & Kitsuregawa, M. (1998) Improving Performance in Distributed Database Systems Using Speculative Transaction Processing. *The Second European Parallel and Distributed Systems Conference*, p.275-285.

[6] Sasaki, T. *et al.* (1999) Performance Improvement of Thakore's Algorithm in Parallel Object-oriented Databases. *International Symposium on Database, Web and Cooperative Systems*, p.123-130.

[7] Thakore, A.K. *et al.* (1995) Algorithms for Asynchronous Parallel Processing of Object-Oriented Databases. *IEEE Transactions on Knowledge and Data Engineering*, Vol.7, No.3, p.487-504.

[8] Yamana, H. *et al.* (1995) A Macrotask-level Unlimited Speculative Execution on Multiprocessors. *ACM International Conference on Supercomputing*, p.328-337.

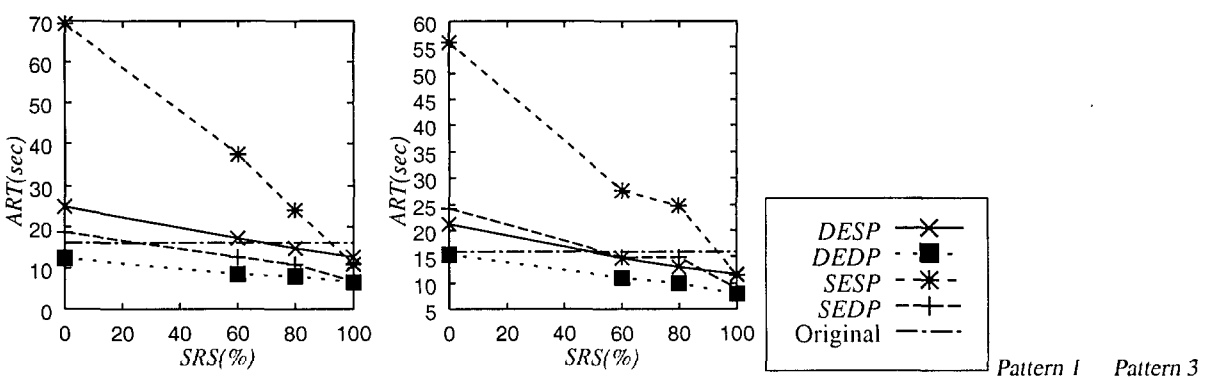


Figure 6: "Success ratio of speculation (*SRS*)" vs. "average response time (*ART*)".

An Ontological Mathematical Framework for Electronic Commerce and Semantically-Structured Web

Vladimir A. Fomichov
Faculty of Applied Mathematics, Moscow State Institute of Electronics
and Mathematics - Technical University, 109028 Moscow, Russia
and Department of Information Technologies
K.E.Tsiolkovsky Russian State Technological University - "MATI"
Orshanskaya 3, 121552 Moscow, Russia
Tel: 007-095-930-9897; Fax: 007-095-939-2090
E-mail: vladfom@yahoo.com , vaf@mech.math.msu.su

Keywords: multi-agent system; electronic commerce; content language; agent communication language; World Wide Web; semantically-structured Web; Resource Description Framework; semantic representation; conceptual formalism; universal conceptual metagrammar; integral formal semantics; restricted K-calculuses; restricted standard K-languages, FIPA Semantic Language

Edited by: Anton P. Železnikar and Yanchun Zhang
Received: August 18, 1999 **Revised:** November 9, 1999 **Accepted:** November 15, 1999

Formalizing conversation of intelligent agents (IAs) which realize Electronic Commerce (EC) includes such aspects as representing knowledge, goals, intentions, and actions of IAs, representing contents of messages (including arbitrary protocols of negotiations) and communicative acts. A common mathematical framework is suggested for all these purposes. This framework is the theory of restricted K-calculuses and K-languages (RKCL-theory), published by the author in Informatica, 1996, No. 1, 1998, No. 4. The suggested mathematical framework possesses all expressive possibilities of the Semantic Language published in 1999 in Geneva by the Foundation for Intelligent Physical Agents (FIPA SL) and of FIPA Agent Communication Language (FIPA ACL). Besides, this framework provides many advantages in comparison with FIPA SL. One of the principal advantages is that a collection of 10 rules is suggested such that one is able to construct the representations of the contents (or structured meanings) of arbitrary natural language discourses and, as a consequence, to construct the representations of arbitrarily complicated goals, actions, and negotiation protocols proceeding from elementary informational items and applying these rules. As a result, the RKCL-theory allows for building compound designations of arbitrary entities considered in the field of EC: sets, finite sequences, relationships, concepts, and structured meanings of texts. That is why the described framework is called an ontological framework.
It is shown that the expressive power of restricted standard K-languages exceeds the expressive power of the Resource Description Framework (RDF) and RDF Schema Specification Language elaborated in 1998-1999 by the WWW-Consortium. It is suggested to use the RKCL-theory as a reference-point and a tool for developing more powerful and flexible conceptual formalisms for the advanced, semantically-structured Web

1 Introduction

The stormy extension of the World Wide Web (WWW), the achievements of the Artificial Intelligence theory, and the progress in the theory and practice of Multi-Agent Systems (MASs) have created the preconditions for the realization of Electronic Commerce (EC). Nowadays, EC is considered as a major application domain for the exploitation of the WWW in the nearest future (FIPA CFP6, 1998).

The Foundation for Intelligent Physical Agents (FIPA), registered in Geneva, suggested in 1998 a special standard for representing various communicative acts (CAs) carried out by computer intelligent agents (IAs) in the course of their interaction. This standard is called the FIPA Agent Communication Language, or the FIPA ACL (FIPA ACL, 1998). Metaphorically speaking, the essence is as follows. 20 different envelopes (corresponding to different CAs) are proposed whose form reflects the purpose of an IA sending a message in this envelope. The purpose may be to make a proposal, to confirm the acceptance of the proposal, to reject a proposal, to raise a question, etc.

One of the subjects of this paper is a possible standard for representing the contents of the messages in such envelopes. At the moment, FIPA has suggested only a preliminary standard for representing the contents of

messages; it is called FIPA Semantic Language, or FIPA SL (FIPA CCL, 1999). The 6th FIPA's Call for Proposals (FIPA CFP, 1999) raises the question of forming a Library of Content Languages and of the criteria for the inclusion of a language into such a library.

Throughout the world, commerce is realized by means of negotiations. Hence the implementation of EC requires the elaboration of such formal means which would be convenient for representing contents of arbitrary negotiation protocols. Obviously, the content of a negotiation may be conveyed by a complicated discourse. Hence we need to elaborate content languages which are convenient for representing the contents (or structured meanings) of complicated real discourses in natural language (NL).

The analysis shows that the expressive power of FIPA SL is insufficient for effective representation of the contents of arbitrary negotiation protocols (see Section 2). That is why there is the necessity of elaborating much more powerful and flexible formal means of representing contents of arbitrary negotiation protocols.

As for the use of the Web in EC , such a use includes, in particular, two activities: (a) the search for suppliers of some products; (b) looking for possible consumers of the manufactured production. Nowadays, HTML is the main language of representing information on the Internet. The structure of HTML documents says nothing about the contents of these documents. The information is semantically unstructured, and conceptual searches for information encounter huge difficulties. That is why the following purpose has emerged: to develop a more semantically-structured, a more knowledge-based Web.

To this end, the World Wide Web Consortium (W3C) elaborated in 1998 - 1999 two new language systems for representing information on the Internet. The aim is to replace HTML and to create a basis for the elaboration of algorithms destined for conceptual searching relevant information proceeding from the requests of the WWW-users.

The first language system is a frame-like system for representing metadata (data about data) and contents of WWW-resources, it is called the Resource Description Framework (RDF). RDF provides the possibility not only to represent metadata (to indicate the names of the creators of a resource, the date of its creation, etc.) but also introduces some special formal structures for representing the meanings of messages. For instance, RDF enables us to reflect in a formal expression that some action was carried out by a set of persons as a single entity (a committee may take a decision by a majority of votes) or that some action was carried out separately by several persons.

RDF provides a restricted spectrum of such possibilities. In particular, RDF gives restricted expressive means for representing alternatives and finite sequences of entities, for constructing object-oriented semantic representations (SRs) of statements.

The second language system elaborated in 1998 - 1999 by W3C is called RDF Schema Specification Language (RDF SSL). This second system allows for determining classes and subclasses, indicating domains and ranges of functions, and describing attributes of relationships.

It appears that the creation of RDF and RDF SSL is a sign of a new phase in the evolution of the WWW. The distinctive feature of this phase is the construction of more subtle expressive means for representing the contents of natural language sentences and discourses.

The main purpose of this paper being an extended version of (Fomichov, 1999a) and continuing the line of (Fomichov, 1998, 1999b) is to draw the attention of researchers throughout the world to the unique opportunities provided by the theory of restricted K-calculuses and K-languages, or the RKCL-theory (Fomichov, 1996, 1998) for representing the contents of arbitrary protocols of negotiations, elaborating logics of the functioning of intelligent agents realizing EC, and for developing more powerful and flexible conceptual formalisms for the advanced Web. The RKCL-theory is a central part of an original theory of formalizing semantics and pragmatics of NL called Integral Formal Semantics (Fomichov, 1994, 1996).

2 The Concept of an Ontological Mathematical Framework for Electronic Commerce

Let's pay attention to the fact that negotiation protocols may include: (a) questions with interrogative words;
(b) questions with the answer "Yes" or No";
c) infinitives with dependent words ("to sell 50 boxes with oranges");
(d) constructions formed out of infinitives with dependent words by means of the logical connectives "and", "or", "not";
(e) complicated designations of sets (" a consignment consisting of 50 boxes with oranges"); (f) fragments where the logical connectives "and", "or" join not the designations of assertions but the designations of objects ("the product A is distributed by the firms B1, B2, ..., BN");
(g) explanations of the terms being unknown to an IA (because the firms invent and produce new products);
(h) fragments containing the references to the meanings of phrases or larger fragments of a discourse ("this proposal", "that order", etc.);
(i) the designations of the functions whose arguments and/or values may be the sets of objects ("the staff of the

firm A", "the suppliers of the firm A", "the number of the suppliers of the firm A").

The analysis shows that the expressive power of FIPA SL is insufficient for effective representing contents of arbitrary negotiation protocols. In particular, from the standpoint of describing semantic structure of the fragments of the types (d) - (h) and taking into account the existence of the designations of the type (i). That is why there is the necessity of elaborating much more powerful and flexible formal means allowing us to represent the contents of arbitrary negotiation protocols.

It is clear that the progress in realizing EC essentially depends on the success of transferring the experience and intuition of human experts in traditional commerce into a machine-understandable form. That is why it may be conjectured that we need a special *formal tool for constructing a bridge* between, on one hand, the experience and intuition of human experts in the field of commerce and, on the other side, the FIPA SL and FIPA ACL.

It is to be a definition of such a class of formal languages that the expressions of these languages are convenient for designating various entities mentioned in the protocols of traditional negotiations. These entities may be sets, sequences, compound concepts, relationships, and structured meanings of sentences and of larger parts of texts. Besides, such languages are to be convenient for representing *knowledge pieces* and *the goals* and *actions* of intelligent systems.

Such a formal tool may be called *an Ontological Mathematical Framework (OMF)*. Using such a framework, it would be much easier to transfer the experience of human experts - in EC and in other application fields - to artificial intelligent agents and to develop and to verify the logics of the functioning of these intelligent agents.

The main purpose of the following sections is to state serious arguments in favour of the hypothesis that the RKCL-theory may be interpreted as an OMF for Electronic Commerce.

3 Briefly about the inference rules of restricted K-calculuses

The RKCL-theory makes the following discovery both in non-mathematical and mathematical linguistics: a system of 10 operations on structured meanings (SMs) of NL-texts is found such that, using primitive conceptual items as "blocks", we are able to build SMs of arbitrary NL-texts (including articles, textbooks, etc.) and arbitrary pieces of knowledge about the world. Such operations will be called *quasilinguistic conceptual operations*. Hence the RKCL-theory suggests a *complete collection* of quasilinguistic conceptual operations (it is a hypothesis supported by many weighty arguments). It may be noted that the RKCL-theory effectively takes into account the existence of the NL phenomena (a) - (i) indicated above.

A complete system of mathematical definitions stated in Fomichov (1996, 1998) describes a collection consisting of 10 quasilinguistic conceptual operations. This system of definitions determines also a class of formal languages called restricted standard K-languages (RSK-languages). The formal side is stated very briefly below; a more detailed outline may be found in (Fomichov, 1998).

At the first step (consisting of a rather long sequence of auxiliary steps), the RKCL-theory defines a class of formal objects called *simplified conceptual bases (s.c.b.)*. Each s.c.b. B is a system of the form $((c_1, c_2, c_3), (c_4, \ldots, c_7), (c_8, \ldots, c_{14}))$ with components $c_1, \ldots, c_{14}$ being mainly finite or countable sets of symbols and distinguished elements of such sets. In particular, $c_1 = St$ is a finite set of symbols called sorts and designating the most general considered concepts; $c_2 = P$ is a distinguished sort "sense of proposition"; $c_4 = X$ is a countable set of strings used as elementary blocks for building knowledge modules and semantic representations (SRs) of texts; X is called a primary informational universe; $c_5 = V$ is a countable set of variables; $c_7 = F$ is a subset of X whose elements are called functional symbols.

Each s.c.b. B determines three classes of formulas, where the first class $Lrs(B)$ is considered as the principal one and is called *the restricted standard K-language in the s.c.b.* B. Its strings (they are called K-strings, or l-formulas) are convenient for building semantic representations (SRs) of NL-texts. We will consider below only the formulas from the first class $Lrs(B)$.

In order to determine for an arbitrary s.c.b. B three classes of formulas, a group of inference rules P[0], P[1],..., P[10] is defined. The ordered pair $Krs(B) = (B, Rls)$, where Rls is the set consisting of all these rules, is called *the restricted K-calculus in the s.c.b.* B. The rule P[0] provides an initial stock of formulas from the first and second classes. E.g., there is such an s.c.b. $B1$ that, according to the rule P[0], $Lrs(B1)$includes the elements $box1$, $green$, $city$, set, $India$, 7, all, any, $Weight$, $Distance$, $Staff$, $Suppliers$, $Quantity$, $x1$, $x2$, $P1$, $P2$, $Manufactured-in$, $delivery$, $Addressee$, "$Spencer\ \&\ Co.$", $Propose$, $Want$.

Lets regard (ignoring many details) the structure of strings which can be obtained by applying any of the rules P[1],..., P[10] at the last step of inferencing these formulas. The rule P[1] enables us to build l-formulas of the form $Quant\ Conc$ where $Quant$ is a semantic item corresponding to the meanings of such words and

expressions as "some", "any", "arbitrary", "each", "all", "several", "many", etc. (such semantic items will be called *intensional quantifiers*), and $Conc$ is a designation (simple or compound) of a concept. Examples of K-strings for P[1] as the last applied rule are as follows: :: $box1$, $all\ box1$, :: $consignment$, :: $box1 * (Content2, ceramics)$, where the last expression is built with the help of both the rules P[0], P[1] and some other rule (with the number 4), and the symbol '::' is to be interpreted as the informational item corresponding to the word "some" in cases when this word is associated with singular.

The rule P[2] allows for constructing the strings of the form $f(a_1, \ldots, a_n)$, where f is a designation of a function, $n \geq 1$, $a_1, \ldots, a_n$ are l-formulas built with the help of any rules from the list P[0],..., P[10]. The examples of l-formulas built with the help of P[2]:

$$Distance(Moscow, Tokyo),\ Weight(::\ box1 *$$

$$(Colour, green)(Content2, ceramics))$$

Using the rule P[3], we can build the strings of the form $(a_1 \equiv a_2)$, where a_1 and a_2 are l-formulas formed with the help of any rules from P[0],..., P[10], and a_1 and a_2 represent the entities being homogeneous in some sense. The examples of K-strings for P[3]:

$$(Distance(Moscow, Tokyo) \equiv x1),\ (y1 \equiv y3),$$

$$(Weight(::\ box1) \equiv \langle 8, kg \rangle)$$

The rule P[4] is destined, in particular, for constructing K-strings of the form $rel(a_1, \ldots, a_n)$, where rel is a designation of n-ary relation, $n \geq 1$, $a_1, \ldots, a_n$ are the K-strings formed with the aid of some rules from P[0],..., P[10]. The examples of K-strings for P[4]:

$$Belong(Osaka, Cities(Japan)),$$

$$Subset(Cities(Belgium), Cities(Europe)).$$

The rule P[5] enables us to construct the K-strings of the form $Expr\ :v$, where $Expr$ is a K-string not including v, v is a variable, and some other conditions are satisfied. Using P[5], one can mark (by means of variables) in the SR of any NL-text: (a) the descriptions of diverse entities mentioned in the text (physical objects, events, concepts, etc.), (b) the SRs of sentences and of larger texts' fragments to which a reference is given in any part of a text. Examples of K-strings for P[5]:

$$::\ box1\ :\ x3\ ,$$

$$Higher(::\ box1\ :\ x3, ::\ box1\ :\ x5)\ :\ P1\ .$$

The rule P[5] provides the possibility to form SRs of texts in such a manner that these SRs reflect the referential structure of NL-texts; the examples are considered below.

The rule P[6] provides the possibility to build the K-strings of the form $\neg Expr$, where $Expr$ is a K-string satisfying a number of conditions. The examples of K-strings for P[6]:

$$\neg ship,\ \neg Belong(Osaka, Cities(Belgium)).$$

Using the rule P[7], one can build the K-strings of the forms $(a_1 \wedge \wedge a_2 \wedge \ldots \wedge a_n)$ or $(a_1 \vee a_2 \vee \ldots \vee a_n)$, where $n \geq 1$, $a_1, \ldots, a_n$ are K-strings designating the entities which are homogeneous in some sense. In particular, $a_1, \ldots, a_n$ may be SRs of assertions (or propositions), descriptions of physical things, descriptions of sets consisting of things of the same kind, descriptions of concepts. The following strings are examples of K-strings (or l-formulas) for P[7]:

$$(Finnland \vee Norway \vee Sweden),$$

$$(Belong((Namur \wedge Leuven \wedge Ghent),$$

$$Cities(Belgium)) \wedge \neg Belong(Bonn,$$

$$Cities((Finnland \vee Norway \vee Sweden)))).$$

The rule P[8] allows us to build, in particular, K-strings of the form

$$c * (rel_1, val_1), \ldots, (rel_n, val_n)\ ,$$

where c is an informational item from the primary universe X designating a concept, for $i = 1, \ldots, n$, rel_i is a function with one argument or a binary relation, val_i designates a possible value of rel_i for objects characterized by the concept c. The following expressions are examples of K-strings for P[8]:

$$box1 * (Content2, ceramics),$$

$$firm1 * (Fields, chemistry),$$

$$consignment * (Quantity, 12)(Compos1, box1 *$$

$(Content2, ceramics))$.

The rule P[9] permits to build, in particular, the K-strings of the forms $\forall v(conc)D$ and $\exists v(conc)D$, where $\forall$ is the universal quantifier, $\exists$ is the existential quantifier, *conc* and D are K-strings, *conc* is a designation of a prime concept ("person", "city", "integer", etc.) or of a compound concept ("integer greater than 200", etc.). D may be interpreted as a SR of an assertion with the variable v about any entity qualified by the concept *conc*. The examples of K-strings for P[9] are as follows:

$$\forall n1(nat)\exists n2(nat)\ Less(n1, n2)\ ,$$

$$\exists y(country * (Location, Europe))$$

$$Greater(Quantity(Cities(y)), 15)\ .$$

The rule P[10] is destined for constructing, in particular, the K-strings of the form $\langle a_1, \ldots, a_n\rangle$, where $n > 1$, $a_1, \ldots, a_n$ are K-strings. The strings obtained with the help of P[10] at the last step of inference are interpreted as designations of n-tuples. The components of such n-tuples may be not only designations of numbers, things, but also SRs of assertions, designations of sets, concepts, etc. Using jointly P[10] and P[4], we can build the K-strings $\langle 8, kg\rangle$,

$$Work1(\langle Agent1, :: man*$$

$$(F.name,' Ulrich')(Name,' Stein'))$$

$$\langle Organization, Siemens\rangle, \langle Start - time, 1996\rangle),$$

where the thematic roles $Agent1$, $Organization$, $Start - time$ are explicitly represented.

4 Some Possibilities of Representing Contents of Messages and Communicative Acts by Means of Restricted Standard K-languages

Consider some properties which make restricted standard K-languages (RSK-languages) convenient for representing contents of messages and describing communicative acts.

Property 1 It is possible to build formal representations of compound concepts. E.g., the K-strings

$$ceramics * (Manufactured - in,$$

$$(India \vee Sri - Lanka)),$$

$$Container1 * (Content1, (:: set * (Quantity,$$

$$8)(Compos1, box1 * (Content2,$$

$$Service1 * (Kind, tea)(Country, China)))\wedge$$

$$:: set * (Quantity, 4)(Compos1,$$

$$Box1 * (Content2, service1 * (Kind,$$

$$dinner)(Country, (India \vee Sri - Lanka))))))$$

may designate, respectively, the concepts "ceramics manufactured in India or Sri Lanka" and "a container containing 8 boxes with tea services from China and 4 boxes with dinner services from India or Sri Lanka", where the symbol '::' is to be interpreted as the informational item corresponding to the word "some" (associated with singular).

Property 2 RSK-languages provide large possibilities for representing knowledge items being definitions of concepts. Suppose that E is a NL-expression, $Semp$ is a string of some RSK-language and $Semp$ is a possible semantic representation (SR) of E. Then we will say that $Semp$ is a RK-representation (RKR) of E. If T1 = "Freight forward is a freight to be paid in the port of destination" then T1 may have a RKR of the form

$$(freight - forward \equiv freight * (Description,$$

$$\langle x, Payment - at(x,$$

$$:: port1 * (Destination - of, x))))).$$

Property 3 RSK-languages allow us to build compound designations of various entities, including designations of sets.

Example 1. The sea port Murmansk in the north-west of Russia can be designated by the K-string

$$:: port1 * (Title, "Murmansk")$$

or by the K-string

$$:: port1 * (Title, "Murmansk")\ :\ x28\ ,$$

where the string $x28$ is to be interpreted as a variable marking just this particular entity.

Example 2. We can designate a concrete planned series of 5 consignments, each consisting of 60 tea services No. 53 and 36 dinner services No. 65, as follows:

$$:: set * (Quantity, 5)(Compos1, consignment*$$

$$(Compos2, (:: set * (Quantity, 60)$$

$$(Compos1, service1 * (Kind, tea)(No, 53))$$

$$\wedge :: set * (Quantity, 36)(Compos1,$$

$$service1 * (Kind, dinner)(No, 65))))) \ : \ S1.$$

Property 4 RSK-languages enable us to build formal representations of simple and complicated goals. E.g., the goal "To deliver to the firm "Spencer & Co." during 12 - 19 November 1999 five consignments, each consisting of 60 tea services No. 53 and 36 dinner services No. 65" may have the following RKR:

$$delivery * (Addressee, :: firm1 * (Title,$$

$$"Spencer \ \& \ Co.") \ : \ x1)(Time, \langle(12, 11, 99\rangle,$$

$$\langle 19, 11, 99\rangle\rangle)(Object1, setdescr1),$$

where $setdescr1$ is the K-string constructed above in Example 2 (Property 3). One can form complicated goals with the help of logical connectives "and", "or", "not". E.g., if $g1, g2, g3, g4$ represent simple goals like the K-string above, then it would be possible to build the representations of compound goals $((g1 \wedge g2) \vee (g3 \wedge \neg g4))$, $(g1 \wedge g2 \wedge g3)$, etc.

Property 5 It is the possibility and the convenience of describing structured meanings (SMs) of arbitrary assertions. This possibility is substantially grounded in (Fomichov, 1994, 1996, 1998). Besides, it was illustrated above. That is why consider only one example (pertaining to commerce) demonstrating the possibility of representing SMs of discourses with references to the meanings of the fragments being phrases or larger parts of the discourse.

Let T2 = "As we propose to run a series of 12 consecutive advertisements, we should like to know what discount you can allow for this". Suppose that the pronouns "we" and "you" are associated in a concrete situation of communication with the firms "Spencer & Co." and "Smith and Brown", respectively. Then a possible RKR of T2 is as follows:

$$(Propose(\langle Agent1, :: firm1 * (Title, "Spencer \&$$

$$Co.") \ : \ x1\rangle\langle Addressee, :: firm1 * (Title,$$

$$"Smith \ \& \ Brown") \ : \ x2\rangle\langle Action1, running2 *$$

$$(Object1, :: set * (Quantity, 12)$$

$$(Kind - of - set, consecutive - elements)(Compos1,$$

$$advertisement)) \ : \ x3\rangle\langle Moment, t1\rangle \ : \ P1$$

$$\wedge Want(\langle Agent1, x1\rangle, \langle Moment, t1\rangle,$$

$$\langle Action1, knowing * (Content3,$$

$$:: discount * (Provided, x2)(Reason, P1))))).$$

In this formula, the variable $P1$ marks the meaning of the sentence being the first part of T2.

Property 6 RSK-languages allow us to build formulas almost coinciding with the summary definitions of standard communicative acts given in (FIPA ACL, 1998).

Example 1. Semantics of the communicative act "accept-proposal" may be represented by the RSK-expression

$$(\langle i, accept - proposal(j, \langle j, a\rangle,$$

$$p(e, \langle j, a\rangle)) \equiv \langle i, inform(j, I(i),$$

$$Will - occur - when(\langle j, a\rangle, p(e, \langle j, a\rangle))))$$

Example 2. Semantics of the communicative act "inform-if" may be represented by the formula

$$(\langle i, inform - if(j, p)\rangle \equiv (\langle i, inform(j, p) >$$

$$\vee\langle i, inform(j, \neg p))))$$

The same approach may be used for approximating all summary definitions of standard communicative acts considered in (FIPA ACL, 1998).

Property 7 The RKCL-theory provides the possibility of representing communicative acts (CAs). Suppose that agent Client-agent asks Ontology-agent for the reference of instances of a class citrus (FIPA OS, 1998). This CA may be reflected by the following expression of a RSK-language:

$$:: com - act * (Kind, query - if)(Sender, client -$$

$$agent)(Receiver, ontology - agent)(Content,$$

$$Question(x1, (x1 \equiv :: set * (Compos1,$$

$$any \ concept * (Instance, citrus)))))$$

$$(Language,\ sl)(Ontology,$$

$$\langle fipa - ontol - service - ontology,$$

$$fruits - ontology))$$

$$(Reply - with, citrus - query)$$

So the structure of constructed strings may be very close to the structures used in the FIPA ACL for representing CAs. Using RSK-languages, it is easy to represent in the similar ways all communicative acts considered in (FIPA ACL, 1998).

5 Significance of Obtained Results for Developing a Semantically-Structured Web

One of the two principal preconditions of realizing Electronic Commerce (EC) has been the stormy development of the Web. E.g., it was mentioned above that the processes of EC include the search for the suppliers of some products and looking for possible consumers of the manufactured production. Hence an OMF for EC is to harmonize with the advanced means of representing information in Web suggested recently by the WWW-Consortium. Lets see that the RKCL-theory satisfies this criterion.

5.1 Approximation of the Expressive Means Provided by RDF

Example 1. According to (RDF, 1999), the sentence T1 = "The students in the course 6.001 are Amy, Tim, John, Mary, and Sue" is translated (in some pragmatic context) into the RDF structure

$$\langle rdf : RDF\rangle\langle rdf : Description$$

$$about = "U1/courses/6.001"\rangle$$

$$\langle s : students\rangle\langle rdf : Bag\rangle$$

$$\langle rdf : liresource = "U1/stud/Amy"/\rangle$$

$$\langle rdf : liresource = "U1/stud/Tim"/\rangle$$

$$\langle rdf : liresource = "U1/stud/John"/\rangle$$

$$\langle rdf : liresource = "U1/stud/Mary"/\rangle$$

$$\langle rdf : liresource = "U1/stud/Sue"/\rangle$$

$$\langle/rdf : Bag\rangle\langle/s : students\rangle$$

$$\langle/rdf : Description\rangle\langle/rdf : RDF\rangle$$

, where $U1$ is an URL.

In this expression, the item Bag is the indicator of a bag container object. It is possible to construct the following similar expression of some RSK-language:

$$:: course1 * (W3ad, "U1/courses/6.001")$$

$$(Students, :: bag * (Compos2, (:: stud * (W3ad,$$

$$"U1/stud/Amy") \wedge :: stud * (W3ad,$$

$$"U1/stud/Tim") \wedge :: stud * (W3ad,$$

$$"U1/stud/John") \wedge :: stud * (W3ad,$$

$$"U1/stud/Mary") \wedge :: stud * (W3ad,$$

$$"U1/stud/Sue")))).$$

Here the symbol '::' is interpreted as the referential quantifier, i.e. as the informational item corresponding to the word "some" in cases when it is used for building the word combinations in singular ("some job", "some personal computer", etc.).

Example 2. Following (RDF, 1999), the model for the sentence T2 = "The source code for X11 may be found at U3, U4, or U5" (where U3, U4, U5 are some URLs) may be written in RDF (with respect to some pragmatic context) as:

$$\langle rdf : RDF\rangle\langle rdf : Description$$

$$about = "U2/packages/X11"\rangle$$

$$\langle s : DistributionSite\rangle\langle rdf : Alt\rangle$$

$$\langle rdf : liresource = "U3"/\rangle$$

$$\langle rdf : liresource = "U4"/\rangle$$

$$\langle rdf : liresource = "U5"/\rangle$$

$$\langle /rdf : Al\rangle\langle /s : DistributionSite\rangle$$

$$\langle /rdf : Description\rangle\langle /rdf : RDF\rangle$$

Here the informational item Alt is the indicator of an alternative container object. The RKCL-theory suggests the following similar expression:

$$:: resource * (W3ad, "U2/packages/X11")$$

$$(DistributionSite, (:: resource * (W3ad, "U3")$$

$$\vee :: resource * (W3ad, "U4")$$

$$\vee :: resource * (W3ad, "U5"))).$$

Example 3. Consider the sentence T3 = "Ora Lassila is the creator of the resource U6 and the corresponding RDF-structure

$$\langle rdf : RDF\rangle\langle rdf : Descriptionabout = "U6"\rangle$$

$$\angle s : Creator = "OraLassila"/\rangle\langle /rdf : RDF).$$

Using some RSK-language, we can build the following description of the mentioned resource:

$$:: resource * (W3ad, "U6")(Creator,$$

$$"OraLassila")$$

Example 4. The RKCL-theory enables us also to build reified conceptual representations of statements, i.e. the representations in the form of named objects having some external ties: with the set of the authors, the date, etc. For instance, we can associate the sentence T3 = "Ora Lassila is the creator of the resource U6" with the expression of some RSK-language

$$:: info - piece * (RDF - type, Statement)$$

$$(Predicate, Creator)$$

$$(Subject, "U6")(Object, Ora_Lassila) : i1024$$

where $i1024$ is the name of an information piece. This form is very close to the RDF-expression (RDF, 1999)

$$\{type, [X], [RDF : statement]\}$$

$$\{predicate, [X], Creator]\}$$

$$\{subject, [X], [U6]\}\{object, [X], "OraLassila"\}$$

Proceeding from the ideas considered in the Examples 1 - 4 and in Sections 3 and 4, we would be able to approximate all RDF-structures by the similar expressions of RSK-languages.

5.2 Approximation of the Expressive Means Provided by RDF Schema Specification Language

Example 1. The RDF SSL description of the class "Marital status" from (RDF SSL,1999)

$$\langle rdfs : Classrdf : ID = "MarStatus"/\rangle\langle MarStatus$$

$$rdf : ID = "Married"/\rangle$$

$$\langle MarStatusrdf : ID = "Divorced"/\rangle$$

$$\langle MarStatusrdf : ID = "Single"/\rangle$$

$$\langle /rdfs : Class\rangle$$

can be represented by the following K-string:

$$(any\ \#MarStatus \equiv$$

$$(Married \vee Divorced \vee Single))$$

Example 2. The RDF SSL definition of the property "marital status" from (RDF SSL, 1999)

$$\langle rdf : PropertyID = "maritalStatus"\rangle$$

$$\langle rdfs : rangerdf : resource = "\#MarStatus"/\rangle$$

$$\langle rdfs : domainrdf :$$

$$resource = "\#Person"/\rangle\langle /rdf : Property\rangle$$

can be approximated by the K-string

$$(maritalStatus\#1(any\ \#Person) \equiv$$

$$any\ \#MarStatus)$$

where the substring #1 means that we consider a name of a function with one argument.

5.3 Theory of K-calculuses as a Tool for Elaborating New Conceptual Formalisms for the World Wide Web

It appears that RDF and RDF SSL are only the first steps of the Web Consortium along the way of developing semantically-structured (or conceptual) formalisms, and hence the next steps will be made in the future. Lets try to imagine what may be the result of the evolution of consequent Web conceptual formalisms, for instance, 7 - 10 years later.

In order to formulate a reasonable assumption, let's consider such important applications of Web as Digital Libraries and Multi Media Databases. If the resources are articles, books, pictures or films, then, obviously, important metadata of such resources are semantic representations of summaries (for textual resources and films) and high-level conceptual descriptions of pictures. As for EC, conceptual representations of the summaries of business documents are important metadata of resources. That is why it seems that the Web conceptual formalisms will evolve in 7 - 10 years to a Widely-Applicable Conceptual Metagrammar.

Hence the following fundamental problem emerges: how to construct a Universal Conceptual Metagrammar (UCM) enabling us to build semantic representations (in other words, conceptual representations) of arbitrary sentences and discourses in NL? Having such a UCM, we will be able, obviously, to build high-level conceptual descriptions of visual images too.

An answer to this question is given in (Fomichov, 1996): the hypothesis is put forward that the RKCL-theory may be interpreted as a possible variant of a UCM. With respect to this hypothesis and the fact that the RKCL-theory enables us to effectively approximate the expressive means of RDF and RDF SLL, we may suppose that the RKCL-theory can be used as an effective tool and as a reference-point for developing comparable, more and more powerful and flexible conceptual formalisms for the advanced Web.

The analysis (in particular, that carried out above) shows that the RKCL-theory is a convenient tool for a constructing formal representations of the contents of arbitrary messages sent by intelligent agents, for describing communicative acts and metadata about the resources, and for building high-level conceptual representations of pictures. That is why the RKCL-theory together with the recommendations concerning its application in the enumerated directions may be called a *Universal Resources and Agents Framework (URAF)*; this term was introduced for the first time in (Fomichov, 1999b).

6 Related approaches

The analysis shows that everything that may be expressed by means of Knowledge Interchange Format, or KIF (Genesereth, 1999), FIPA Semantic Language (SL), FIPA ACL (FIPA ACL, 1998), may be expressed also by means of RSK-languages.

The particular advantages of the RKCL-theory in comparison with FIPA SL (FIPA CCL, 1999), Discourse Representation Theory (van Eijck and Kamp, 1996), and Episodic Logic (Hwang and Schubert, 1993) are, in particular, the possibilities: (1) to distinguish in a formal way objects (physical things, events, etc.) and concepts qualifying these objects; (2) to build compound representations of concepts; (3) to distinguish in a formal manner objects and sets of objects, concepts and sets of concepts; (4) to build complicated representations of sets, sets of sets, etc.; (5) to describe set-theoretical relationships; (6) to describe effectively structured meanings (SMs) of discourses with references to the meanings of phrases and larger parts of discourses; (7) to describe SMs of sentences with the words "concept", "notion"; (8) to describe SMs of sentences where the logical connective "and" or "or" joins not the expressions-assertions but designations of things or sets or concepts or goals; (9) to consider non-traditional functions with arguments or/and values being sets of objects, of concepts, of texts semantic representations, etc.; (10) to construct formal analogues of the meanings of arbitrary constructions built out of infinitives with dependent words.

The RKCL-theory provides the possibility to approximate all RDF-structures by the similar expressions of RSK-languages. The same applies to the RDF Schema Specification Language (RDF SSL, 1999). The analysis of RDF expressive means supports such basic ideas of the RKCL-theory as: building compound formal designations of sets; joining by logical connectives not only the designations of assertions but also the designations of things, events, and concepts; considering assertions as objects having some external ties: with a date, the set of the authors, a language, etc.

The principal advantage of the RKCL-theory in comparison with all approaches mentioned above is that it indicates a small collection of operations enabling us to build semantic representations of arbitrary NL-texts and, as a consequence, to express in a formal way arbitrarily complicated goals and plans of actions, to represent contents of arbitrary protocols of negotiations.

7 Conclusions

A new theoretical framework for Electronic Commerce and for developing a more semantically-structured Web is introduced. It is called the Universal Resources and Agents Framework (URAF). The mathematical component of the new framework is the RKCL-theory. The suggested framework provides:

(a) a common syntax for representing communicative acts, ontologies of application domains, contents of messages, metadata (data about data);
(b) a formalism for representing contents (or structured meanings) of arbitrarily complicated real sentences and discourses pertaining to commerce, law, medicine, technology, etc.;
(c) a universal and flexible formalism for representing knowledge about the world and, hence, for building ontologies of application domains;
(d) the possibility of representing communicative acts carried out by intelligent agents in the same way as one does while building semantic representations (SRs) of texts mentioning such acts with the aid of the verbs " inform", "say", "propose ", "confirm", etc.;
(e) a much richer formalism for representing metadata and contents of messages than the Resource Description Framework (RDF) suggested in 1998-1999 by the World Wide Web Consortium;
(g) such means of representing communicative acts which are not less flexible than the means provided by the ACL suggested by the Foundation for Intelligent Physical Agents (FIPA, Geneva) and which can approximate the means considered in FIPA 98 Specification - Part 2;
(h) the opportunities to describe formally semantics of ACL which are very close to the opportunities used for describing semantics of communicative acts in the FIPA 98 Specification - Part 2.

The properties (a) - (e) are the main distinctive features and advantages of the suggested framework in comparison with the framework given by the FIPA 98 Specification - Part 2.
The property (b) provides the possibility to suggest a standard for the design of Content Languages. This property means, in particular, that URAF may be used for representing the contents of arbitrary documents needed for the realization of Electronic Commerce (EC), including the protocols of negotiations.

Due to the properties (a) - (f), the new framework bridges a considerable gap between the multi-agent systems community and the community of the designers of the World Wide Web. Besides, URAF provides a powerful theoretical background for developing the WWW in the direction of perfecting means of the conceptual search of various information pieces, proceeding from the requests of the end users.

EC is considered nowadays as a mayor application domain for the exploitation of Internet in the nearest future. Besides, EC is ideally suited for multi-agent and intelligent agent technologies. That is why, due to the properties (a)-(g), the suggested mathematical framework opens new prospects both for EC and for other applications of multi-agent systems. Lets underline now only two ideas. Firstly, it would be considerably easier to elaborate and to verify the logics of functioning of intelligent agents in the situation when the structure of knowledge representation in computer is very close to the structure of NL-texts expressing the experience of human experts in commerce. Secondly, it will be possible to use in the nearest future the NL-processing programs for transferring texts reflecting human experience and intuition into computer-understandable forms.

Acknowledgements

I am grateful to the anonymous referees for the remarks helped to improve the presentation style of this paper.

References

[1] Eijck, D.J.N. van and H. Kamp (1996): Representing Discourse in Context; Amsterdam, The University of Amsterdam.

[2] FIPA ACL (1998): The Foundation for Intelligent Physical Agents. FIPA'98 Specification. Part 2 - Agent Communication Language. The text refers to the specification dated October 1998. Geneva, 1998, on-line at http://www.fipa.org/spec/FIPA98.html.

[3] FIPA OS (1998): FIPA'98 Specification, Part 12, Ontology Service, on-line at http://www.fipa.org/spec/FIPA98.html.

[4] FIPA CLL(1999): FIPA Spec 18-1999. DRAFT, Version 0.1. FIPA Content Language Library. Geneva, 1999, on-line at http://www.fipa.org/spec/fipa99spec.htm

[5] FIPA CFP6 (1999): FIPA Sixth Call for Proposals, Geneva.

[6] Fomichov, V.A. (1994): Integral Formal Semantics and the Design of Legal Full-Text Databases. Cybernetica (Belgium), Vol. XXXVII, No. 2, 145-177.

[7] Fomichov, V.A. (1996): A Mathematical Model for Describing Structured Items of Conceptual Level. Informatica (Slovenia), Vol. 20, No. 1, 5-32.

[8] Fomichov, V.A. (1998): Theory of Restricted K-calcules as a Comprehensive Framework for Constructing Agent Communication Languages. Special Issue on NLP and Multi-Agent Systems, edited by V.A.Fomichov and A.P.Zeleznikar, Informatica. An International Journal of Computing and Informatics (Slovenia), Vol. 22, No. 4, 451-463.

[9] Fomichov, V.A. (1999a): Theory of Restricted K-calculuses as a Universal Informational Framework

for Electronic Commerce. Database, Web and Cooperative Systems. Vol. 1. The Proceedings of 1999 International Symposium on Database, Web and Cooperative Systems, August 3-4, 1999 in Baden-Baden, Germany - DWACOS'99. Edited by George E. Lasker, University of Windsor and Yanchun Zhang, University of Southern Queensland. The International Institute for Advanced Studies in Systems Research and Cybernetics, University of Windsor, Windsor, Ontario, Canada, 41-46.

[10] Fomichov, V.A. (1999b): A Universal Resources and Agents Framework for Electronic Commerce and Other Applications of Multi-Agent Systems. Peter Kopacek (ed.), Preprints of the 7th International Workshop on Computer Aided Systems Theory and Technology 1999 - EUROCAST'99. September 29th - October 2nd, 1999, Vienna, Austria, Vienna University of Technology, 99-102.

[11] Genesereth, M.R. (1999). Knowledge Interchange Format. Geneva, FIPA, 1999; on-line at http://www.fipa.org.

[12] Hwang, C.H. and L.K. Schubert (1993): Episodic Logic: a Comprehensive, Natural Representation for Language Understanding; Minds and Machines, 3, 381-419.

[13] RDF (1999). Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation. January 1999, on-line at http://www.w3.org/TR/WD-rdf-syntax.

[14] RDF SSL (1999). Resource Description Framework (RDF) Schema Specification. W3C Recommendation, March 1999, on-line at http://www.w3.org/TR/WD-rdf-schema.

A Technique of Watermarking for Digital Images Using (t,n)-Threshold Scheme

Chin-Chen Chang and Pei-Fang Chung
Department of Computer Science and Information Engineering
National Chung Cheng University, Chaiyi, Taiwan 621, R.O.C.
e-mail:ccc,pfc87@cs.ccu.edu.tw
AND
Tung-Shu Chen
Department of Computer Science and Information Management
Providence University, Taichung, Taiwan 433, R.O.C.
e-mail:tchen@pu.edu.tw

Keywords: (t, n)-threshold scheme, Torus automorphism, digital watermarking

Edited by: Yanchun Zhang, Vladimir Fomichov, and Anton P. Železnikar
Received: August 15, 1999 **Revised:** November 1, 1999 **Accepted:** November 8, 1999

In this paper, a new watermarking method is proposed for copyright protection of two-color bitmaps. It is called the threshold watermarking method (TW). TW combines a watermark and a two-color bitmap together. This combination is basically achieved following the concept of the (t, n)-threshold scheme. TW extracts n characteristic values from the bitmap first and thus generates some information to protect the copyright of this bitmap. TW guarantees that it can fight against modificative attacks if there are more than t characteristic values existing in the modified bitmap. This phenomenon has been shown in our experiments. Hence TW is a robust watermarking method.

1 Introduction

Nowadays, computers have become a part of the basic electronic equipment of our daily life. Following this is a gradual growth of computer networks. As the population of network users increases rapidly, more and more makers put their digital products on networks which are convenient for their customers to access. But the makers of these products cannot expect their customers all to be law-abiding ones. Some of them might reproduce these digital products for illegal use. Therefore, enforcing copyright protection of these products is important and urgent. Watermarking is a way to protect copyright [1]. With watermarking, makers can hide some related publishing data of the copyright property into the products to make a claim of their ownership.

Associated literature [2][3] shows evidence of much research into watermarking methods. However, until now, there is still no watermarking method proposed to protect the copyright of two-color bitmaps. This is because, if a watermark is embedded into a two-color bitmap by changing the values of some of the pixels of the bitmap, the watermark will not be perceptually undetectable, and people can detect the protection directly with their eyes. These kinds of watermarking methods, embedding the watermark by altering the original image's pixel values, cannot apply to a two-color bitmap. Hence a new watermarking method is proposed in this paper to deal with this problem. It is called the *threshold watermarking method* (TW) for protecting the copyright of bitmaps. This new method is derived from the conception of the (t, n)-threshold scheme and the theory of Torus automorphism [5]. TW incorporates Torus automorphism to scramble the contents of the original bitmap first. Next, TW extracts n characteristic values from the scrambled bitmap and generates some information to protect the copyright of this bitmap. TW is robust since, for a modified bitmap, if more than t characteristic values exists in it, then TW can restore the watermark following the (t, n)-threshold scheme. Hence TW is capable of working against modificative attacks.

2 Previous works

2.1 (t, n)-Threshold scheme

The (t, n)-threshold scheme is usually implemented by the interpolation polynomial method proposed by Shamir [4]. This method can be portrayed as follows. Assume a dealer has a secret K and a prime number p. The value of p is greater than or equal to the value of K. According to the interpolation polynomial method, the dealer constructs an arbitrary polynomial $h(x)$ with the degrees of t-1. Let $h(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1x + K (\mathrm{mod}\, p)$, where each coefficient a_i belongs to [1, p-1] and $1 \leq i \leq t-1$. Notice that the secret K is hidden in the constant item of the above polynomial. It is easy to acquire the secret K from $h(x)$ since $h(0) = K$.

Suppose that each participant i has a unique and public identification ID_i. Next, the dealer will generate each shadow k_i according to each participant's ID_i. Let $k_i = h(ID_i)$, where $i = 1, 2, \ldots, n$. The pair $(ID_i, h(ID_i))$, i.e., (ID_i, k_i), can be seen as a coordinate point of the polynomial $h(x)$ in a two-dimensional space. Since $h(x)$ is a polynomial with the degree of t-1, it can be inferred that t or more than t coordinate points can determine the polynomial $h(x)$. In other words, the polynomial $h(x)$ can be reconstructed with any t pairs (ID_{p_1}, k_{p_1}), (ID_{p_2}, k_{p_2}), $\ldots$, and (ID_{p_t}, k_{p_t}) among n pairs. Here $1 \leq p_i \leq n$ and also $i \neq j$, $p_i \neq p_j$. The reconstruction can be achieved easily by the Lagrange interpolation polynomial which can be simply put as below:

$$h(x) = \sum_{s=1}^{t} k_{p_s} \prod_{j=1, j\neq s}^{t} \frac{x - ID_{p_j}}{ID_{p_s} - ID_{p_j}} (\text{mod } p).$$

With the cooperation of t participants, $h(x)$ can be reconstructed, and the secret K can be obtained easily by $h(0)$.

2.2 Torus automorphism

Given an image, Torus automorphism [5] scrambles it and generates a chaotic mixed image. We label each pixel with a coordinate (X_0, Y_0) defined by the grid. The actions of Torus automorphism on the given pixels are represented by the matrices $\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ q & q+1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} (\text{mod } N)$, $\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ q & q+1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} (\text{mod } N)$, and so on. The above formulas can be generalized as

$$\begin{bmatrix} X_d \\ Y_d \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ q & q+1 \end{bmatrix} \begin{bmatrix} X_{d-1} \\ Y_{d-1} \end{bmatrix} (\text{mod } N).$$

Here (X_d, Y_d) is a coordinate in a two-dimensional space. It is the result of Torus automorphism applied to (X_0, Y_0) up to d times. The parameters of Torus automorphism are q, d, and N. The value of q is assigned by the user arbitrarily. As for d and N, they denote the number of action times of Torus automorphism and the size of the given image, respectively.

3 Proposed method

The basic idea of our watermarking method is derived from the (t, n)-threshold scheme, and that is why we call it the threshold watermarking (TW) method. Suppose that there is a two-color bitmap O to be protected. First, TW asks the owner of O to define a polynomial with the degrees of t-1. Let this polynomial be $F(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1 x + a_0 (\text{mod } 257)$. The coefficients (i.e., a_{t-1}, a_{t-2}, $\ldots$, and a_0) of the polynomial are the watermark. They are defined and only known by the owner of O. TW combines this watermark with O. Next, TW divides O into a sequence of bytes and selects m bytes from that sequence randomly, and then it stores them in an array. For each byte of the array, if there is a number $x \in [1, m]$ that makes the value of the byte equal to $F(x)$, this byte is then a characteristic value. We assume that there are n characteristic values in m bytes. For each characteristic value, TW records the relationship between the value and its related x, and this finishes the combination process of TW.

There is of course one way to take off the bitmap after the above process of TW. If t or more than t unchangeable characteristic values remain in those n values, TW will reconstruct the polynomial $F(x)$ successfully. Then the copyright of the two-color bitmap will be verified by the retrieved watermark. This is done in the verification process of TW.

TW consists of two parts. One is the *combination* process, and the other is the *verification* process. Next, we shall describe these two processes in detail.

3.1 Combination

Given a two-color bitmap O, TW scrambles O following Torus automorphism first. After the scrambling process, the bitmap O is rearranged chaotically. Torus automorphism does not only scramble the bitmap O but also provides secure protection for TW. If any attacker wants to predict the characteristic values, they should know the parameters of Torus automorphism first. After scrambling the bitmap O, TW separates the mixed image into r blocks from left to right and from top to bottom. Each block consists of eight pixels. Since each pixel has one bit only in two-color bitmaps, each block has eight bits. Hence TW can take each block as a byte and store these bytes into the array A. A is a big array in general. For the sake of acceleration and security, TW randomly selects m bytes from the array A and sets them exclusively to be the characteristic values. For this purpose, we import a dedicated seed s as a pseudo-random-number generator (PRNG) and employ this PRNG to produce a specific sequence whose length is m, where m is smaller than or equal to r. Note that this specific sequence is associated with the seed s. When we import the same seed s to PRNG, TW acquires the random numbers in the corresponding sequence as mentioned above. Next, TW stores the random numbers in this sequence into another array B. Based on this array B, we can take m bytes from the array A and store them in still another array C. After the generation of C, TW constructs a polynomial $F(x)$ according to the watermark a_{t-1}, a_{t-2}, $\ldots$, a_0, and then it selects some characteristic values from the array C based on the values of $F(x)$. Here C[i] is defined as a *characteristic value* if and only if such an integer as $x \in [1, m]$ exists so that $F(x)$ is equal to C[i]. The relation between characteristic values and their associated x is recorded in the array D. Finally, the owner of O has to keep the parameters of Torus automorphism q, d, N; the watermark a_{t-1}, a_{t-2}, $\ldots$, a_0; and the array D. They are the input to the verification algorithm of TW. The combination procedures of TW are described in detail below:

Algorithm I [Combination of TW]

Input: A bitmap O, the watermark $\{a_{t-1}, a_{t-2}, \ldots, a_0\}$, the seed s of PRNG, three parameters q, d, and N for Torus automorphism.

Output: An array D that keeps the relationship between characteristic values and their associated x.

Step1: Scramble the bitmap O using Torus automorphism with parameters q, d, and N. Let the resultant bitmap be O'.

Step2: Split O' into r blocks. Each block consists of eight pixels. Let A[i] denote the value of the i-th block, where $1 \leq i \leq r$. The value of A[i] must belong to [0, 255].

Step3: Import a dedicated seed s to a PRNG and produce a sequence of random numbers with the length m. TW labels each random number as B[i], where $1 \leq i \leq m$. The value of B[i] must belong to [1, r].

Step4: Pick A[B[i]] from the array A and label it as C[i]. That is, let C[i] equal A[B[i]], where $1 \leq i \leq m$. The value of C[i] must belong to [0, 255].

Step5: Establish a polynomial $F(x)$ over GF(257), i.e.,$F(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1x + a_0 (\mathrm{mod}\ 257)$. The coefficients of $F(x)$ constitute the watermark.

Step6: Calculate $F(j)$, where $1 \leq j \leq m$, and store $F(j)$ in the array F.

Step7: Generate the array D. Set the value of D[i] to be j if there is an F[j] to ensure that F[j] is equal to C[i]. C[i] is the characteristic value in this case. Otherwise, D[i] equals 0. Both the values of i and j belong to [1,m].

Step8: Output the array D.

3.2 Verification

After the combination of TW, the owner of the original bitmap must keep some security information. They are the parameters of Torus automorphism, the seed of PRNG, the watermark, and the array D. These parameters will be further used to verify the copyright of the bitmap. The process of retrieving and verifying the watermark is the other part of TW. It is called the *verification* process.

The basic idea of the verification of TW is described as follows. Given a two-color bitmap O, suppose the owner wants to retrieve its watermark with TW. First, the user must employ the same three parameters (i.e., q, d, and N) in Torus automorphism to scramble the bitmap O. Next, TW divides the mixed bitmap O' into r blocks and reads these blocks as a sequence of bytes as mentioned before. TW stores them in an array A'. After the above process, TW generates the same sequence of random numbers by PRNG and the same seed s. Further, TW stores them in the array B. Based on the random number (i.e., B[i]), TW picks out the characteristic value from the array A' (i.e., A'[B[i]]). A'[B[i]] is a characteristic value if D[i] is nonzero. Next, TW checks if A'[B[i]] is equal to A[B[i]]. In other words, the pixels in the block A'[B[i]] are unchanged. Since the input of the verification algorithm does not include the array A, TW should check A'[B[i]] through F(D[i]). If A'[B[i]] is equal to F(D[i]), the content of A'[B[i]] has not been changed due to modificative attacks of the bitmap. TW then selects t different characteristic values from the array A' following the process above. With the t characteristic values, the polynomial $F(x)$ can be reconstructed using the Lagrange interpolation polynomial. The coefficients of $F(x)$ (i.e., the watermark) can then be retrieved and verified in the verification process. The verification algorithm of TW is expressed in detail below:

Algorithm II [Verification of TW]

Input: If a bitmap O needs to be verified, the inputs are the three parameters q, d, and N, the seed s of PRNG, the watermark$\{a_{t-1}, a_{t-2}, \ldots, a_0\}$, and the array D.

Output: Yes or No depends on whether the retrieved watermark $\{a_{t-1}', a_{t-2}', \ldots, a_0'\}$ is or is not the same as the input watermark $\{a_{t-1}, a_{t-2}, \ldots, a_0\}$.

Step:1 Scramble the bitmap O using Torus automorphism with three parameters q, d, and N. Generate a mixed bitmap O'.

Step:2 Divide the mixed bitmap O' into r blocks. Each block consists of 8 pixels. Let A'[i] denote the value of the i-th block, where $1 \leq i \leq r$. The value of A'[i] must belong to [0, 255].

Step:3 Feed the seed s to PRNG and produce a corresponding sequence of m random numbers. Label each random number as B[i], where $1 \leq i \leq m$, and the value of B[i] belongs to [1, r].

Step:4 Based on the random numbers B[i_j] chosen arbitrarily from the array B, select t different values A'[B[i_1]], A'[B[i_2]], ... , and A'[B[i_t]] from the array A' so that D[i_j] is nonzero and the A'[B[i_j]]'s are different from each other. The value of A'[B[i_j]] should also satisfy the expression A'[B[i_j]] = F(D[i_j]), where $F(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1x + a_0 (\mathrm{mod}\ 257)$. The value of i_j belongs to [1, m], and j belongs to [1, t].

Step:5 If there are t points (i.e., (D[i_1], A'[B[i_1]]), (D[i_2], A'[B[i_2]]), ... , and (D[i_t], A'[B[i_t]]) and the condition is satisfied as above, reconstruct the polynomial with the Lagrange interpolation polynomial. Output "Yes" to show that there is indeed a watermark in O. Otherwise output "No."

4 Experiment results

Some experiments on two-color bitmaps have been done to prove the robustness of our method. There were two bitmaps employed in our experiments. They are images of a Whale and a Horse,Figures 1 and 2, respectively. The size of the bitmaps used in our experiments is 512 × 512 pixels. Figures 3 and 5 are the modified version of the original Whale. They are Haze Whale and Snow Whale, respectively. Likewise, Figures 4 and 6 are the modified version of the original Horse. They are Haze Horse and Snow Horse, respectively. We experimented on these bitmaps to check the robustness of TW. Our experiments were conducted on a personal computer. It consists of an AMD K6 200 MHz CPU and 64 MB RAM. The operating system was MS Windows 98, and the programming language used is C++.

In our experiments, all parameters are fixed. We set t = 3, a_2 = 2, a_1 = 1, and a_0 = 7. As a result, the polynomial $F(x)$ employed in our experiments is $2x^2 + x + 7$ (mod 257). The parameters q, d, and N of Torus automorphism are 10, 5, and 512, respectively, and the seed s of PRNG is 7. According to these parameters, our experiments apply the combination algorithm of TW to the original Whale first to generate a coordinating array D. Next, to check the copyright of Whale, we use the fixed parameters and the above D in the verification algorithm. It takes a total of 3.14 seconds to retrieve the watermark 2, 1, and 7. It appears efficient. Thus TW verifies the copyright of the original Whale successfully. Besides the original Whale, we also check the copyright of Haze Whale and Snow Whale. We use the verification algorithm of TW to verify the copyright of Haze Whale and Snow Whale. It takes 3.23 seconds and 3.21 seconds to retrieve the watermarks of Haze Whale and Snow Whale, respectively. Consequently, TW is robust as to the verification of the watermark {2, 1, 7}.

After that, we apply the combination process of TW and another set of parameters to the original image of Horse. The three parameters q, d, and N and the seed s remain the same as before, that is 10, 5, 512, and 7, respectively. The parameter t is still three, but the watermark is not 2, 1, and 7 anymore. Instead, we set a_2 = 1, a_1 = 2, and a_0 = 3 this time. Namely, the original Horse is now protected by TW. We can also retrieve the watermark {1, 2, 3} successfully from the original Horse, Haze Horse, and Snow Horse following the verification process of TW. TW is nice and robust when retrieving the watermark from the modified bitmap.

>From the above results, it is shown that TW can verify the copyright of a bitmap efficiently and robustly. Besides this, TW is so secure that only legal users can retrieve and verify the watermark successfully. The inputs of the verification algorithm of TW are secret parameters, which are only known to the legal user (i.e., the owner of the original bitmap). They are the three parameters of Torus automorphism, the seed of PRNG, the watermark defined by the user, and the array D.

Besides the impressive robustness and execution time, the size of the storage space is also an important issue to qualify a watermarking method. In TW, besides the storage needed by the two-color bitmap with the size of 512 × 512, we need only some tiny extra storage room to store arrays D, F, the parameters of Torus automorphism, the seed of PRNG, and the watermark. As for arrays A, B, and C, they do not need to be put in storage at all. In the combination process, the values of the array A can be obtained from the mixed bitmap directly. Similarly, A' can be obtained from the mixed bitmap in the verification process. That is, no extra storage room is required for A and A'. The space of the array B can be reduced to a temporary byte by coding. TW reads the random number from the temporary byte. According to the random number, TW selects the related byte from the mixed bitmap. Instead of storing this related byte into the array C, TW compares the related byte with the values of the array F to find out the proper x, and then it stores the value x into the array D. Then, PRNG can generate the next random number and store it in the temporary byte. TW repeats the above action until the array D is created successfully. In other words, it is not necessary to store arrays A, B, and C; only one byte is required to store the random number. In our experiments, the storage room for the bitmap, arrays D, F, several parameters, and the temporary byte is totally less than 33KB.

Note that the values of m and t are defined by the user arbitrarily in TW, except the value of t should be much smaller than that of m. For example, in our experiments, the values of m and t are set to be 256 and 3, respectively. In fact, the execution time and storage space of TW is determined by m. The bigger the value of m is, the more the execution time and storage space required by TW will be. That is, the performance of TW will be worse if the value of m is bigger. However, when TW picks out more bytes from the bitmap (i.e., the value of m is larger), there will be more characteristic values in the array C, which will be useful for verification after modificative attacks. Hence the choice of the m value is a trade-off. By the same token, the choice of the t value is also a trade-off. The verification process of TW is determined by t. If the value of t is big (such as t=100) and modification of the bitmap is observable, it is then difficult to find as many as t unchanged characteristic values from the modified bitmap. TW cannot verify the copyright of the modified bitmap in this situation. Hence the value of t should not be too big. However, the value of t should not be too small, either. Otherwise, it will be too easy to find t characteristic values from a bitmap, even if this bitmap does not belong to the current user. Watermarking is a trade-off problem basically. For an image, a watermarking method has to find out the owner of the image even if this image has been modified. But, if this image has been completely changed and transformed to be another image, the watermarking method has to detect this change and decide that this new image does not belong to the owner of the original image. The value of t in TW is exactly what carries out this operation.

5 Conclusion

TW can protect the copyright of two-color bitmaps. The concept of TW is derived based on the (t, n)-threshold scheme. In the (t, n)-threshold scheme, the secret can be obtained through the cooperation of t participants. Similarly, for TW, the watermark can be obtained with t legal characteristic values. TW consists of two components. One is the combination process, and the other is the verification process. By the combination of TW, the owner of the original bitmap can combine the watermark and the bitmap together and generate some information that can be used to verify the watermark. The verification of TW works through importing this information to retrieve the watermark of the bitmap. When the bitmap is destroyed, if there are t or more than t unchanged characteristic values left, the polynomial can be reconstructed. Furthermore, the coefficients of the polynomial can be verified successfully. Based on our experiments, it is shown that TW can verify the copyright efficiently and robustly without the help of the original bitmap.

References

[1] O. Bruyndonckx, J-J. Quisquater and B. Macq (1995) Spatial method for copyright labeling of digital images. *IEEE Workshop on Nonlinear Signal and Image Processing, Vol. 1*, p.456-459.

[2] W. Bender, D. Gruhl, N. Morimoto and A. Lu (1996) Techniques for data hiding. IBM Systems Journal, Vol. 25, p. 315-335.

[3] C. T. Hzu and J. L. Wu (1997) Digital watermarking for video. *IEEE Digital Signal Processing Proceedings, Vol. 1*, p. 217-220.

[4] A. Shamir (1979) How to Share a Secret. *Comm. ACM, Vol.* 22, p. 612-613.

[5] G. Voyatzis and I. Pitas (1996b) Applications of Toral automorphisms in image watermarking. *Proc. of ICIP96, Vol.* 2, p. 237-240.

Figure 1: Original Whale.

Figure 2: Original Horse.

Figure 3: Haze Whale.

Figure 4: Haze Horse.

Figure 5: Snow Whale.

Figure 6: Snow Horse.

A Framework for Query Formulation Aid in a Multi-user Environment

Mourad Oussalah and Abdelhak Seriai
Site EERIE/LGI2P de l'EMA
Parc Scientifique G.BESSE
30035 Nîmes cedex 1 France
Fax: +33 4 66 38 70 74
E-mail:{oussalah, seriai}@eerie.fr

Keywords: Object-oriented database, query formulation, reuse, user help, cooperation

Edited by: Yanchun Zhang, Vladimir Fomichov and Anton P. Železnikar
Received: August 15, 1999 **Revised:** November 4, 1999 **Accepted:** November 10, 1999

The formulation of queries is often regarded as a difficult task for a large class of users. In this article we propose an approach facilitating query formulation for users sharing a common database. Our approach is based on two considerations. On one hand, query formulation can be considered as a skill which can be shared entirely or partly between users. Accordingly we propose a model for storing and making available this skill to help users to formulate their new queries. On the other hand, we base our design on custom construction of a database adapted to each business group of users in question. In using it, the users of a given business cooperate transparently through their queries to design databases specific to their business. Thus, they take part co-operatively, if indirectly in reducing the difficulty of query formulation.

1 Introduction

Database query languages aim to allow users to obtain the data necessary for the realization of their tasks. However, query formulation has always been regarded as a difficult task for a large class of users (Denneboury, 1993). The designers of database systems have been proposing languages, which are increasingly directed towards the needs of users. The principal aim has been the simplification of the syntax used in formulating queries. Indeed, these languages have passed from algebraic (Codd, 1970) through declarative forms of expression (Kim, 1990) and on to visual languages relieving the user almost totally from the need to know the syntax concerned (Vadaparty, 1993). Nevertheless, query formulation remains an unintuitive and delicate task. There are multiple reasons for this, some of them being the following:

- The size of the database schemas handled by users often exceeds their assimilation capacity. These schemas usually shared by different groups, which manipulate different relations and entities. These groups usually pertain to different skills.
- The structural and semantic variation between the vision that users have of their data and the database schema model representing these data. The Absence of schema evolution is one cause of this divergence (Lenner & Habermann, 1990).
- The need for formulating queries using semantics, which are increasingly difficult to express. To formulate these queries, the user will have to answer multiple questions, examples of which might be: Which entities in the schema correspond most closely to the information requirement? Which properties, in terms of attributes and methods, to use? What conditions must these entities verify and how should they be combined? (Fleury, 1996). Aided decision applications, and those using a data warehouse are good examples of the need for complex query semantics.

In this article, we propose an approach which allows using user queries to try to find a solution to these problems and thus to reduce the difficulty of query formulation. The means we adopt for this purpose include:

- Reducing both the size of the databases (schema and data) and the structural and semantic mismatch between users' views of their data and the schema model. The idea is to build local databases adapted to each business by an incrementall design. These local databases are built out of a common database shared by all users. Local schemas are created in a transparent way using user queries. Then, data necessary to populate these databases are extracted from the common database.
- Offering help to users in formulating their queries. This consists of providing users with a range of alternative formulations for their query. These are built from information held about their needs. This information can be for example, a list of attributes or a list of classes that must be used in a query expression, a summary of query expressions (for instance specifying that query structure takes the form *select*

from where), etc. The formulations suggested by the model are extracted from a query definition database, which is progressively built up from queries formulated by users. Our help model is based on the assumption that the users of any business or group of similar businesses often formulate queries which are semantically close. The re-use of the whole or parts of previously formulated queries is thus regarded as a re-use of skills.

The paper is organized as follows. In Section 2, we present the principle of the suggested approach. The query formulation assistance and the local databases incremental design processes are detailed in Section 3. In Section 4, we present the related works. We conclude in section 5.

2 The Principle of the approach

The aim of our work consists of offering to users who exploit a common database in a multi-user environment, a framework for reducing the difficulty of query formulation. The approach is based on the separation and the re-use of two facets of a query: its definition representing its syntactic structure and its result representing the structure of the objects supplied in response to it.
Query definitions formulated by users are reified and organized for re-use by other users. In this way, the re-use of closely related queries allows facilitating the formulation of new queries by users belonging to the same skill groups.

Query results for their part are used for the incremental construction of a local database. This database will thus contain only the entities conceived and chosen by the users themselves through their queries. This method reduces both the size of the schema handled by the users and the mismatch between the view which users have of their data and the schema model.
Figure 1. summarizes the principle of our approach. Users sharing the exploitation of a common database can belong to different businesses (X and W in Figure 1). Let us imagine that our model is in use by the user group of business W only. We propose two modes of use: an interrogation mode and an assistance mode.
In interrogation mode, a query formulated by a user of this group launches the execution of two parallel processes simultaneously:

- The incremental construction process of a database (which is initially empty) local to the business W user group. This process consists of: first, identification and reification (materialization) of the set of classes necessary to answer it (step A1 in Figure 1). Secondly, classification of the classes resulting from the previous step for incorporation into the local database schema (step A2 in Figure 1). Lastly, extraction from the shared database of the data necessary for creation of instances of classes newly incorporated into the local database schema (step A3 in Figure 1).

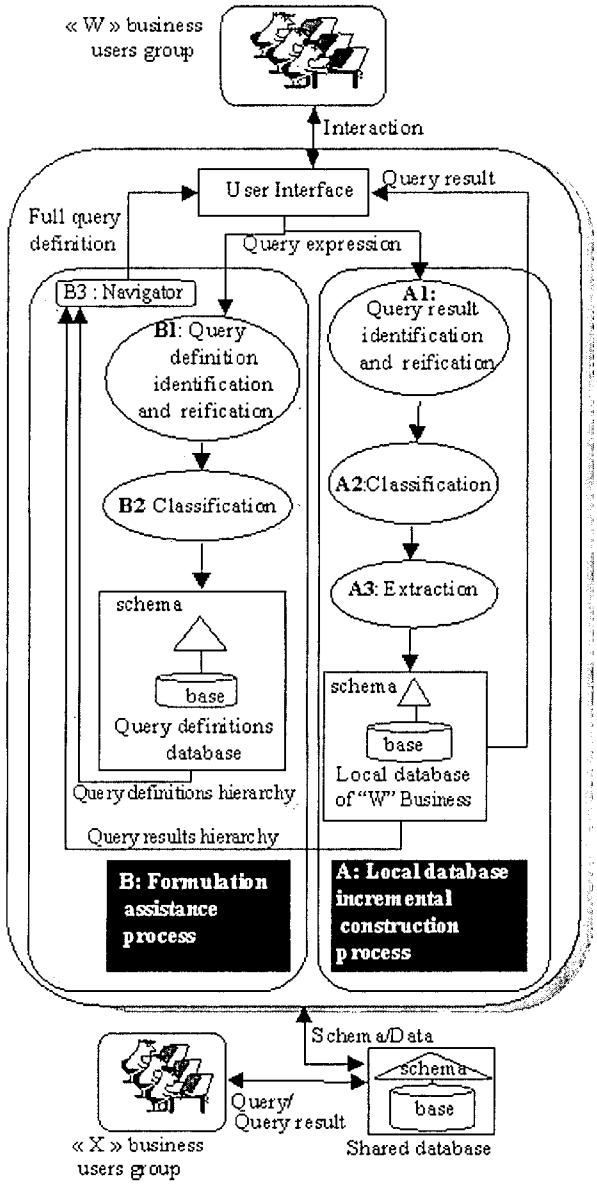


Figure 1: The principle of the suggested approach

Once these three steps have been taken, a response to the query is then found for the user by querying the local database.

- The formulation assistance process in its phase of enriching the query definition database (we called it the learning phase). This process phase consists of: first, identification and reification of the object classes representing the query definition (step B1 in Figure 1). Secondly, classification of the classes resulting from the reification of the query definition for incorporation into the hierarchy of query definition (step B2 in Figure 1).

The assistance mode can be called upon by a user who finds difficulty in formulating a given query. The formulation assistance process in its use phase is then launched.

This phase aims to propose to users, already formulated queries corresponding eventually to there needs. This consists of: first, executing the same steps as those of the learning phase but using a partial definition. However, classification in this phase allows inferring full definitions - of which the partial definition can form a part. Secondly navigation, which is used as a means of interaction between model and user (step B3 in Figure 1). In effect, we use a navigator which gives to the user the possibility of browsing the definition and result hierarchies. The user can then adapt or complement the formulations that the model will have proposed to him/her and which will be indicated on the query definitions hierarchy graph.

3 Model

3.1 Concepts

We distinguish two types of query definition, full and partial. Full definitions are formulated by users to elicit responses in terms of data (interrogation mode). Partial definitions are created by users to obtain assistance in formulating queries correctly (assistance mode). For the representation of query definitions, we define two types of entity, elementary and complex. Elementary entities represent algebraic operators (projection, selection, join). Complex entities represent the grammatical structure used by users to formulate a query in a specific query language (*select from where* queries, set queries, arithmetic queries, etc.). Each complex entity is composed of elementary entities issued from query expression rewriting in an algebraic form.

3.2 The formulation assistance process

3.2.1 Identification and reification of 'query definition' classes

Identifying the class representing the query definition is a matter of analyzing the query to determine the type and structure of the object class corresponding to it. This means, specifying which complex operator is used for the query expression. This one is rewritten using algebraic operators represented by elementary entity classes.
All the classes representing the same type of elementary entity are grouped in a common composite class: projection, selection and join component classes. Figure 2 represents the classes corresponding to the query *the list of any customers who are employees of the company ?* using UML formalism (Booch & al., 1998). This query definition is given below in OQL query language (Alashqur & al., 1989).

Select	Customer.name, Employee.name
From	Customer in The-Customers,
	Employee in The-Employees
Where	Customer.name = Employee.name

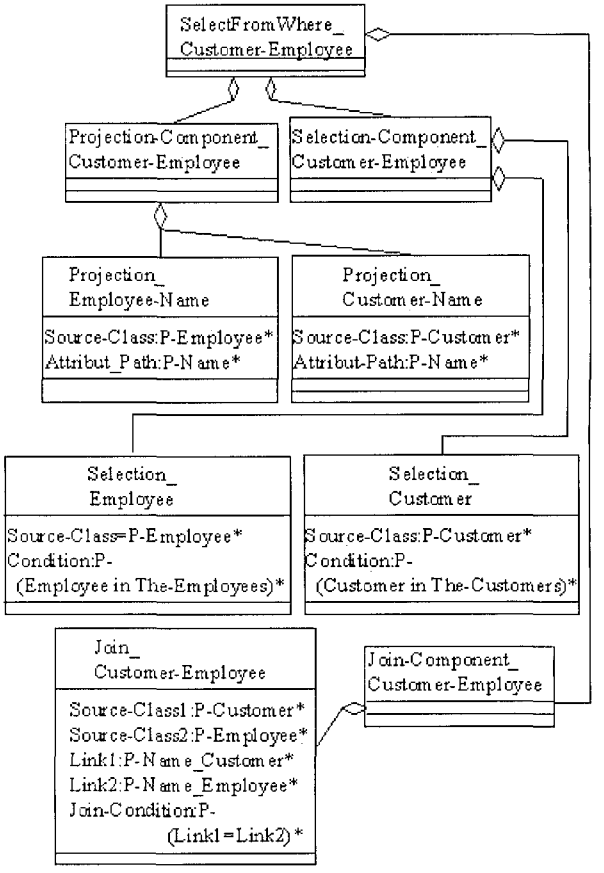


Figure 2: Example of a query definition class

The classes represented in Figure 2 are a representation of the following algebraic translation of the query presented above. The classes attributes represent the parameters of algebraic operators (projection, selection and join). For example the *Source-class1, Source-class2, Link1, Link2, Join-condition* attributes defined in the Join-Employee-Customer class represent respectively two class descriptions necessary to carry out the join operation, two links defined in these classes, and the join condition.

Projection	[Projection-1 (Customer, name)
	Projection-2 (Employee, name)]
Selection	[Selection-1 (Customer, in,
	The-Custoimers),
	Selection-2 (Employee, in,
	The-Employee)]
Join	[Join-1(Customer, name,
	Employee, name, Equality)]

3.2.2 Classification of 'query definition' classes

The classes resulting from the identification stage are inserted in a 'query definition' classes hierarchy (see figure 1). This maintains the set of specialization/generalization

relations existing between these classes. Thus, these relations represent the fact that a particular query definition is more general or more specific than another. The classification of classes representing a partial query definition allows the finding of classes representing already formulated queries which can be a complement of this partial one. The classification algorithm used in the model is an adaptation of the algorithm defined in (Napoli, 1991). This particular one is characterized by two basic components: pattern matching and a class graph search, aimed to find the best position in the class graph. For example, the algorithm indicates that the query *select customer.name, Employee.name from ? where ?* can be inserted as a subclass of the root class representing the query of Figure 2.

3.3 The incremental conception process of local database

3.3.1 'Query result' classes identification and reification

Identification consists of determining all the classes needed to answer a query which are not in the local database schema. These class descriptions are later retrieved from the shared database schema. For example, the execution of identification step of query presented in paragraph 3.2.1 determines that classes representing the customers and the employees of the company must be integrated into the local database to answer the concerning query.
In addition to data consultation, some queries are used to create new classes not defined in the shared database schema; this is by reifying query results and incorporating them in the local database schema. This reification function allows adapting the schema classes to the needs of the different business categories of the user. The structures of derived classes are automatically inferred from their query definitions. The model can thus infer the type of these classes, their attributes and of course the types of these attributes.

3.3.2 Classification of 'query result' classes for incorporation into local database schema

The position of classes resulting from the identification stage in the local database class hierarchy is inferred automatically from their position in the shared database class hierarchy. Indeed, for each class, the identification stage retrieves its set of super-classes. The class thus keeps the same position in the hierarchy. For example, the *Employee* class used in the query of paragraph 3.2.1 is inserted as a subclass of class *Person*. The class *Person* representing the super class of class *Employee* in the shared schema is inserted also in the local one.
Classes resulting from reification are inserted in positions calculated by a specific classification algorithm.

3.3.3 Object extraction and instantiation

The extraction stage in the local database construction process consists of extracting the data necessary to create the local database schema class instances. For creation of these instances, we distinguish classes resulting from the identification stage from those resulting from the reification stage. Indeed, instances of the classes resulting from identification will be copies of those of the classes belonging to the shared schema. For those resulting from reification and representing new entities, new instances are created.

4 Discussion and related work

To our knowledge, all approaches proposing to facilitate the syntactic aspect of queries only -formal, declarative, visual, for instance. We find no approaches in the literature which propose the re-use of syntactic and semantics inherent to already formulated queries for the formulation of new queries -which are potentially more complex. However, we do acknowledge a link between the incremental conception of local databases presented here and the numerous approaches based on use of the view and materialized view concepts ((Abiteboul & Bonner, 1991), (Heiler, 1990), (Ullman, 1988)). We distinguish two fundamental classes of view approaches. The first proposes creating partial views adapted to particular groups of users i.e. views which are separate from the common database schema ((Souza, 95), (Abiteboul & al., 1994)). This though doesn't allow having a complete and integral view of the entities available to a group of users. New entities created by users (which in our approach arise from the query result reification step) are separated from those already in the global schema (which in our approach arise from the query result identification step).
The second class of approaches works by integrating a specific user view to the global database schema ((Kuno & al., 1995), (Kuno & Rundersteiner, 1996)). These, in contrast of our approach don't allow the creation of separate and independent users views. This amplifies the users assimilation difficulty (v. introductory paragraph).

In addition, our approach has some similarities with some works related to structure semi-structured data. Indeed, our approach resembles that of the AKIRA system (Lacroix & al. 1998) in its use of user queries for incrementally constructing local databases. It differs from it in taking a shared database as its data source - not material downloaded from the Web. Thus it is not a case of one user building an individual smart-cache database, but a number of users belonging to the same business cooperating to build their own business database transparently. In addition, we reify the query result to adapt the content of the database to the needs of its users - which AKIRA doesn't do.

Re-use of queries is used in some works to optimize query evaluation (query semantic optimization) ((Abiteboul & Duschka, 1998), (Chaudhuri, 1995), (Duschka,

1997)). However, in addition to differing in our objective (query formulation assistance rather than query optimization), we manipulate the query definition concept rather than the view concept (query result concept in our approach). This choice allows:

- First, distinguishing between the different facets of use of query reification. On one hand, the view mechanism to represent novel data entities manipulated by users; on the other, the reuse mechanism to represent query formulation expressions.
- Second, the use of structure inherent to query expression, specifically the conditional parts of selection and join operators. Using this structure allows us to infer all the generalization, specialization or composition relations between different queries. This is impossible when one has recourse only to exploiting view (query result) (Staudt, 1994).

5 Conclusion

In this article we have proposed an approach designed to facilitate query formulation. It consists in use/reuse of queries formulated by users sharing a common database. For the part of the model concerned with help with query formulation, we take inspiration from work investigating the use of heuristic classification in CAD systems (Clancey, 1993). We take partial query formulations to be incomplete design models and correct query formulations to be complete design models. To our knowledge, no model has yet proposed exploiting both results and definitions of queries to assist users in their formulation. In our future work, we intend to investigate the use of patterns (Gamma & al., 1996) to bring users more effective help as they formulate their queries.

References

[1] Abiteboul S. & Bonner A. (1991) Objects and views. *Proceedings ACM SIGMOD Conference on Management of Data*, San Francisco, California, p. 238-247.

[2] Abiteboul S. Delobel C. & Souza dos Santos C. (1994) Virtual Schemas and Bases. *Proceedings EDBT 94.*

[3] Abiteboul S. & Duschka O. (1998) Complexity of answering queries using materialized view. *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS 98*, Seattle, Washington, USA.

[4] Alashqur A. Su S. & Lam H. (1989) OQL: a query language for manipulating object-oriented databases. *Proceeding VLDB 89*, Amsterdam, Holland.

[5] Booch G. Rumbaugh J. Jacobson I. (1998) The Unified Modeling Language User Guide, *Addison-Wesley ISBN 0-201-57168-4.*

[6] Chaudhuri S. Krishnamurthy R. Potamianos S. & Shim K. (1995) Optimizing queries with materialized views. *Proceedings ICDE 95, IEEE Comput. Soc. Press*, p. 190-200, Los Alamotos, CA.

[7] Clancey W.J. (1993) Heuristic Classification, *Artificial Intelligence, Vol. 27*

[8] Codd E. (1970) A Relational Model of Data for Large Shared Data Bank. *ACM Transactions on Database Systems Vol. 13.*

[9] Dennebouy Y. (1993) Un langage visuel pour la manipulation des données. *Thesis, Ecole Polytechnique Fédérale de Lausanne (in French).*

[10] Duschka O.M. & Genesereth M.R. (1997). Answering recursive queries using views. *Proceedings of the Sixteenth ACM PODS conference*, p.109-116, Tuscon, AZ.

[11] Fleury.L. (1996) Knowledge discovery in databases for staff management. *Ph.D. Thesis, university of Nantes (in French).*

[12] Gamma E. Helm R. Johnson R. & Vlissides J. (1996) Design patterns. *International Thomson Publishing.*

[13] Heiler S. & Zdonik S.B. (1990) Object Views: Extending the vision. *IEEE Data Engineering Conference*, Los Angeles, USA, p. 86-93.

[14] Kim W. (1990). Introduction to object-oriented databases. *The MIT Press.*

[15] Kuno H. & Rundensteiner E. (1995) Object-Slicing: A Flexible Object Representation and Its Evaluation. *Technical Report*, EECS Dept., University of Michigan, CSE-TR-241-95 Ann Arbor.

[16] Kuno H.A. & Rundersteiner E.A. (1996) The Multi-View OODB View System: Design and Implementation. *Journal of Theory and Practice of Object Systems. Special Issues on Subjectivity in Object-Oriented Systems.*

[17] Lacroix Z. Sahuguet A. & Chandrasekar R. (1998) User-oriented smart-cache for the Web: What You Seek is What You Get!. *ACM SIGMOD Research prototype demonstration*, Seattle, Washington, USA.

[18] Lerner B. and Habermann A. (1990) Beyond Schema Evolution to Database Reorganization. *Proceedings ACM Conf. OOPSLA and Proc. ECOOP*, P.67-76.

[19] Napoli A. (1991) Classification in Object-Based Representations, *8th AFCET congress pattern recognition & AI Vol. 1*

[20] Souza C. Design and implementation of Object-Oriented Views. *Proceedings of the 6th DEXA, Springer verlag*, London, UK.

[21] Staudt M., Nissen H.W. and Jeusfeld M.A. (1994) Query by Class, Rule and Concept.*Special Issue on knowledge Base Management. Vol. 4, No.2*, p. 133-157

[22] Ullman J.D (1988) Principles of Database and Knowledgebase Systems, *Computer Science Press Vol.1*

[23] Vadaparty K. Aslandogan Y.A. & Ozsoyoglu G. (1993) Towards a unified Visual Database Access. *Int. Conf. Of SIGMOD*, pp.357-366.Washington, USA

Evaluating Word Similarity in a Semantic Network

Masanobu Kobayashi, Xiaoyong Du and Naohiro Ishii
Dept. of Intelligence and Computer Science
Nagoya Institute of Technology
Gokiso-cho,showa-ku,Nagoya 466-8555,JAPAN
Phone: +81 52 735 5474, Fax: +81 52 735 5473
E-mail: {mkoba,duyong,ishii}@egg.ics.nitech.ac.jp

Keywords: word similarity, semantic similarity, information content, query expansion, information retrieval

Edited by: Yanchun Zhang, Vladimir Fomichov, Anton P. Železnikar
Received: August 15, 1999 **Revised:** November 8, 1999 **Accepted:** November 15, 1999

Evaluating the semantic similarity of a pair of words is a basic activity in text information search and retrieval. It can, for example, be applied to query expansion to support an intelligent information retrieval system. This technique makes it easy to find relevant information from the World Wide Web (WWW) even though users cannot input all the keywords which might express their needs. For these types of systems, similarity measures are required to closely approximate human judgement. In this paper, we propose a new measure of word similarity based on the normalized information content of concepts in a semantic network. It overcomes shortcomings in existing measures. The result of experimental evaluation indicated that our measure can judge word similarity like human beings, a correlation of 0.81, which is much higher than that of existing measures.

1 Introduction

Recent advances in the Internet make it easier to access available information on the WWW. However it is not so easy to find the valuable and relevant information we need from a huge resource such as the WWW. In a classical information retrieval system, a user poses a set of keywords and the system responds by supplying the user with a set of documents that contain those keywords. The quality of the retrieval results depends largely on the keywords posed by the user. It is a natural requirement for an intelligent information retrieval system to respond with not only the documents which contain the keywords posed by the user but also those which contain keywords similar to the posed ones. In other words, while searching a document with the query, the system should expand the query keyword set by adding those similar keywords based on a semantic network. To measure word similarity is a fundamental task in the intelligent information retrieval. In addition, we can classify documents automatically by using the semantic similarity of words in the document. Some keywords in the document are compared with a classname to evaluate the similarity between the class and the document. In this way, we can find a proper class for the document.

Generally speaking, one word has several senses. We call them "concepts" in this paper. To measure word similarity with respect to its senses, we have to measure concept similarity first. There are already some existing concept similarity measures using semantic networks. The most simple way is to use the "distance" between two concepts in the semantic network(Lee et.al. 1993). In this method, the shorter the path from one concept to the other, the more similar they are. Another method uses the information content(Ross 1976) of a superior concept which subsumes the two concepts to be compared(Resnik 1995). The information content of a concept can be quantified by the probability of the concepts it subsumes. In this method, two concepts are considered to be similar if the information content of their superior concept is high.

As we will show in the following sections, both measures, however, suffer from some critical problems. In the distance-based measure, it is difficult to define the distance between concepts. Usually, distance is defined by the number of edges between two nodes in a semantic network. But there is not sufficient reason to assign the same weight for all edges. In the information content based method, concept similarity is very related to the number of nodes subsumed by the two concepts. Two abstract concepts, which subsume a lot of concepts, will never have a high similarity measure (We describe this problem in detail in the following sections). In this paper, we present a new information content based similarity measure which overcomes all the shortcomings of existing measures.

2 A Framework for Evaluating Word Similarity

In natural language, a word usually has several senses called concepts. In other words, a word is just a label for these different concepts. Therefore, it is necessary to compare concepts when we measure semantic similarity

between words.

Two kinds of information are needed in evaluation of word similarity.

1. *Relationship between words and concepts.* It is a mapping from a word to a set of concepts which can be represented as a table (Table 1). This relationship can be viewed as the definition of a word. For example, the word "mouse" means certain small animal in general, but it has other meanings such as an input device of computers. People use the same word "mouse" to represent two completely different concepts.

Word	Concept1	Concept2	...
mouse	animal	device	...
head	body parts	chief	...
:	:	:	:

Table 1: Mapping Table

2. *Relationship between concepts.* It can be represented as a semantic network which is a directed acyclic graph(DAG)(Figure 1). There are different semantic relations between concepts in a semantic network, such as is-a, a kind of, a part of, and so on. Similarly to the existing methods, we consider only the is-a relation in this paper. For example, concept "cat" is a "feline", and "feline" is a concept of "animal". A dot line indicates that some intermediate nodes are omitted in Figure 1.

Figure 1: A part of a semantic network

Based on the two kinds of information, the similarity of a pair of words can be evaluated in the following framework. Assume we evaluate the similarity of two words w_1 and w_2. We first map the two words to a set of corresponding concepts respectively, say C_1 and C_2. We then evaluate the similarity of each pair of concepts which come from concept set C_1 and C_2, respectively. Finally, we get the word similarity from the obtained concept similarities. This process is shown in Figure 2.

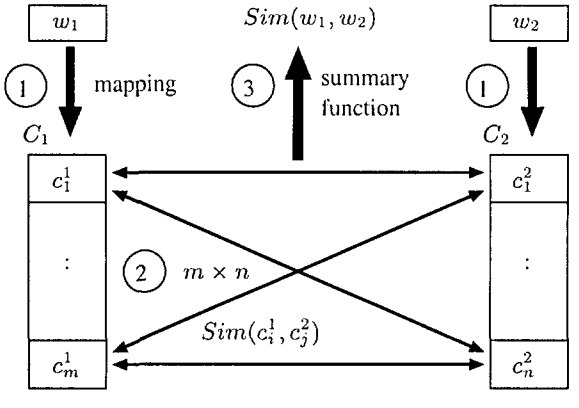


Figure 2: Process of evaluating similarity

In the following, we formalize this framework.

Definition 1 *Let W be a collection of labels called words and C be another collection of labels called concepts. $W \cap C = \phi$. A binary relation $d(w, c)$ belongs to $W \times C$ is defined for a word $w \in W$ and a concept $c \in C$ such that the concept c is a sense of the word w. The set of relations $d(w, c)$ is denoted as D. $C(w) = \{c | d(w, c) \in D\}$ is called the definition of the word w.*

Definition 2 *A semantic network is a DAG $G(C, E)$, where C is a set of nodes each of which corresponds to a concept, and E is a set of edges each of which corresponds to an is-a relation between two concepts. For a DAG, we can define a transitive closure relation T as usual. Each relation in T is called a path in the DAG. If $(c_1, c_2) \in T$, we say c_2 is a superior concept of c_1. The length of the path from c_1 to c_2 is called the distance between them, denoted by $dis(c_1, c_2)$. If $(c_1, c_3) \in T$ and $(c_2, c_3) \in T$ are two paths in G, then we say c_3 is a common superior concept of c_1 and c_2. The set of all common superior concepts of c_1 and c_2 is denoted as $CS(c_1, c_2)$.*

Example *Consider the semantic network in Figure 1. Concept "feline" is a superior concept of concept "cat". It is also a superior concept of concept "lion". Therefore, concept "feline" is a common superior concept of concept "cat" and "lion" Similarly, concept "animal" is also a common superior concept of "cat" and "lion".*

Next, we describe the process of evaluating the similarity of a pair of words w_1 and w_2. It consists of the following three steps.

1. Construct $C(w_1) = \{c | d(w_1, c) \in D\} = \{c_i | 1 \leq i \leq n\}$ and $C(w_2) = \{c | d(w_2, c) \in D\} = \{c_j | 1 \leq j \leq m\}$.

2. Evaluating the similarity $sim(c_i, c_j)$ for all $1 \leq i \leq n, 1 \leq j \leq m$.

3. Evaluating $sim(w_1, w_2) = F(\{sim(c_i, c_j) | 1 \leq i \leq n, 1 \leq j \leq m\})$, where F is a summary function. F is usually a maximal function. This is because there are some nonsense pairs of concepts in which the similarity will be very low, and we do not want them to affect the similarity of the two words.

Obviously, how to evaluate the concept similarity in a semantic network is the key to this framework. In the next two sections, we first introduce two existing methods of measuring concept similarity as well as their shortcomings, and then propose a new measure for word similarity.

3 Distance-based and Information content based Measurement

There are already some methods to measure concept similarity in semantic networks. The most simple way is using the "distance" between two concepts(Lee et.al.1993). The shorter the distance between two concepts, the more similar

they are. However, two concepts are usually not located in a path. Therefore it is necessary to redefine the "distance" for a pair of concepts.

Definition 3 *Let c_1 and c_2 be two concepts in a semantic network. Assume that $CS(c_1, c_2)$ is the set of common superior concepts of c_1 and c_2. Then the distance between c_1 and c_2, denoted by $dis(c_1, c_2)$, is defined as follows:*

$$dis(c_1, c_2) = \min_{c_3 \in CS(c_1, c_2)} (dis(c_1, c_3) + dis(c_2, c_3)). \quad (1)$$

Definition 4 *Let c_1 and c_2 be two concepts in a semantic network. A distance-based measure for the similarity of concept c_1 and c_2 is defined as follows:*

$$Sim_{edge}(c_1, c_2) = 2H - dis(c_1, c_2) \quad (2)$$

where H is the maximal length of paths in the semantic network.

This measure suffers from two problems. First, concept similarity is measured by the number of edges between concepts. It is therefore strongly dependent on the semantic networks we used. If an unappropriated semantic network is created or selected, we can not measure concept similarity accurately. Second, there is no sufficient reason to assign all edges with the same weight 1.

To overcome the disadvantages of the distance-based measure, P.Resnik proposed an information content based measure(Resnik 1995). In contrast with the distance-based method, this measure does not consider the edges but the subsumption relation among nodes in the semantic network.

In this method, the information content of a concept is derived from its probability(Ross 1976).

$$info(c) = -\log_2(p(c)) \quad (3)$$

where $p(c)$ can be estimated by the relative frequency of c appearing in a corpus, a large collection of preprepared documents. That is,

$$p(c) \approx \hat{p}(c) = \frac{freq(c)}{N} \quad (4)$$

where N is the total number of occurrences of nouns in the corpus(excluding those not appeared in the semantic networks) and $freq(c)$ is computed in the following way.

Let $word(c) = \{c' | (c', c) \in T\}$, that is, the set of concepts subsumed by the concept c. This set $word(c)$ can be treated as a set of nouns, and we can count the appearance frequency of each noun in $word(c)$. Assume $count(n)$ is the count of n's occurrence in the corpus. Then,

$$freq(c) = \sum_{n \in words(c)} count(n). \quad (5)$$

Definition 5 *Let c_1 and c_2 be two concepts in a semantic network. A information content based measure for the similarity of concepts c_1 and c_2 is defined as follows:*

$$Sim_{info}(c_1, c_2) = \max_{c \in CS(c_1, c_2)} [info(c)]. \quad (6)$$

An experiment by P.Resnik suggests that information content based similarity performed better than the distance-based measure(Lee et.al.1993). However, it still suffers from some problems. First, information content is affected by the corpus. It is difficult to select appropriate samples (documents) to construct a corpus. In addition, it takes a lot of time to evaluate the information content of a concept.

4 A New Measure for Concept similarity

By observing semantic networks, we can easily find the fact that the closer to the bottom a concept is in the semantic network, the more concrete the concept. Obviously, a concrete concept has more information content than an abstract one. For example, consider the semantic network in Figure 1. The pair of concepts "cat" and "lion" is considered more similar than the pair of concepts "cat" and "frog", because there is a common superior concept of "cat" and "lion" which is located lower than the common superior concepts of pair "cat" and "frog". Concept "feline" is more concrete than concept "animal". This abstractness can be used to measure the information content of a concept.

More important, the previous method considers the information content of a concept as an absolute value and compares them to each other. If fact, this information content is a relative value. That is,

$$0 \leq sim_{info}(c_1, c_2) \leq max(info(c_1), info(c_2)).$$

Two abstract concepts, although they are very similar, may have a very low similarity. Hence, we have to "normalize" the information content before we compare them.

Let us consider a special case, that is to compare two identical concepts. Obviously, its similarity should be the highest value 1. However, the previous information content based measure gets a value which is usually less than 1 and depends on its position in the semantic network, that is the degree of abstraction of the concept.

The above shortcomings should be overcome in our *normalized measure.*

Definition 6 *Let $G(C, E)$ be a semantic network. The information content of a concept c w.r.t. G can be defined as follows:*

$$info(c) = -\log_2(\frac{subs(c)}{|C|}) \quad (7)$$

where function $subs(c)$ stands for the number of concepts which c subsumes. That is, $subs(c) = |\{c' | (c', c) \in T\}|$.

Definition 7 *Let c_1 and c_2 be two concepts in a semantic network. A normalized measure for the similarity of concept c_1 and c_2, denoted by Sim_{nrml}, is defined as follows:*

$$Sim_{nrml}(c_1, c_2) = \max_{c_{12} \in CS(c_1,c_2)} (\alpha \times info(c_{12})) \quad (8)$$

where $\alpha = max(info(c_1), info(c_2))^{-1}$.

5 Evaluation

We designed a set of experiments to evaluate our measure by comparing it with the other two measures as well as human judgement. A semantic network called WordNet1.6, which was developed at Princeton University by George A. Miller (Miller et al,1990), is used in our experiment. WordNet is a hand-crafted, general-purpose thesaurus. Therefore, we can use this semantic network to evaluate word similarity for general applications.

In order to get a human judged word similarity, 27 students were given 40 pairs of words(Table 3). We asked these students to rate similarity for each pair on a scale from 0 to 4.(The larger the number, the more similar). All subjects were computer science graduate and undergraduate students. We use the average of these ratings as the human judged word similarities. We also evaluate the word similarity measured by our method for the same 40 pairs of words, and compared it against the human judgements.

Please note that to implement the information content based measure, we have to prepare some corpuses for similarity. For simplicity, we don't use corpus in this experiment. Instead, we set the function $count(n)$ in both Sim_{info} and Sim_{nrml} to 1.

The correlation between our similarity measure and the human judgement came out to be of a very high value, 0.81, while both the other measures had lower correlations (0.73 and 0.62, respectively). This result showed that our measure is better than the other two measures.

The human judged similarity is an average value, and each person had different judgement about these 40 pairs of words. If we treat the average judgement as a standard one, we can also evaluate the correlation between a person's judgement and the standard one. Table 2 sorted all correlations. We noted that our method was in the exact middle of the list. This means our method can judge word similarity like human beings.

Rank	Subject	Correlation
1	student1	0.9254
:	:	:
21	student21	0.8249
22	$\mathbf{Sim_{nrml}}$	**0.8160**
23	student22	0.7938
:	:	:
27	student26	0.7481
28	Sim_{info}	0.7366
29	student27	0.6941
30	Sim_{edge}	0.6268

Table 2: Experimental Result

6 Conclusion

This paper presents a new measure of word similarity based on information content. Our method judges word similarity using concepts represented by a semantic network. We believe that good similarity measures can produce new advances in the information search and retrieval system by keyword expansion. This makes it much easier for users to operate the system.

The difference between our similarity measure and the existing methods is proximity with the human judgement. The experimental results indicate that our method can judge word similarity like human beings, and is more accurate than the other methods.

Future work is to apply our similarity measure to a practical information search and retrieval system, and evaluate its performance.

References

[1] Lee,Joo Hoo et al(1993)Information Retrieval Based on Conceptual Distance in IS-A Hierarchies;Journal of Documentation,49(2),p.188-207.

[2] Miller,A.,George et al(1990)Introduction to WordNet:An On-Line Lexical Database;In Princeton University (Ed.),Cognitive Science Laboratory: Five Papers on WordNet ,p.1-10,Princeton University. `http://www.cogsci.princeton.edu/~wn/`

[3] Rada, Roy et al(1989)Development and Application of Metric on Semantic Nets;IEEE Transaction on Systems,Man, and Cybernetics,19(1),Feb,1989,p.17-30.

[4] Resnik, Philip(1995)Using Information Content to Evaluate Semantic Similarity in a Taxonomy;Proceedings of IJCAI-95,p.448-453.

[5] Ross,Sheldon(1976)A first Course in Probability;Macmillan.

[6] Winston,H.,Patrick(1984)Artificial Intelligence (3rd edition),Addison-Wesley.

No	Word Pair		Sim_{human}	Sim_{edge}	Sim_{info}	Sim_{nrml}
0	car	automobile	3.852	4.000	2.798	4.000
1	gem	jewel	3.777	4.000	4.000	4.000
2	boy	lad	3.740	3.857	3.075	3.075
3	midday	noon	3.703	4.000	4.000	4.000
4	furnace	stove	3.333	3.000	0.681	0.681
5	journey	voyage	3.296	3.857	2.688	2.984
6	tool	implement	3.185	3.857	1.672	3.442
7	coast	shore	3.111	3.857	3.208	3.857
8	animal	beast	3.111	4.000	1.017	4.000
9	magician	wizard	3.000	4.000	3.000	4.000
10	man	male	2.925	3.857	2.416	3.828
11	bird	cock	2.666	3.857	1.562	1.667
12	food	fruit	2.592	3.142	0.455	0.940
13	asylum	madhouse	2.481	3.857	3.750	3.750
14	location	land	2.407	3.714	1.231	2.369
15	bird	crane	2.296	3.571	1.562	1.667
16	creator	expert	2.074	3.714	0.878	1.584
17	lad	brother	1.925	3.428	0.878	0.937
18	oracle	church	1.888	3.571	2.812	2.999
19	monk	oracle	1.740	3.000	0.878	1.028
20	cell	organism	1.592	3.714	0.789	0.789
21	brother	monk	1.444	3.857	3.419	3.419
22	coast	hill	1.296	3.428	2.060	2.457
23	monk	slave	1.185	3.428	0.878	1.028
24	food	rooster	1.185	2.285	0.222	0.237
25	crane	implement	1.148	3.428	0.942	1.077
26	lad	wizard	0.851	3.428	0.878	0.937
27	chord	smile	0.851	2.571	1.039	1.108
28	coast	forest	0.814	3.142	0.455	0.542
29	journey	car	0.555	0.000	0.000	0.000
30	forest	beast	0.518	3.285	0.222	0.280
31	forest	graveyard	0.481	3.000	0.455	0.485
32	organism	tool	0.481	3.571	0.541	0.541
33	wizard	church	0.481	2.571	0.222	0.273
34	smile	graveyard	0.296	3.142	1.039	1.108
35	rooster	voyage	0.185	0.000	0.000	0.000
36	glass	magician	0.185	2.857	0.222	0.296
37	noon	string	0.185	0.000	0.000	0.000
38	smile	graveyard	0.185	0.000	0.000	0.000
39	fruit	creator	0.185	2.857	0.222	0.422

Table 3: Word Similarity by Items

STepLib: a SpatioTemporal Digital Library

Claudio de Souza Baptista and Zarine Kemp
Computing Laboratory, University of Kent at Canterbury
Canterbury, Kent CT2 7NF UK
Phone: +44 1227 764 000, Fax: +44 1227 762 811
E-mail: {cdsb1, zk}@ukc.ac.uk

Keywords: digital libraries, spatiotemporal systems, multimedia, databases, metadata

Edited by: Yanchun Zhang and Vladimir Fomichov
Received: August 15, 1999 **Revised:** November 8, 1999 **Accepted:** November 14, 1999

The advent of digital libraries has motivated research in some specific areas in order to apply innovative techniques to managing and retrieving information. This paper focuses on the design of a spatiotemporal digital library. It presents a historical evolution of digital libraries, discusses the main requirements and issues involved in spatiotemporal digital libraries and proposes a hierarchical metadata model based on four layers of abstraction.

1 Introduction

Traditional libraries have evolved from manual cataloguing, searching, and management of collections to computerised systems based on database technology and information retrieval. However, these library information systems, though very efficient and useful, are not sufficient to fulfill the new requirements of large-scale information dissemination in the digital era. Functionality for effective *retrieval* of *multimedia* data is limited. This limitation has been overcome with the advent of Digital Libraries (DL).

When documents in collections contain georeferenced information the DL needs to be extended in order to cope with specialised requirements. Handling information repositories where the space and time dimensions are crucial requires additional functionality for modelling, indexing, searching, retrieving and presentation of these data types.

In this paper, we discuss a model for a spatiotemporal digital library. Section 2 presents a brief survey of the evolution of digital libraries, section 3 discusses requirements and issues of spatial digital libraries, section 4 presents a spatiotemporal metadata model and a discussion of architecture and implementation issues. Section 5 concludes the paper with suggestions for future research directions.

2 Evolution of Digital Libraries

DL have evolved from the concepts associated with traditional paper based libraries. They include mechanisms that support electronic documents in different formats and media involving new issues and challenges. A brief history of this evolution is presented below to provide a context for the work on special-purpose libraries such as geolibraries.

The **first generation: Traditional Libraries** is characterised by clearly defined missions and roles of a library, and the services provided, but without making use of information systems. Resources include books, journals, magazines, games, maps, video and sounds. Services include loan, reservation, searching, and facilities to physically access the collections. There is a specific copyright legislation, in which ownership and authorship are very strong concepts. The resources do not change, for instance an individual book will never change its contents and authorship, although new editions of the same book may appear. It is also important to mention that, as the resources are physical, the notions of loan/reservation services are very important. That implies a one-to-many relationship between a user and resources.

The **second generation: Library plus information systems** involves the computerisation of the library system, which results in transforming the manual system to an electronic one, in which collections and resources are indexed and searched via special purpose software. The Online Public Access Library Catalogue, well known as OPAC, was widely adopted as the library system. Although, it is still used in libraries, OPAC demonstrates several limitations such as restricted display, poor user interface, and often provides a centralised solution implemented on expensive mainframes. Apart from that, OPAC provides information about user borrowing details, search based on different attributes such as title, author, subject, ISBN, classmark, boolean search including stop-lists (words that should not appear in the search), search based on the type of resource, such as book or periodical, and browsing based on attributes mentioned previously.

The **third generation: Digital Libraries** assumes the fact that now the library information systems not only provide index and search services but also retrieval as the resources move from a hard copy paper-based format to a mainly digital format. Like traditional libraries, DL are a

combination of information plus services to access them. DL involve actors, who interact with the system, and components which execute the different services provided by the DL. These actors can be categorised according to the role they play in the system: data providers, data consumers, and data managers or librarians. Data providers are responsible for organising the dataset in a way that makes it interesting for the other classes of users. A semantic description of the dataset is provided using metadata. These metadata include, but are not limited to, information about: the quality of the data, originator, price, formats available, where to get it, and when it is available. Data consumers are the DL end-users who utilise the DL services in order to discover a particular dataset that meets their requirements in a particular application domain. Finally, librarians are responsible for managing the resources. They determine subject classification, define policies and rules of utilisation, maintain a catalogue of users and data providers, and decide with which other DL they should intercommunicate.

The **fourth generation: Multimedia DL** introduces the retrieval of resources of different media such as video, audio, maps, images and hypertext documents. Previously, the search and indexing were restricted to alphanumeric data types. In the context of textual resources this is acceptable and efficient, but is not true for multimedia data types where interpretation of their semantics is required for effective indexing and searching. Furthermore, there are specific domains such as spatial and temporal applications which require tailored searching, browsing and indexing mechanisms. This generation is still evolving; while it is feasible to think in terms of a general digital library that can deal with all the complexities of those different data types, it is likely that specific type-dependent data repositories will emerge such as video DL, image DL, georeferenced DL, and textual DL which will be required to interoperate.

3 Spatial Digital Libraries

Concepts similar to those of traditional libraries, may be applied to physical map libraries. They contain spatial information which is static in the sense that users cannot change resolution, zoom and pan through space or generate new data. The same issues previously discussed for traditional libraries, apply to libraries of paper maps.

However, the use of Spatial Digital Libraries (SDL), which assume by definition that all data are georeferenced, opens up endless possibilities for user interaction. SDL are part of the fourth generation. They have the functionality of a DL tailored to the spatiotemporal application domain. As a fourth generation DL it is assumed that SDL will cope with georeferenced multimedia data. The library holdings may be maps in a digital format as well as other georeferenced data including photographs, satellite images, aerial photographs, textual documents, video and audio [1].

We can divide the users of a SDL into groups of data providers and data consumers. Data providers are responsible for delivering georeferenced data. The data must comply with the metadata model requirements in order to be able to be inserted into the underlying database and be searched by data consumers [2]. There are several standards proposed for spatiotemporal metadata including FGDC/CSDSM [3] and ISO/TC 211 [4]. However, none of them include the full semantics needed to cope with multimedia data, hence an extended metadata model is necessary. It has been required that data providers can access the system remotely and that the billing system for the use of the data enables them to receive payment electronically, so cost models should be investigated [5]. Moreover, the system should guarantee copyright rules, so that data providers can trust the SDL, to enforce secure access to the data [6].

Data consumers are explorers of the SDL. They have differing knowledge background and perspectives when accessing geo-information and range from naive to expert. Naive users do not care about details of the geo-information such as originator and accuracy. They expect a user friendly interface which requires minimum computer skills and no knowledge of data dependent formats. On the other hand, expert users know exactly what they are looking for and how to specify precisely the spatial, temporal and thematic dimensions. They also know about the metadata schema which enables them to retrieve detailed information from the SDL using a query mechanism. In addition, both categories of user also expect to connect to the SDL remotely. The physical distribution of the spatial data should be transparent in all cases.

3.1 Requirements for a SDL

The services required of a SDL arise, on one hand, from the specialised, complex digital representation of spatiotemporal datasets and on the other hand, the cognitive requirements of users when dealing with space and time. The characteristics of user-SDL interaction can be categorised as follows:

- Requirements for specifying queries in the three main dimensions, spatial, temporal and thematic, or any combination thereof. In essence, retrieval from SDL can be described as queries which focus on 'what is in this space?' and 'where does a particular phenomenon occur?'. The same broad categories apply to the temporal dimension but due to limits on the length of the paper the discussion and illustrative examples focus primarily on the spatial dimension.

- It should be noted that humans use a range of mechanisms to represent and refer to space e.g. nominal, coordinate, topological, absolute/relative reference which have to be accommodated in the user interface to a SDL.

- Queries may not always be capable of one-off specification. The user may need to refine a query in successive stages so an interactive interface in browse mode may be required. In fact, this style of interface may be necessary for performance reasons in SDL, to successively reduce the problem space where large volumes of data may

be involved.

- Output and presentation of data from SDL can be problematic. Retrievals often require results to be presented in cartographic form with some user input on the fly to control the look-and-feel of the display.

- Spatial data manipulation requires very specific spatial operators either for explicit user queries or to hide the complexities of the underlying data representations from the user in a transparent manner. As noted earlier, georeferenced data may be stored in multiple formats and multiple scales. Transformations to handle these should be handled by the underlying SDL.

3.2 User interaction

The issues highlighted above inform the design of the user interface of the prototype SDL being developed. It provides the following services embedded in a graphical user interface:

- *Basemap*: display the entire world using either a virtual globe representing the earth in which users can rotate and zoom in, or a base map of the earth in which users can pan and zoom in to determine the spatial region of interest. The base map can be a projection of the earth on a two dimensional map or a digital topographic map and/or an image map which has satellite data covering the entire Earth surface in different resolutions in order to enable zooming. These base map representations are not exclusive as they provide different perspectives of the same data (e.g. the earth's surface) and they are of interest to different user groups.

- *Gazetteer*: this is a specific SDL function. It contains a set of attribute pair values consisting of a placename and its corresponding geographical coordinates in the georeferenced system utilised such as latitude/longitude. This is very useful to enable users to specify the bounding boxes of the locations they are interested in, by using placenames. For instance, a user interested in some information related to the Amazon forest in Brazil can browse through the gazetteer to find the placename Amazon. Then the SDL automatically sets the spatial dimension with the spatial coordinates obtained from the gazetteer entry. Table 1 shows an example of a gazetteer which uses latitude/longitude as a reference system.

Placename	North	East	South	West
World	90.0	180.0	-90.0	-180.0
Europe	73.0	43.0	33.0	-25.0
United Kingdom	59.5	2.0	49.8	-8.3
England	56.0	2.0	49.8	-6.0

Table 1: An instance of a gazetteer

- *Ontology*: ontologies can be used to express a complex structure of concepts and their relationships in a SDL. It comprises a vocabulary specific to a particular application domain. The relationship between concepts can be expressed via synonyms, thesauri and multi-lingual dictionaries, so that users can specify concepts in a language very close to natural language [7]. Table 2 shows an example of such an ontology based on a land cover classification. Ontologies can also be used to map between different structures embodying similar concepts (e.g. between the land cover classifications used in the EU and in the USA).

Land cover classification
1. Urban or built-up
2. Agricultural land
3. Forest land
4. Water
5. Snow

Table 2: An example of ontology

- *Spatial functions*: Spatial functions in a SDL enable users to express their spatial queries. Spatial operators can be classified into topological, directional, and metric; the topological ones include intersection, overlapping, inside, touches, crosses between regions; the directional ones include left, north, below; the metric ones include appropriate numeric representations for concepts such as far and near.

- *Temporal functions*: temporal operators are equivalent to the ones in the spatial dimension and include, for example, temporal topological operators such as: before, meet, during, after, touches and overlaps of two temporal intervals.

- *Datatype specific functions*: these are functions defined according to a specific data type. For text, search can be based on keywords, using stop-lists and synonyms. Exact or fuzzy match of phrases and words can be also utilised. For images, content-based retrieval (using low-level features of an image: texture, shape and colour) is more appropriate [8]. Search facilities for images are therefore based on a combination of feature vectors and textual keywords. For video, queries based on keyframes or using content-based retrieval similar to that for images is also appropriate. In the case of video, there is the temporal variable which is inherent to this data type, as a video can be viewed as a set of images through time. In both images and video there is the possibility to extract subobjects contained in the images and the spatial relationships between them.

- *Display of data*: complex multimedia data requires a different display function for each data type available beside the metadata. For instance, maps should be displayed in vector form which includes point, polygons, lines; images are generally displayed as a raster; text in a text format and video and audio in the appropriate formats.

- *Zooming*: this is a very useful function which allows users to get a more detailed resolution of the underlying dataset (zoom in) or a more coarse resolution (zoom out), thus providing navigation through space in different resolutions (referred to as vertical navigation).

- *Pan*: this is the ability to scroll in all directions of the data in cases where the entire dataset cannot be displayed on the screen device. This function is also used for navigation through space in the same resolution (horizontal navigation).

4 The STepLib SDL

Following the requirements discussed in the previous sections a SDL metadata model was designed using different levels of abstraction. Metadata is generally described as data about data [9] and it has been used with success in the library community for describing information about the resources a library holds. Three main issues arise with respect to incorporating metadata in spatial and multimedia databases: the structure of the metadata, capture or creation of relevant metadata and storage of the metadata objects. In this multimedia database model, metadata plays a central role: it is used to help the search process (data discovery) and interpretation of the underlying multimedia datasets. Figure 1 presents the STepLib spatiotemporal metadata model. It is designed as a hierarchy of abstraction which is divided into four layers: at the top level is the *Knowledge Object Layer*, at the second level is the *Spatiotemporal Object Layer*, at the third level is the *Datatype Specific Object Layer* and at the bottom layer is the *Raw Data Layer*.

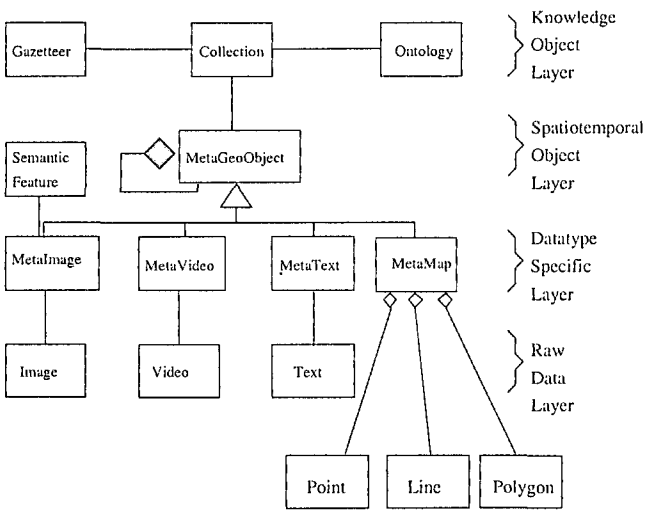


Figure 1: STepLib metadata model.

The *Knowledge object layer* comprises domain-dependent metadata. It contains the Collection class, which is a concept inherited from the library community, a Gazetteer, which consists of pairs of location names and spatial coordinates (e.g. latitude/longitude), and an Ontology class as described in section 3. A Collection can be viewed as a cluster of images, text documents and videos which relates to the same knowledge domain. Hence, users can, at a high level of abstraction, browse through existing collections in order to narrow the search and increase precision without losing recall. The classification of data sets into collections is determined by the data provider. Collection is an aggregation of MetaGeoObjects. Attributes in a collection include title, descriptive annotation, originator, date of creation, duration and overall spatial and temporal footprints of the underlying data sets.

The *Spatiotemporal object layer* consists of MetaGeoObjects. This class holds the spatial and temporal dimensions and contains attributes and methods that are applied to all subobjects that are inherited from it: MetaVideo, MetaImage, MetaText and MetaMap. The attributes include spatial and temporal footprints, name, collector and annotation. The MetaGeoObject class implements a *partOf* relationship which enables nested metadata to be described and composite objects modelled.

The *Datatype specific object layer* comprises the datatype specific classes which inherit the attributes and operations from the MetaGeoObject class and implement their own ones. Each respective subclass has metadata which is data dependent. The *MetaImage* subclass contains format, image resolution, a thumbnail representation of the image, which allows progressive retrieval, image size and lineage, which is used for quality control in satellite images and sensors. The *MetaText* subclass contains attributes such as length, format and keywords. The *MetaVideo* subclass contains duration, category, format and frames/sec. *MetaMap* is an aggregation of vector data types such as points, lines and polygons. This vector data is represented by its geometry and a theme (e.g. river, hotel).

The bottom level, known as the *Raw data layer* contains the different multimedia data types with their respective data. At that level large objects such as image, textual documents, points, lines, polygons and video are stored and maintained. Apart from representing the data in the database, it is also possible to access data external to the database over the Internet. In this case the data is represented by its respective URL which is used by the system to access the data on the fly and present it to the user. In such situations it is not possible to maintain the integrity of the data. If the content of a URL changes there is a consistency problem to be resolved.

Indexing these complex and heterogeneous datasets is also an important issue. From our basic assumption that the underlying data is geo-referenced it is imperative to use spatial indexing that enables efficient access to it. STepLib uses R-tree for indexing the spatial bounding boxes. Images are indexed using the low-level features such as colour, shape and texture. Colour is used for photographs, texture for satellite and aerial images, and shape for hand-drawn images. Other metadata attributes are indexed using the B-tree technique. Textual documents use a proprietary indexing mechanism which enables keyword searching.

A prototype has been developed using a client-server architecture. Clients are Java applets that are responsible for user interaction, display of data and communication with the database system. This communication is realised via JDBC protocol, 2.0 API specification which enables the use of SQL99. The server holds the database system which is responsible for the transaction management, spatial and multimedia indexing.

Examples of queries STepLib is able to deal with are:
1) Retrieve information about the Amazon forest between 1981 and 1990.
2) Retrieve pubs that are within 50 metres of the River

Thames in London.
3) Retrieve all reports about this area?
4) Retrieve fish larvae photographs that look like this one.
5) Find a sequence of satellite images showing how the Amazon area in Brazil has been deforested over the last 10 years along with maps of Indian reserves.

5 Conclusion and Future Work

This paper has concentrated on the evolution of digital libraries and presented a metadata model which encompasses multimedia objects with spatiotemporal footprints. The model is designed to support different levels of abstraction so that users can query the underlying datasets based on thematic knowledge, spatial and temporal dimensions, data types (image, text, video), content-based retrieval for images and information retrieval techniques for textual documents. Future work on this model involves the provision of interoperability using the XML standard and a formalization of the user interaction scenarios. Also, content-based retrieval for video will be developed.

6 Acknowledgements

The first author would like to thank the CAPES-Brazil for partially funding this research.

References

[1] Goodchild, M. (1998) The geolibrary. Innovations in GIS 5 (Carver, S. ed.). Taylor & Francis.

[2] Beard, K., Smith, T., and Hill, L. (1997) Meta-Information Models for Georeferenced Digital Libraries Collections. *Proceedings of the 2nd IEEE Metadata Conference*. Maryland, USA.

[3] Federal Geographic Data Committee (1995) Content Standards for Digital Spatial Metadata,*Workbook Version 1.0*. National Spatial Data Infrastructure, USA.

[4] International Standard Organization (1998) ISO/TC 211 Geographic information / Geomatics Part 15: Metadata. International Standard Organization.

[5] Sistla, A., Wolfson, O., Yesha, Y., and Sloan, R. (1998) Towards a theory of cost management for digital libraries and electronic commerce. *ACM Transactions on Database Systems*, 23, 4, p. 411-452.

[6] Onsrud, H. and Lopez, X. (1998) Intellectual property rights in disseminating digital geographic data, products and services: conflicts and commonalities among EU and US approaches. *European Geographic Information Infrastructures opportunities and pitfalls, GISDATA 5*, (Burrough, P. and Masser, I. eds.). Taylor & Francis.

[7] Mena, E., Kashyap, V., Illarramendi, A. and Sheth, A. (1998) Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. *Proceedings of the International Conference on Formal Ontology in Information Systems*, Trento, Italy.

[8] Yoshitaka, A., and Ichikawa, T. (1999) A Survey on Content-Based Retrieval for Multimedia Databases, *IEEE Transactions on Knowledge and Data Engineering*, 11, 1, p. 81-93.

[9] Sheth, A. and Klas, W. (1998) Multimedia Data Management - using metadata to integrate and apply digital media. McGraw Hill.

Is Consciousness not a Computational Property? — Response to Caplain

Damjan Bojadžiev
Department of Intelligent Systems, Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: damjan.bojadziev@ijs.si, WWW: http://nl.ijs.si/~damjan/me.html

Keywords: knowledge, reflexivity, consciousness, computation, automata

Edited by: Rudi Murn
Received: October 15, 1998 **Revised:** February 12, 1999 **Accepted:** December 2, 1999

Caplain's argument that a conscious automaton would violate a certain principle of cognition is inconclusive. Its central part has the non-demonstrative form: X is sufficient for Y because Z is not and nothing else could be. The argument and the principle are also not specific to automata.

1 Introduction

Caplain has recently argued — first in a special issue of this journal and later in book form — that 'consciousness cannot be adequately described as a computational structure and (or) process' because a conscious automaton would violate a certain general principle of cognition (Caplain, 1997, p. 190). The argument seems original, interesting and quite intricate, to the point of becoming slippery in its decisive steps. Since it also runs against my own views (Bojadžiev, 1997), I was strongly motivated to analyze it (and reanalyze, and ...) in order to find its weak or missing link(s). I point out these links in section 3 and analyze the principle on which the argument is based in section 4.

2 Consciousness and knowledge

Caplain sets up his argument against automatic consciousness by recalling the distinction between knowledge and belief — knowledge as justified true belief — and connecting human consciousness with the capacity for knowledge (p. 190–2). Even this initial step already appears puzzling, since it is not obvious that the capacity for belief is any less characteristic of human consciousness. The connection seems even more puzzling when Caplain qualifies the knowledge in question as generally partial, approximate and subject to improvement (p. 190). The connection becomes clear only when Caplain moves to a different kind of knowledge and connects consciousness in general with the capacity for *self*-knowledge:

> Any conscious being, whose consciousness is active at some moment, is able to know something for sure at that moment: the fact that "there is conscious impression there" (p. 191).

Caplain then introduces apparently yet another kind of knowledge, namely

> truths which are "basic", or "primordial", in the sense that we rightfully consider them as *self-evident*, without having any clear idea of how we got to know them. Examples of such statements are: our own existence, the real existence of the world external to ourselves, the ability of our senses to provide us with some reflection of that external world (p. 191).

This kind of knowledge is supposed to illustrate what Caplain calls the reflexivity of consciousness, which is the key concept in his main argument. Caplain says that the capacity for knowledge entails the capacity for self-checking, which he calls reflexivity (p. 191). He does not spell out more exactly what reflexivity or self-checking is, so that it remains unclear how the kind of knowledge he cites illustrates this concept. Going by ordinary meaning, this third kind of knowledge is better described as consisting in *evident* (rather than *self*-evident) truths, and reflexivity is better illustrated by the previous kind of knowledge (active consciousness knowing itself). But Caplain quotes Putnam's brain in a vat scenario (Putnam, 1981), though not his argument, and adds that 'unless we fall into absolute skepticism, we are compelled to admit that strange property of reflexivity' (p. 191). This could be taken as an oblique reference to the reflexive, "counter-performative" nature of Putnam's argument. Put simply, Putnam's argument is that entertaining the notion that we are brains in a vat refutes it, a negative twist on Descartes' dictum: I think (in a vat), therefore I'm not (in it). Similarly, going again by ordinary meaning, self-checking or reflexivity might literally be the tendency of knowledge to somehow check itself, keep itself in check by preferring what is apparent (the "basic" truths above) to what may be conceivable (the pickled brain hypothesis).

3 The argument against conscious automata

Caplain's argument revolves — the verb is carefully chosen — around the question how could an automaton check or

verify its knowledge. He argues that a conscious automaton could verify that all it knows is true — or, more precisely, establish the truth of the statement that all it knows is true — merely on the basis of containing the formula of that knowledge i.e. the statement expressing that all it knows is true. This would contradict what Caplain calls the cognitive separation principle, which means that the premise of a conscious automaton has to be rejected. However, in this clash of principle and particular case, it is doubtful that Caplain actually establishes the particular case. Since this is the central part of the argument, with words and clauses in it under considerable inferential stress, an extended quotation is appropriate. Caplain uses the following notation: E is the hypothetical conscious automaton, $\Sigma(E, C)$ is the set of informations of which E is certain, recorded in it through some method C, and $T(E, C)$ is the statement expressing that all E knows through C is true, i.e. that all informations in $\Sigma(E, C)$ are true (p. 192). This statement expresses the 'true' part of the definition of the automaton's knowledge i.e. justified true beliefs, and it is itself included in $\Sigma(E, C)$, which Caplain refers to as condition 2:

$$T(E, C) \in \Sigma(E, C) \tag{2}$$

This condition now expresses the 'true' part of the definition of the automaton's knowledge for the automaton itself, and the argument now centers on the way in which the automaton verifies its knowledge, the 'justified' part:

> The realization of condition 2 would be sufficient to validly warrant to E that $T(E, C)$ is true. This is why. We could imagine, for a moment, that E makes sure of $T(E, C)$ [...] by another means: by [...] But in this case, it would be necessary that [...] In other words, *to infer* $T(E, C)$, *it would be necessary to already know* $T(E, C)$*!* Finally, the realization of condition 2 would be the only means for the automaton to get such a guarantee. Hence our condition 3: *The realization of condition 2 i.e. the recording of* $T(E, C)$ *through C, is sufficient for guaranteeing to E that* $T(E, C)$ *is true* (p. 193–4).

The reasoning here is implicit enough to invite the impression that it is merely a roundabout way of restating, rather than proving, the sufficiency of (2), the detour being what looks like an argument for its *necessity*. A clearer way of putting the argument would be this: an automaton must, as such, have sufficient reason for knowing that all it knows through C is true, and its only way of knowing is (again) through C; in particular, an alternative way which may come to mind is not available, because it would be circular. So, if the automaton knows that $T(E, C)$ is true, the sufficient reason for this knowledge can only be the recording of $T(E, C)$ through C. But what this comes down to is that $T(E, C)$ can only be known in the same way as any other $S \in \Sigma(E, C)$, namely by being recorded through C, and that was already sufficiently clear beforehand.

The major weakness of the argument quoted above is its non-demonstrative, eliminative form: X is sufficient for Y because Z is not, and nothing else could be. A stronger, positive argument would show directly that (2) *is* sufficient by showing *how* it is sufficient. Such a demonstration might take into account the special, partly self-referential character of $T(E, C)$: it says that all of $\Sigma(E, C)$ are true, and is itself included in $\Sigma(E, C)$; so, $T(E, C)$ includes itself in its statement of what is true. The inclusion of $T(E, C)$ in $\Sigma(E, C)$ could be compared to saying 'I can speak', thereby establishing the truth of what I'm saying. This self-affirming character of $T(E, C)$ might even provide a better illustration of what Caplain calls self-checking or reflexivity than the ones he offers.

Another weakness of Caplain's argument is its content: the argument is supposed to be about automata, but it does not rely in any way on their defining concepts. No mention or use is made of characteristic restrictions on structure and function, e.g. the fixed number of internal states or state transitions. Thus, the argument is not specific to (finite) automata, and it is hard to see why it would not go through for any kind of being which records information, including humans, though it would not be any more persuasive for them.

A similar point can be noted by returning to the top level of Caplain's argument. Its punchline is that condition 3, which says that (2) is sufficient guarantee for $T(E, C)$, contradicts a certain cognitive principle. Since there is much doubt as to how firmly Caplain actually establishes condition 3, the outcome could also be that a conscious automaton can know, and know that what it knows is true, without sufficient guarantee, "any clear idea of how it got to know it" (the "basic" or "primordial" knowledge above). Thus, automata might be in much the same situation as humans with respect to guarantees of knowledge, tentatively settling for evident or simplest explanations and revising them as they go along, if they must.

4 The principle of cognitive separation

Caplain formulates what he calls the cognitive separation principle for an automaton A and a conscious being E observing A. If I is the method of recording information in A and $\Sigma(A, I)$ and $T(A, I)$ are defined as above, the principle says:

> The inclusion of $T(A, I)$ in $\Sigma(A, I)$ cannot be sufficient to validly guarantee to E that $T(A, I)$ is indeed true, i.e. that all informations in $\Sigma(A, I)$ are true (p. 193).

In his main argument, Caplain uses only the special case in which $I = C$ and $E = A$. But the principle itself is easier to formulate than this special case and it also seems important in itself. In explaining the principle, Caplain says that

> it expresses that any kind of information recording in an automaton cannot contain in itself a sufficient validation of these informations (p. 193).

This formulation invites the comment that it expresses the principle better than the statement of the principle itself, with all its attendant definitions. Indeed, the whole argument with its painful details also appears superfluous, obviously decided in advance by the stipulation of the principle for automata only. Caplain does not go so far as to formulate a principle of cognitive *non*-separation for conscious beings

> The inclusion of $T(E, I)$ in $\Sigma(E, I)$ can be sufficient to validly guarantee to E that $T(E, I)$ is indeed true, i.e. that all informations in $\Sigma(E, I)$ are true

or to say that "some kind of information recording in a conscious being contains in itself a sufficient validation of these informations", but he says that 'the cognitive separation between a field of reality and recorded informations supposed to describe it does not extend to consciousness' (p. 194). By itself, this could mean either that there is no cognitive separation if the information is recorded by a conscious being, whatever the field of reality, or it could mean that there is no separation if the field is consciousness itself. Since Caplain adds that 'a conscious being builds its knowledge of reality only from conscious impressions' (p. 194), he apparently means the former, but the problem is that only the latter clearly supports his claim. That is, there is clearly no cognitive separation, or indeed much difference, between (the content(s) of) consciousness and our informations about it. But in less self-referential cases it is less obvious that cognitive separation is absent, and why it should or might be.

On the other hand, cognitive separation in humans or automata can be reduced or eliminated to the extent that the process of recording information is self-referential, providing information either about the entity in which it functions or about itself. These kinds of information amend what Caplain says in support of the principle of cognitive separation, namely that verifying that the recording process information requires 'an observation of A, I *and* the domain of reality being considered' (p. 193). If the domain is A itself, so that the automaton records information about itself, observation of A and I is sufficient for verifying these informations, but the principle of cognitive separation remains in force: it is not enough to consider what I says about A, even if I says that it is. Similarly, even if someone only talks about himself, it is no guarantee that he tells the truth if he says that he does.

At the next level of self-involvement, the recording process could turn upon itself, though this would not in itself guarantee that it provides only true informations about itself. But checking whether I provides true informations about itself would then require only an observation of I itself. Furthermore, it seems possible to construct an I which would only provide true (though possibly not complete) informations about itself, "a kind of information recording in an automaton that can contain in itself a sufficient validation of these informations". This recalls the second kind of knowledge Caplain mentions, active consciousness registering its own effects:

> having some conscious sensation at some moment entails the knowledge that, at least, there is that conscious sensation (p. 191).

This kind of self-referential knowledge would correspond to a process of automatic self-observation registering its own effects, similar to what Perlis calls self-noting (Perlis, 1997, p. 518); put this way, this kind of self- knowledge may not be that far out of automatic reach (Webb, 1980).

5 Conclusion

Caplain does not prove that consciousness is not a computational property. I do not prove that it is, much less show *how* it could be, but I indicate *why* it might be: by agreeing with Caplain's initial observation about consciousness knowing itself and noting that self-reference is something which formal systems are very good at.

Acknowledgement

I am grateful to the first referee for his detailed comments which prompted me to express myself more precisely, leave out some points which appeared less important, incorrect or in doubt, and make clearer my own position.

References

[1] Bojadžiev, Damjan (1997), Mind versus Gödel, in Gams et al (1997), pp. 202–10; HTML at http://nl.ijs.si/~damjan/g-m-c.html

[2] Caplain, Gilbert (1997), Is Consciousness a Computational Property?, in Gams et al (1997), pp. 190–4

[3] Gams, Matjaz, Paprzycki, Marcin, Wu, Xindong (ed.), (1997), *Mind Versus Computer* (Amsterdam: IOS Press)

[4] Perlis, Donald (1997), Consciousness as Self-Function, Journal of Consciousness Studies, 4 (5–6), pp. 509–25

[5] Putnam, Hilary (1981), *Reason, Truth and History* (Cambridge: Cambridge University Press)

[6] Webb, Judson (1980), *Mechanism, Mentalism and Metamathematics - An Essay on Finitism* (Dordrecht: D. Reidel Publishing Company)

Is Consciousness not a Computational Property? — Reply to Bojadžiev

Gilbert Caplain
ENPC-CERMICS 6 & 8 avenue Blaise Pascal. Cité Descartes – Champs-sur-Marne
F-77455 Marne-la-Vallée Cedex2 , FRANCE
E-mail: caplain@cermics.enpc.fr

Keywords: consciousness, knowledge, belief, artificial intelligence

Edited by: Rudi Murn
Received: October 17, 1999 **Revised:** November 12, 1999 **Accepted:** December 4, 1999

In [2], I have argued that consciousness cannot be adequately described as a computational structure and/or process. This argument is challenged by Damjan Bojadžiev [1] (this issue). This paper contains my replies and comments.

1 Introduction

In [2], I have argued that consciousness cannot be adequately described as a computational structure and/or process. The proof makes use of a well-known, but paradoxical, ability of consciousness to reach *ascertained knowledge*, as opposed to *mere belief*, in some cases. My argument is challenged by Damjan Bojadžiev [1] (this issue). I will bring a few comments and replies. For clarity, I will follow the same order (and the same sectioning) as Bojadžiev's paper, and take some quotes from it.

For the sake of brevity, I will focus only on the parts of Bojadžiev's developments which seemed the most interesting and/or objectionable to me, and leave aside a few points which would maybe have deserved a comment. Moreover, as well in this reply as in my original paper, the argument is merely outlined: some notions involved here call for further developments.

2 Consciousness and knowledge

Bojadžiev mentions that I recall the distinction between knowledge and belief, and that I connect

> human consciousness with the capacity for knowledge (p. 190–2). Even this initial step already appears puzzling, since it is not obvious that the capacity for belief is any less characteristic of human consciousness.

I do not imply that capacity for belief is any less characteristic of human consciousness than capacity for knowledge. At that point, I just mention a familiar property of human consciousness to reach *ascertained knowledge*, as opposed to *mere belief*, in some cases; but both knowledge and belief are parts of the human conscious experience. (In what follows, consistently with [2], the word "knowledge" will be used in its strong sense of "ascertained knowledge".) Then, however, I mention an observation pertaining to *any kind* of consciousness, human or not – or, more accurately, an *element of definition* of what we will term as *consciousness* in general, not merely human consciousness: *any conscious being, whose consciousness is active at some moment, is able to know something for sure at that moment: the fact that "there is conscious impression there"* [2, p.191]. Some *minimal* ability for knowledge is thereby attached to *any* consciousness, human or not. It is this connection between consciousness and knowledge – between *any* consciousness and *at least some minimal* knowledge – which will reveal interesting afterwards, allowing us to derive that an automaton cannot be conscious, just by deriving that a conscious automaton would not be able of knowledge. (Notice, by the way, that we do not know whether a *capacity for mere belief* – belief that is not knowledge – is common to any kind of consciousness...)

3 The argument against conscious automata

Both Bojadžiev and I use the following notations here: E is the hypothetical conscious automaton, $\Sigma(E, C)$ is the set of informations of which E is certain, recorded in it through some method C, and $T(E, C)$ is the statement expressing that all informations in $\Sigma(E, C)$ are true [2, p.192–3].

Bojadžiev states that the condition I refer to as condition 2:

$$T(E, C) \in \Sigma(E, C)$$

> now expresses the 'true' part of the definition of the automaton's knowledge for the automaton itself, and the argument now centers on the way in which the automaton verifies its knowledge, the 'justified' part:

Later, Bojadžiev translates and comments some part of my argument as follows:

> So, if the automaton knows that $T(E,C)$ is true, the sufficient reason for this knowledge can only be the recording of $T(E,C)$ through C. But what this comes down to is that $T(E,C)$ can only be known in the same way as any other $S \in \Sigma(E,C)$, namely by being recorded through C, and that was already sufficiently clear beforehand.

The interpretation of Conditions 2 and 3 which underlies these two quotes is not correct – but, admittedly, I did not make the point sufficiently clear in my paper, and this indeed requires a clarification. In order for a conscious automaton E to be possible, there has to be some recording method C which will *finally* represent E's *certainty label* – this is Condition 1 –, but which is not *presupposed beforehand* to be so. Then, I derive that Conditions 2 and 3 are necessary so that C could possibly, indeed, represent E's certainty label. Condition 2 merely states that $T(E,C)$ is included in $\Sigma(E,C)$: at this point of the reasoning, this inclusion does not entail that $T(E,C)$ is true. Condition 3 then states that this recording of $T(E,C)$ through C would be sufficient to warrant to E that $T(E,C)$ is true – because any alternative way for E to know that would be circular. Now, under Condition 3, and contrarily to what Bojadžiev states, the way $T(E,C)$ comes to be known by E is *very different* from the way any other $S \in \Sigma(E,C)$ then comes to be known by E. *Once $T(E,C)$ is justified as E's knowledge and C is thereby confirmed to be E's certainty label,* the recording of any other S through C translates the fact that E is certain of S. This just does not work if, instead of S, we consider $T(E,C)$ itself ! The very paradoxical character of Condition 3 precisely lies in the strange way $T(E,C)$ would come to be known by E.

> The inclusion of $T(E,C)$ in $\Sigma(E,C)$ could be compared to saying 'I can speak', thereby establishing the truth of what I'm saying. This self-affirming character of $T(E,C)$ might even provide a better illustration of what Caplain calls self-checking or reflexivity than the ones he offers.

As a matter of fact, supposing for a moment the possibility of a conscious automaton E, the inclusion of $T(E,C)$ in $\Sigma(E,C)$ exactly corresponds to a statement of E's reflexivity – E's certainty that anything E is certain of is true. (The problem is that this inclusion cannot be sufficient to *warrant to* E this statement of reflexivity.)

As for the comparison with saying "I can speak", we must notice something important here. Suppose that we observe some device (an automaton, a conscious being...) showing us its speaking ability by uttering the words "I can speak". This will prove *to us* that this device is able to speak (independently, by the way, of the words uttered – replacing "I can speak" by "To be or not to be" in this experiment will not change this conclusion of ours). But nothing warrants us that this device proves anything *to itself* by so doing, nor that this utterance indeed expresses any knowledge of this device about itself and its own abilities. Especially, in this example, *we* know that this device states something true by uttering the words "I can speak", but at this point, we do not necessarily know (lacking sufficient information) whether *it* knows that it can speak.

Later, Bojadžiev states:

> the outcome could also be that a conscious automaton can know, and know that what it knows is true, without sufficient guarantee, "any clear idea of how it got to know it" (the "basic" or "primordial" knowledge above). Thus, automata might be in much the same situation as humans with respect to guarantees of knowledge, tentatively settling for evident or simplest explanations and revising them as they go along, if they must.

Here, there seems to be a misconception about the notion of *knowledge* which is involved. When I refer to "truths which are 'basic' or 'primordial' in the sense that we rightfully consider them as *self-evident*, without having any clear idea of how we got to know them"[2, p.191], this does not entail the notion that "we have not sufficient guarantee", nor that we "tentatively settle for simplest explanations and revise them as we go along". As a matter of fact, *we have quite sufficient guarantee* of self-evident truths – there is nothing *tentative* about admitting them. (Consider, just as an example mentioned in [2, p.191], our knowledge of the real existence of the world external to ourselves.) We have to admit this strange ability for knowledge, unless we stand for *absolute skepticism* – the denial of the possibility of knowledge – a view within which, as I mentioned in [2, p.191], it is impossible to prove anything anyway... In the quote above, it does not seem clear whether Bojadžiev advocates absolute skepticism.

4 The principle of cognitive separation

Bojadžiev makes a few remarks about the special case when the information recorded by a conscious being refers to consciousness itself (self-reference). Let us examine two quotes:

> there is clearly no cognitive separation, or indeed much difference, between (the content(s) of) consciousness and our informations about it.

and later, considering some recording method I:

> But checking whether I provides true informations about itself would then require only an observation of I itself. Furthermore, it seems possible to construct an I which would only provide

> true (though possibly not complete) informations about itself, "a kind of information recording in an automaton that can contain in itself a sufficient validation of these informations". This recalls the second kind of knowledge Caplain mentions, active consciousness registering its own effects: (...)

Again, there is a misconception here – and again, this is a difficult point that I did not emphasize enough in my paper. The difference between *true* statements recorded in a being and *knowledge* of these statements by this being, is no less essential in the special case when these statements refer to the being itself and/or to the recording method I itself. In other words, reflexivity is no less paradoxical when restricted to self-referential cases. In the second quote above, it is certainly possible to construct an I providing only true informations about itself – a *truthful self-referential* recording method I –; this *does not entail, by any means,* that these informations "contain a sufficient validation of themselves"! Similarly, regarding the first quote above, *there is a difference* between the contents of consciousness and *the character of certainty*, for the consciousness itself, of the existence and aspects of these contents of consciousness. Indeed, as I mentioned, consciousness is able to know for sure that "there are these conscious contents", but this is *a noticeable property* of consciousness, which should be accounted for, and which is not just implied by the self-referential aspect of this certainty.

5 Conclusion

In this paper, I attempted to clarify some aspects of my previous paper [2] which seemed insufficiently explained back then, in light of Damjan Bojadžiev's response [1].

Since some notions involved are both intricate and unfamiliar, such exchanges and discussions will hopefully contribute to the future advancement of scientific knowledge about these notions.

References

[1] D. Bojadžiev. Is Consciousness not a Computational Property ? – Response to Caplain. *Informatica : This issue.*

[2] G. Caplain. Is consciousness a computational property ? In : Mind versus Computer. M. Gams & M. Paprzycki & X. Wu. IOS Press, 1997. (Previously published in : *Informatica*, (19):615 – 619, 1995.)

Characterization Results for the Poset Based Representation of Topological Relations – II: Intersection and Union†

Luca Forlizzi
Dipartimento di Matematica Pura ed Applicata, Univ. of L'Aquila,
Via Vetoio, Coppito, I-67010 L'Aquila, Italia.
E-mail: {forlizzi,nardelli}@univaq.it
AND
Enrico Nardelli
Istituto di Analisi dei Sistemi ed Informatica,
Consiglio Nazionale delle Ricerche, Viale Manzoni 30, I-00185 Roma, Italia.

Keywords: topological model, poset, spatial relations, lattice completion.

Edited by: Rudi Murn
Received: May 16, 1999 **Revised:** January 11, 2000 **Accepted:** February 22, 2000

Formal methods based on the mathematical theory of partially ordered sets (i.e., posets) have been used for the description of topological relations among spatial objects since many years.
In particular, the use of the lattice completion (or normal completion) of a poset modeling a set of spatial objects has been shown by Kainz, Egenhofer and Greasley to be a fundamental technique to build meaningful representations for topological relations.
In a companion paper [3] we have discussed the expressive power of the lattice completion as a formal model for a set of spatial objects. In this paper we prove sufficient and necessary conditions for its use to give a correct representation of intersection and union relations among spatial objects.
We also show how to use lattice completion when working on a subset (i.e., a view) of the set of spatial objects so that the computation only considers objects relevant to the view itself.

1 Introduction

In a companion paper[1] [3] we have reviewed the poset-based representation for the topological data model, examined problems arising from a naive extension to the most general case, and shortly described how we tackle them. We have introduced definitions related to posets and lattices and some basic facts about them. Finally, we have introduced formally the definition of closure of a class S of objects with a set-containment relation with respect to a certain set operator, of representation and of universal partition. In this paper we focus on the study of the representation of the closure of a class with respect to set-intersection (Sect. 2) and set-union (Sect. 3) operators. This paper also contains conclusions and final remarks.

2 Representation of Set-Intersection Closure

Most of the proofs of the results in this section are either almost straightforward or rather technical. Hence they have been omitted for clarity. They are reported for completeness in Appendix A (p.90).

[1]Research partially supported by the European Union TMR project "Chorochronos"

2.1 Sufficient Conditions for Representation of Set-Intersection Closure

The following theorem tells us that given a representation with a universal partition, the greatest lower bound of the representatives of two sets represents, if it exists, the intersection between the two sets.

Theorem 2.1 *Let S be a class of sets with a set-containment relation and let P be its representation. Assume P has a universal partition U_P. For every $x_1, x_2 \in P$, if there exists $x_o = glb(x_1, x_2)$, then*

$$Rep^{-1}(x_1) \cap Rep^{-1}(x_2) = Rep^{-1}(x_o) \ .$$

The previous theorem suggests that given a class S of sets with a set-containment relation, in order to provide a representation for the intersection of every subclass of S (i.e. to provide a representation for $S^{\cap}$), we need to extend the representation of S to a poset that has a glb for every subset of its elements, namely a lattice. Since the MacNeille completion of a poset to a lattice is the most common way to realize such an extension (and indeed the resulting lattice has interesting properties) we investigate

the possibility of representing $S^\cap$ by means of $M(P)$, the MacNeille completion of P. We prove in the following that if a universal partition of S exists, $M(P)$ is a representation of $S^\cap$. Afterwards we discuss what happens if a universal partition does not exist.

In the next Theorem we will build an isomorphism between the closure of the class S with respect to the set-intersection operation and the normal completion of its representation.

Theorem 2.2 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. The mapping $IRep : S^\cap \mapsto M(P)$ defined as*

$$IRep(s) = (\{g \in P \mid g = Rep(r), r \in S_{\text{Base}}\}^*)_*$$

is an isomorphism. Hence $M(P)$ is a representation of $S^\cap$.

The result of Theorem 2.2, in the restricted formulation for simplicial complexes, where a universal partition always exists, was proved by Kainz, Egenhofer and Greasley [4]. An obvious consequence of Theorem 2.2 is that $\forall s_1, s_2 \in S$, $IRep(s_1 \cap s_2) = glb(IRep(s_1), IRep(s_2))$, namely the representative of the intersection of two sets is the glb of the representatives of the sets.

Since $M(P)$ is a representation of the class of sets $S^\cap$ which has a universal partition U_S, then a universal partition $U_{M(P)}$ of $M(P)$ and a mapping $M(P)_{\text{Base}} : M(P) \mapsto 2^{U_{M(P)}}$ are defined. By definition, $\forall s \in S$ we have $M(P)_{\text{Base}}(IRep(s)) = IRep(S_{\text{Base}}(s))$. Note that since $S \subseteq S^\cap$, $M(P)$ represents also the sets of S. This fact is consistent with the fact that P, the representation of S, is a subposet of $M(P)$, in the sense that for each $s \in S$ the representative of s in P is mapped to the representative of s in $M(P)$ by means of the canonical order embedding $\varphi(.)$ (see Sect. 3 of the companion paper [3]), as the following lemma shows.

Lemma 2.3 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. For each $s \in S$, we have $\varphi(Rep(s)) = IRep(s)$.*

The next corollary follows trivially from the previous Lemma.

Corollary 2.4 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. We have $\varphi(U_P) = U_{M(P)}$ and $\forall s \in S$, $\varphi(P_{\text{Base}}(Rep(s))) = M(P)_{\text{Base}}(IRep(s))$.*

Often we are interested in applying the intersection operator only to a subclass of a given class of sets with a set-containment relation. In these cases it is not convenient to build the MacNeille completion of the whole representation of the class, since it is likely that it contains much more elements than the ones we are interested in. Given the result of Theorem 2.2, we now show how to build a representation of a subclass of a given class of sets with a set-containment relation. We begin with a preliminary definition.

Definition 2.1 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. For each $V \subseteq S$ we define $C_V = ((Rep(V))_*)^\circ$, where $Rep(V)$ is the image of V by the mapping $Rep(.)$, namely $Rep(V) = \{x \in P \mid x = Rep(r), r \in V\}$.*

Given a class S of sets with a set-containment relation, a universal partition U_S, and a representation P, the following lemma characterizes the representative in $M(P)$ of a generic set $t_o \in S^\cap$. Note that by definition of $S^\cap$ there exists at least a set $V \subseteq S$ such that $t_o = \bigcap_{t \in V} t$.

Lemma 2.5 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. Let us consider $t_o \in S^\cap$ and $V \subseteq S$ such that $t_o = \bigcap_{t \in V} t$. Then we have $IRep(t_o) = (C_V{}^*)_*$.*

Note that, given the representation P, the characterization offered by Lemma 2.5 allows one to build the representative of t_o without the need to build $M(P)$. Using Lemma 2.5, the following theorem allows one to build a representation of the closure with respect to the intersection operator of a subclass T of S. This representation can also be built starting from P, without using $M(P)$.

Theorem 2.6 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. Let $T \subseteq S$ be a subclass of S and $M(P)_T = \{(C_V{}^*)_* \mid V \subseteq T\}$. The restriction of the mapping $IRep(.)$ to the domain $T^\cap$ is an isomorphism between the posets $\langle T^\cap, \subseteq\rangle$ and $\langle M(P)_T, \subseteq\rangle$. Hence $M(P)_T$ is a representation of $T^\cap$.*

Given a class S of sets with a set-containment relation and a subclass T of S, Theorem 2.6 tells us how to build a representation for $T^\cap$, the closure of T with respect to the set-intersection operator. The representation of $T^\cap$ obtained in this way, $M(P)_T$, has no links with P, the representation of S. Since we are often interested in the joint representation of a class S and the closure $T^\cap$, i.e. in the representation of the class $S \cup T^\cap$, we combine together P and $M(P)_T$. Note, however that the sets S and $T^\cap$ are not disjoint, since at least the sets of T are contained in both S and $T^\cap$ (and there can also be other common sets). From this fact it follows that the sets contained in both S and $T^\cap$, have a representative in both P and $M(P)_T$, hence the set $P \cup M(P)_T$ cannot be a representation of $S \cup T^\cap$ because it would be redundant. To eliminate this redundancy, once again we make use of the universal partition as a key to provide a unique representation of $S \cup T^\cap$.

Definition 2.2 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. Let T be a subclass of S. For each $A \in P \cup M(P)_T$, we define the mapping $I(T,P)_{\text{Base}} : P \cup M(P)_T \mapsto 2^{U_P}$ as:*

$$I(T,P)_{\mathrm{Base}}(A) = \begin{cases} P_{\mathrm{Base}}(A) & if\ A \in P \\ \varphi^{-1}(M(P)_{\mathrm{Base}}(A)) & if\ A \in M(P)_T. \end{cases}$$

Also we define a relation $\cong_I$ on the set $P \cup M(P)_T$. For every $A_1, A_2 \in P \cup M(P)_T$ we define $A_1 \cong_I A_2$ if and only if $I(T,P)_{\mathrm{Base}}(A_1) = I(T,P)_{\mathrm{Base}}(A_2)$. It is easy to see that the relation $\cong_I$ is an equivalence relation.

The fact that $I(T,P)_{\mathrm{Base}}(A) \in 2^{U_P}$ when $A \in M(P)_T$ is assured by Corollary 2.4 (p.84). The equivalence relation $\cong_I$ identifies elements of $P \cup M(P)_T$ that represent the same set in $S \cup T^{\cap}$. To have a unique representative we simply consider the quotient set of $P \cup M(P)_T$ with respect to the equivalence $\cong_I$.

Definition 2.3 *Let $[P \cup M(P)_T]_{\cong_I}$ be the quotient set of $P \cup M(P)_T$ with respect to the equivalence $\cong_I$. For every $\mathbf{A_1}, \mathbf{A_2} \in [P \cup M(P)_T]_{\cong_I}$ we define $\mathbf{A_1} \leq_I \mathbf{A_2}$ if and only if $I(T,P)_{\mathrm{Base}}(A_1) \subseteq I(T,P)_{\mathrm{Base}}(A_2)$, with $A_1 \in \mathbf{A_1}$ and $A_2 \in \mathbf{A_2}$. It is easy to see that the relation $\leq_I$ is defined independently from the choice of the representatives A_1, A_2 of the equivalence classes $\mathbf{A_1}, \mathbf{A_2}$ and that it is an order relation.*

The following theorem shows that $[P \cup M(P)_T]_{\cong_I}$ is a representation of $S \cup T^{\cap}$.

Theorem 2.7 *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. Let T be a subclass of S. The mapping $I(T,P)Rep : S \cup T^{\cap} \mapsto [P \cup M(P)_T]_{\cong_I}$ defined as*

$$I(T,P)Rep(s) = \begin{cases} [Rep(s)]_{\cong_I} & if\ s \in S \\ [IRep(s)]_{\cong_I} & if\ s \in T^{\cap}, \end{cases}$$

is an order isomorphism between the posets $\langle S \cup T^{\cap}, \subseteq \rangle$ and $\langle [P \cup M(P)_T]_{\cong_I}, \leq_I \rangle$. Hence $[P \cup M(P)_T]_{\cong_I}$ is a representation of $S \cup T^{\cap}$.

2.2 Necessary Conditions for Representation of Set-Intersection Closure

Theorem 2.2 (p.84) tells us that given a class S of sets with a set-containment relation and its representation P, the existence of a universal partition is a sufficient condition for the isomorphism between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$. Such a condition is not necessary, however, as the example in Figs. 2 and 3 of [3], presented here again as Figs. 1 (p.85) and 2 (p.85), shows.

In fact in that example, even though there is not a universal partition, we can build the isomorphism by representing the intersection of the sets A and B with the new element introduced in the poset by the MacNeille completion. To find a necessary condition for the isomorphism between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$, we can proceed in two ways. Either we have to carry out further investigations about the links between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$

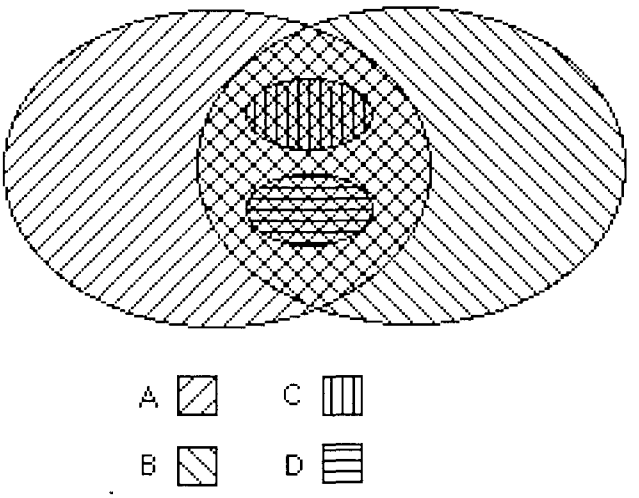


Figure 1: A class of spatial objects in the topological data model

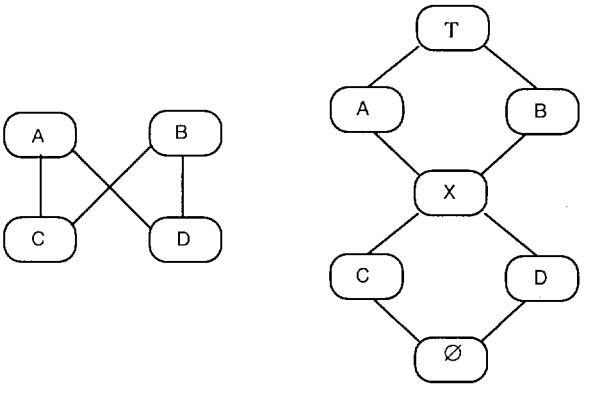


Figure 2: (left) A poset representation of the class S of Fig. 1. (right) The normal completion of the poset to the left.

or we have to find additional conditions for the class S. We now investigate both alternatives. The following definition introduces a mapping $Z : M(P) \mapsto S^{\cap}$ which we use to show further results for the first alternative.

Definition 2.4 *Let S be a class of sets with a set-containment relation and let P be a representation of S. We define the mapping $Z : M(P) \mapsto S^{\cap}$ as*

$$Z(x) = \bigcap_{y \in (\uparrow x)\varphi(P)} Rep^{-1}(\varphi^{-1}(y)) \ .$$

The following lemma shows that the mapping $Z(.)$ is an order embedding.

Lemma 2.8 *The mapping $Z : M(P) \mapsto S^{\cap}$ is an order embedding between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$.*

From previous lemma an important result follows immediately.

Lemma 2.9 *Let S be a class of sets with a set-containment relation and a representation P. We have $|M(P)| \leq |S^{\cap}|$.*

Given the above lemma, a way to find a necessary condition for the existence of an isomorphism between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$ is to find a necessary condition for the sets $S^{\cap}$ and $M(P)$ to have the same cardinality. We achieve this result by means of the mapping $Z(.)$. The following theorem states a necessary condition for the isomorphism between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$.

Theorem 2.10 *Let S be a class of sets with a set-containment relation and a representation P. If $S^{\cap}$ is isomorphic to $M(P)$, then $\forall s_o, s_1, s_2 \in S$, if $Rep(s_o) = glb_{\mathrm{P}}(Rep(s_1), Rep(s_2))$ then $s_1 \cap s_2 = s_o$.*

Theorem 2.10 gives a necessary condition for the isomorphism between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$, namely the fact that $\forall s_o, s_1, s_2 \in S$, if $Rep(s_o) = glb_{\mathrm{P}}(Rep(s_1), Rep(s_2))$ then $s_1 \cap s_2 = s_o$. Note that this condition is not sufficient, as the example of Figs. 4 and 5 of [3], presented here again as Figs. 3 (p.87) and 4 (p.87), shows. Inspecting Figs. 3(left) and 4(left) we see that $\forall s_o, s_1, s_2 \in S$, if $Rep(s_o) = glb_{\mathrm{P}}(Rep(s_1), Rep(s_2))$ then $s_1 \cap s_2 = s_o$. However posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$ are not isomorphic since sets $S^{\cap}$ and $M(P)$ have different cardinalities.

As discussed earlier, the existence of a universal partition is a sufficient, but not necessary condition for the isomorphism between the closure of a class S of sets with respect to the set-intersection operator and the MacNeille completion of a representation P of S. This means that the converse of Theorem 2.2 is not true, namely if there exists an isomorphism between $S^{\cap}$ and $M(P)$ not necessarily a universal partition of S exists (see again the example in Figs. 1 (p.85) and 2 (p.85)).

From Theorem 2.10 the following corollary follows.

Corollary 2.11 *Let S be a class of sets with a set-containment relation and a representation P. If $S^{\cap}$ is isomorphic to the Normal Completion of P, then for each $s_1 \in B_S$ and for each $s \in S$ it is $s_1 \cap s = s_1$ or $s_1 \cap s = \emptyset$.*

Another consequence of Theorem 2.10 is the following corollary, which shows that one of the conditions required for the base of the class a universal partition, namely the fact that elements of the base should not intersect, is indeed a necessary condition for the isomorphism between $S^{\cap}$ and $M(P)$.

Corollary 2.12 *Let S be a class of sets with a set-containment relation and a representation P. If $S^{\cap}$ is isomorphic to the Normal Completion of P, then for every $r_1, r_2 \in B_S$, $r_1 \cap r_2 = \emptyset$.*

Thanks to Corollaries 2.11 and 2.12, we can effectively pursue the other alternative towards defining necessary conditions for the isomorphism, namely imposing additional constraints to class S. For this aim we introduce the following definition.

Definition 2.5 *Let S be a class of sets with a set-containment relation, and let s_{T} be its greatest set. We say that S is* consistent with respect to the set-containment relation *if $\bigcup_{x \in B_S} x = s_{\mathrm{T}}$.*

The assumption of a class of sets to be consistent is reasonable in many cases, since it means that if a set B strictly contains another set A, there exist other sets in the class whose union contains the difference between B and A. Namely, the difference between B and A is an 'entity' which has somewhat to be represented in the class S. For example in a spatial database where a land with apple and pear trees (possibly mixed) is represented together with the zone with only apple trees, it seems reasonable that the zone with only pear trees is also somewhat represented in the database.

We can show that for a consistent class S the isomorphism between $S^{\cap}$ and $M(P)$, implies the existence of a universal partition of S.

Theorem 2.13 *Let S be a class of sets with a set-containment relation and a representation P. If $S^{\cap}$ is isomorphic to the Normal Completion of P and S is consistent, then there exists a universal partition on S.*

Combining the results of Theorem 2.13 and Theorem 2.2 (p.84), we obtain the following corollary that shows how strictly the existence of an isomorphism between $S^{\cap}$ and $M(P)$ is connected with that of a universal partition on S.

Corollary 2.14 *Let S be a class of sets with a set-containment relation and a representation P. Let S be consistent. Then $S^{\cap}$ is isomorphic to the Normal Completion of P iff there exists a universal partition on S.*

This result means that in a spatial database that works with poset representations of consistent classes of sets, the only way to perform spatial intersections among sets by means of the normal completion operator is to provide the database with a universal partition.

3 Representation of Set-Union Closure

3.1 Sufficient Conditions for Representation of Set-Union Closure

To obtain more generality we discuss the case of the set-union closure of a subclass of the given class of sets. We need a poset operator that, starting from the representation of the subclass, builds a representation of the set-union closure of the subclass. More exactly, the poset operator has to provide a representative for the set-union of any collection of sets in the subclass. To obtain a correct representation of the set-union closure, such a representative has to dominate (in the poset order) the representatives (i) of sets in the collection and (ii) of sets contained in a set of the collection.

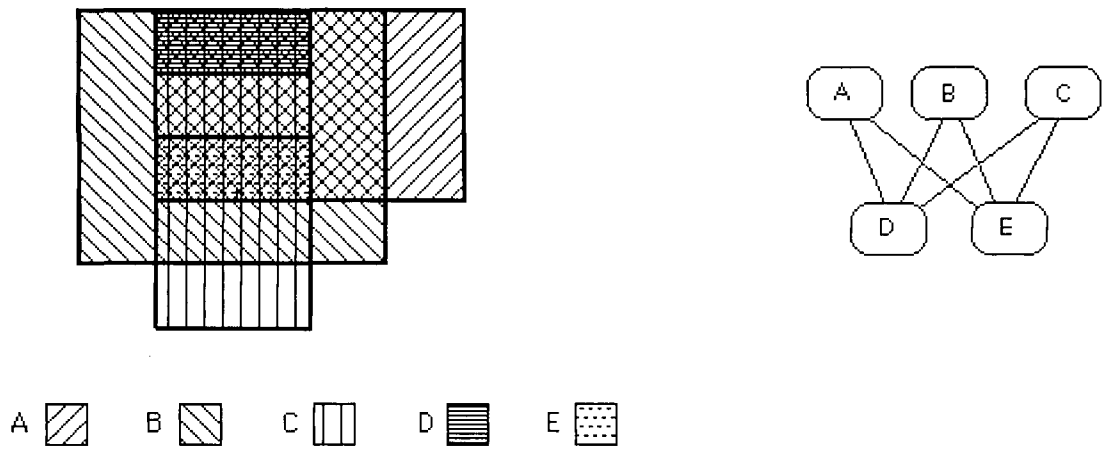


Figure 3: (left) A class S of spatial objects in the topological data model. (right) A poset representation P for this class.

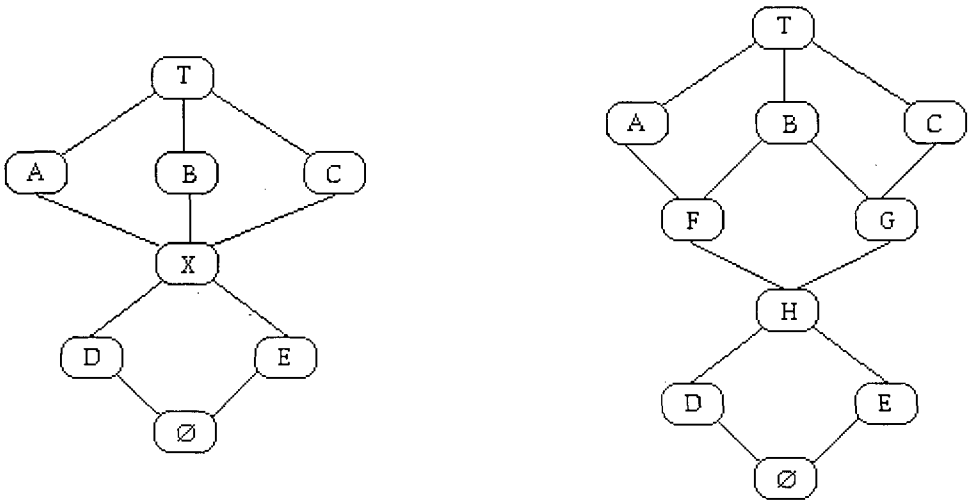


Figure 4: (left) The lattice completion of poset P representing class S of Fig. 3. (right) A poset representation of the set-intersection closure for class S.

Moreover, it has not to dominate the representative of any other set.

Now we define formally the U-completion. We begin with some preliminary definitions and propositions. Proofs of results in this section can be found in Appendix B (p.93).

Given a representation P of a class of sets S with a set-containment relation, we denote with $\langle A(P), \tilde{\preceq} \rangle$ the antichain lattice of P. We define a "natural" mapping that assigns to each element of $A(P)$ the elements of the base that are contained (in the poset) in at least one of the elements composing the considered antichain.

Definition 3.1 *Let S be a class of sets with a set-containment relation and let P be its representation. For each $A \in A(P)$, we define the mapping $A(P)_{\text{Base}} : A(P) \mapsto 2^{B_P}$ as $A(P)_{\text{Base}}(A) = \bigcup_{x \in A} P_{\text{Base}}(x)$. Note that $A(P)_{\text{Base}}(A) \in A(P)$ and $A(P)_{\text{Base}}(A) \tilde{\preceq} A$.*

Note that different antichains may be mapped by $A(P)_{\text{Base}}(.)$ into the same set of elements of the base. This means that different antichains are candidates to represent the same set. We show later how to overcome this difficulty.

The following lemma shows that to consider only the antichains instead of every subset of the poset is correct because for each subset Q of the poset, the elements of the base composing Q are the same ones composing the antichain of the maximal elements of Q.

Lemma 3.1 *Let S be a class of sets with a set-containment relation and let P be its representation. For each $B \subseteq P$, we have $A(P)_{\text{Base}}(B^{\circ}) = \bigcup_{x \in B} P_{\text{Base}}(x)$.*

The following corollary states an important property of the mapping $A(P)_{\text{Base}}(.)$.

Corollary 3.2 *The mapping $A(P)_{\text{Base}}(.)$ is an order-preserving surjective mapping from the poset $\langle A(P), \tilde{\preceq} \rangle$ to the poset $\langle 2^{B_P}, \subseteq \rangle$.*

Let us turn our attention back to the problem of multiple representations. We make use of the base to determine when different antichains represent the same set. For this aim we introduce an equivalence relation.

Definition 3.2 *Let $\langle P, \leq \rangle$ be a poset and let $A(P)$ be the set of its antichains. For each $A_1, A_2 \in A(P)$, we define $A_1 \simeq A_2$ if and only if $A(P)_{\text{Base}}(A_1) = A(P)_{\text{Base}}(A_2)$.*

It is easy to show that relation $\simeq$ over lattice $\langle A(P), \tilde{\preceq} \rangle$ is a congruence relation.

Lemma 3.3 *Let $\langle P, \leq \rangle$ be a poset and let $A(P)$ be the set of its antichains. The equivalence relation introduced by Definition 3.2 is a congruence.*

We denote as $A(P)_{\simeq}$ the quotient set of $A(P)$ modulo the relation $\simeq$. Since is a congruence, we know from elementary algebra results that the relation $\preceq$ over the set $A(P)_{\simeq}$ defined, for each $\mathbf{A}, \mathbf{B} \in A(P)_{\simeq}$, as $\mathbf{A} \preceq \mathbf{B}$ if and only if there exist $A \in \mathbf{A}$, $B \in \mathbf{B}$ such that $A \tilde{\preceq} B$, is a well defined order relation and that poset $\langle A(P)_{\simeq}, \preceq \rangle$ is a lattice. These facts are formally stated in Lemma A.5 of [3]. We write $\mathbf{A} \prec \mathbf{B}$ when $\mathbf{A} \preceq \mathbf{B}$ and $\mathbf{A} \neq \mathbf{B}$.

Definition 3.3 *Let $\langle P, \leq \rangle$ be a poset and let $A(P)$ be the set of its antichains. We define the lattice $\langle A(P)_{\simeq}, \preceq \rangle$* U-completion *of P.*

From elementary algebra (see Lemma A.6 of [3]) we know that from $A(P)_{\text{Base}}(.)$ a mapping can be derived that assigns to each element of the U-completion the unique subset of the base B_P determined by $A(P)_{\text{Base}}(.)$. Moreover, since $A(P)_{\text{Base}}(.)$ is a surjective mapping (Corollary 3.2 (p.87)), the new mapping is bijective. We call such a new mapping $AC(P)_{\text{Base}}(.)$.

The following theorem shows that the U-completion is a completion in the sense of Definition 3.8 of [3], giving a justification for its name.

Theorem 3.4 *Let S be a class of sets with a set-containment relation and let P be its representation. Assume that the base of S is a universal partition U_S, and let $\langle A(P)_{\simeq}, \preceq \rangle$ be the U-completion of P. The U-completion of P is a completion of P via the mapping $\psi : P \mapsto A(P)_{\simeq}$ defined as $\psi(x) = [\{x\}]_{\simeq}$.*

Now we present the main result of this section. Suppose a class of sets S with a set-containment relation and a representation P of S are given. Then, if the base of S is a universal partition U_S, for any subclass T of S, $A(Q)_{\simeq}$ is a representation of $T^{\cup}$, where $Q = Rep(T) \subseteq P$ is a representation of T. To show that $A(Q)_{\simeq}$ is a representation of $T^{\cup}$ we build an isomorphism $URep(\cdot)$ between them.

Theorem 3.5 *Let S be a class of sets with a set-containment relation and let P be its representation. Assume that the base of S is a universal partition U_S. Let T be a subclass of S, and let $Q = Rep(T) \subseteq P$ be a representation of T. Let $A(Q)_{\simeq}$ be the U-completion of Q. The mapping $URep : T^{\cup} \mapsto A(Q)_{\simeq}$ defined as*

$$URep(t) = [\{y \mid y = Rep(r) \text{ and } r \in S_{\text{Base}}(t)\}]_{\simeq},$$

is an isomorphism. Hence $A(Q)_{\simeq}$ is a representation of $T^{\cup}$.

In the rest of this subsection we denote as S a class of sets with a set-containment relation, as T a subclass of S, and as P a representation of S. We also assume that the base of S is a universal partition U_S.

Theorem 3.5 tells us how to build a representation for $T^{\cup}$, the closure of T with respect to set-union operator. The representation of $T^{\cup}$ obtained in this way, $A(Q)_{\simeq}$ (where $Q = Rep(T) \subseteq P$ is a representation of T), has no links with P, the representation of S. We can build a representation of the class $S \cup T^{\cup}$ combining together P and $A(Q)_{\simeq}$. Note however that the classes S and $T^{\cup}$ are not disjoint, since at least the sets of T are contained in both S and $T^{\cup}$ (and there could also be other common sets). From this fact it follows that the sets contained in both S and $T^{\cup}$, have a representative in both P and $A(Q)_{\simeq}$, hence the set $P \cup A(Q)_{\simeq}$ cannot be a representation of $S \cup T^{\cup}$ because it would be redundant. But it is easy to eliminate this redundancy using the universal partition as a key to identify elements of $P \cup A(Q)_{\simeq}$ that represent the same set, hence providing a unique representation of $S \cup T^{\cup}$. We first define an equivalence relation $\cong_U$ to identify elements of $P \cup A(Q)_{\simeq}$ that represent the same set in $S \cup T^{\cup}$.

Definition 3.4 *Let $Q \subseteq P$ be a representation of T and let $A(Q)_{\simeq}$ be the U-completion of Q. For each $X \in P \cup A(Q)_{\simeq}$, we define the mapping $U(T,P)_{\text{Base}} : P \cup A(Q)_{\simeq} \mapsto 2^{U_P}$ as:*

$$U(T,P)_{\text{Base}}(X) = \begin{cases} P_{\text{Base}}(X) & \text{if } X \in P \\ AC(Q)_{\text{Base}}(X) & \text{if } X \in A(Q)_{\simeq}. \end{cases}$$

Also we define a relation $\cong_U$ on the set $P \cup A(Q)_{\simeq}$. For every $X_1, X_2 \in P \cup A(Q)_{\simeq}$ we define $X_1 \cong_U X_2$ if and only if $U(T,P)_{\text{Base}}(X_1) = U(T,P)_{\text{Base}}(X_2)$. It is easy to see that the relation $\cong_U$ is an equivalence relation.

To have a unique representative we simply consider the quotient set of $P \cup A(Q)_{\simeq}$ with respect to the equivalence $\cong_U$.

Definition 3.5 *Let Q be a subposet of P. Let $[P \cup A(Q)_{\simeq}]_{\cong_U}$ be the quotient set of $P \cup A(Q)_{\simeq}$ with respect to the equivalence $\cong_U$. For every $\mathbf{A_1}, \mathbf{A_2} \in [P \cup A(Q)_{\simeq}]_{\cong_U}$ we define $\mathbf{A_1} \leq_U \mathbf{A_2}$ if and only if $U(T,P)_{\text{Base}}(A_1) \subseteq U(T,P)_{\text{Base}}(A_2)$, with $A_1 \in \mathbf{A_1}$ and $A_2 \in \mathbf{A_2}$. It is easy to see that the relation $\leq_U$ is defined independently from the choice of the representatives A_1, A_2 of the equivalence classes $\mathbf{A_1}, \mathbf{A_2}$ and that it is an order relation.*

The following theorem shows that $[P \cup A(Q)_{\simeq}]_{\cong_U}$ is a representation of $S \cup T^{\cup}$.

Theorem 3.6 *Let $Q \subseteq P$ be a representation of T and let $A(Q)_{\simeq}$ be the U-completion of Q. The mapping $U(T,P)Rep : S \cup T^{\cup} \mapsto [P \cup A(Q)_{\simeq}]_{\cong_U}$ defined as*

$$U(T,P)Rep(s) = \begin{cases} [Rep(s)]_{\cong_U} & if\ s \in S \\ [URep(s)]_{\cong_U} & if\ s \in T^{\cup}, \end{cases}$$

is an order isomorphism between the posets $\langle S \cup T^{\cup}, \subseteq \rangle$ and $\langle [P \cup A(Q)_{\simeq}]_{\cong_U}, \leq_U \rangle$. Hence $[P \cup A(Q)_{\simeq}]_{\cong_U}$ is a representation of $S \cup T^{\cup}$.

3.2 Necessary Conditions for Representation of Set-Union Closure

The existence of a universal partition is therefore a sufficient condition for the U-completion operator to correctly build a representation of $S^{\cup}$. However it is not necessary. For example consider the simple class containing two sets A, B, such that $A \cap B \neq \emptyset$, but neither $A \subseteq B$ nor $B \subseteq A$. In this case we have $B_S = \{A, B\}$ which is not a universal partition, but the U-completion builds a representation of $S^{\cup}$, as one can easily check. However, the second condition needed for the existence of a universal partition is indeed a necessary condition to have an isomorphism between $S^{\cup}$ and the U-completion, as shown by the following theorem.

Theorem 3.7 *Let S be a class of sets with a set-containment relation and a representation P such that $S^{\cup}$ is isomorphic to $A(P)_{\simeq}$ under a certain mapping $\mu(.)$. For each $s \in S$ there exist $r_1, r_2 \ldots r_n \in B_S$ such that $s = \bigcup_i r_i$.*

As the example given above shows, the first condition needed for the existence of a universal partition is not a necessary condition. However, a weaker condition regarding sets of the base, namely the fact that no set of the base can be contained in the set-union of other sets of the base, is necessary for the U-completion operator to correctly build a representation of $S^{\cup}$.

Theorem 3.8 *Let S be a class of sets with a set-containment relation and a representation P such that $S^{\cup}$ is isomorphic to $A(P)_{\simeq}$ under a certain mapping $\mu(.)$. For each $r_o \in B_S$ do not exist $r_1, r_2 \ldots r_n \in B_S$ such that $r_i \neq r_o$ for $1 \leq i \leq n$ and $r_o \subseteq \bigcup_i r_i$.*

Theorem 3.7 and Corollary 2.12 (p.86) together shows that the existence of universal partition is a necessary and sufficient condition so that the simultaneous application of both the lattice completion and the U-completion provides a representation for $S^{\cap\cup}$.

4 Conclusions and Future Work

Partially ordered sets (posets) are widely used to model spatial objects and relations. Lattice completion of poset representations are also widely used, due to their richer semantics and better computational complexity for their processing.

In this paper and in the companion one [3] we have addressed the problem of how to characterize sound and complete lattice representations for a set of spatial objects. More precisely, we have stated sufficient and necessary conditions for the correct use of the lattice completion operator to produce a sound and complete representation of the closure of the set of spatial objects with respect to set-intersection. We also introduced the U-completion operator, giving necessary and sufficient conditions for it to produce a a sound and complete representation of the closure of the set of spatial objects with respect to set-union. We have also shown that U-completion commutes with lattice completion and that what is obtained from the application of both completion operators (for which we have provided necessary and sufficient conditions) is minimal and unique up to isomorphism.

Finally, we have shown how to apply these completion operators when working on a subset (i.e., a view) of the set of spatial objects. In such a way the computation of the closure does not produce entities irrelevant for the view itself and only considers objects derived from those present in the view.

These results give further motivations to the use of posets to represent sets of spatial objects [4] and to reason on their topological properties. Indeed, the key role played by the concept of universal partition in the demonstration of the formal properties shows that the topological data model introduced by Paradaens and co-workers[5, 6, 7], where a universal partition is always defined by curves intersecting at their endpoints, is a good formal tool for representing and computing topological relations.

Future work will therefore concentrate, on one side, on studying what can be expressed in the topological data model by using a poset-based representation [1]. On the other side, we will investigate the definitions of algorithms for building lattice completion and U-completion [2]. Since these are known to require exponential time in the worst-case, we will focus on classes of spatial objects for which polynomial algorithms can be defined.

Acknowledgments

We thank Ralf Hartmut Güting, Bart Kuijpers, and Maurizio Talamo for useful discussions on these issues. Careful and detailed comments from two of the anonymous referees have greatly helped in improving and clarifying presentation.

References

[1] L.Forlizzi, B.Kuijpers, E.Nardelli, "Region-based query language for spatial databases in the topological data model", manuscript.

[2] L.Forlizzi, E.Nardelli, "An on-line algorithm for the MacNeille completion of a poset", Technical Report 10/99, Dip. di Matematica, Univ. di L'Aquila, Apr 1999, submitted.

[3] L.Forlizzi, E.Nardelli, "Characterization Results for the Poset Based Representation of Topological Relations – I: Introduction and Models", Informatica 23, 223-237, 1999.

[4] W.Kainz, M.Egenhofer, I.Greasley, "Modelling spatial relations and operations with partially ordered sets", Int. J. of GIS, vol. 7, no. 3, 215-229., 1993.

[5] B.Kuijpers, J.Paredaens, J.Van den Bussche, "Lossless Representation of Topological Spatial Data", 4th Int. Symp. on Large Spatial Databases (SSD'95), LNCS 951, 1-13, 1995.

[6] B.Kuijpers, J.Paredaens, J.Vandeurzen, "Semantics in Spatial Databases", LNCS 1358, 1998.

[7] J.Paredaens, "Spatial Databases, the Final Frontier", ICDT'95, LNCS 893, 14-32, 1995.

A Proofs of Section 2

Theorem 2.1. *Let S be a class of sets with a set-containment relation and let P be its representation. Assume P has a universal partition U_P. For every $x_1, x_2 \in P$, if there exists $x_o = glb(x_1, x_2)$, then*

$$Rep^{-1}(x_1) \cap Rep^{-1}(x_2) = Rep^{-1}(x_o) \ .$$

Proof. Since $x_o \leq x_1$ for each $u \in P_{\text{Base}}(x_o)$, $u \leq x_o \leq x_1$ and so $u \in P_{\text{Base}}(x_1)$. This fact implies that $P_{\text{Base}}(x_o) \subseteq P_{\text{Base}}(x_1)$. A similar argument shows that $P_{\text{Base}}(x_o) \subseteq P_{\text{Base}}(x_2)$, and so $P_{\text{Base}}(x_o) \subseteq P_{\text{Base}}(x_1) \cap P_{\text{Base}}(x_2)$. Now, let us consider a generic element $u_o \in P_{\text{Base}}(x_1) \cap P_{\text{Base}}(x_2)$. As $u_o \in P_{\text{Base}}(x_1)$ then $u_o \leq x_1$. Analogously, $u_o \leq x_2$. Since $x_o = glb(x_1, x_2)$ we have $u_o \leq x_o$, hence $P_{\text{Base}}(x_1) \cap P_{\text{Base}}(x_2) \subseteq P_{\text{Base}}(x_o)$. Therefore $P_{\text{Base}}(x_1) \cap P_{\text{Base}}(x_2) = P_{\text{Base}}(x_o)$. From this equality we have

$$Rep^{-1}(x_1) \cap Rep^{-1}(x_2) = \left(\bigcup_{u_1 \in P_{\text{Base}}(x_1)} Rep^{-1}(u_1)\right) \cap \left(\bigcup_{u_2 \in P_{\text{Base}}(x_2)} Rep^{-1}(u_2)\right)$$

and considering the distributivity of the intersection operation,

$$Rep^{-1}(x_1) \cap Rep^{-1}(x_2) = \bigcup_{u_1 \in P_{\text{Base}}(x_1), u_2 \in P_{\text{Base}}(x_2)} [Rep^{-1}(u_1) \cap Rep^{-1}(u_2)].$$

Also, observing that $\forall u_1, u_2 \in U_P$, if $u_1 = u_2$ then $Rep^{-1}(u_1) \cap Rep^{-1}(u_2) = Rep^{-1}(u_1)$, otherwise $Rep^{-1}(u_1) \cap Rep^{-1}(u_2) = \emptyset$, we can rewrite the last term as follows:

$$Rep^{-1}(x_1) \cap Rep^{-1}(x_2) = \bigcup_{u \in P_{\text{Base}}(x_1) \cap P_{\text{Base}}(x_2)} Rep^{-1}(u).$$

But since $P_{\text{Base}}(x_1) \cap P_{\text{Base}}(x_2) = P_{\text{Base}}(x_o)$, we have $Rep^{-1}(x_1) \cap Rep^{-1}(x_2) = \bigcup_{u \in P_{\text{Base}}(x_o)} Rep^{-1}(u) = Rep^{-1}(x_o)$. □

The following proposition is used in the proof of the subsequent Theorem.

Proposition A.1 *Let P be a representation with a universal partition U_P. For each $Q \subseteq P$ we let $R_Q = \bigcup_{q \in Q} P_{\text{Base}}(q)$. Then it is $(R_Q{}^*)_* = (Q^*)_*$.*

Proof. Let us consider $p \in Q^*$. For each $q \in Q$ we have $q \leq p$. For each $r \in R_Q$ there exists $q_o \in Q$ such that $r \leq q_o \leq p$ so we have $p \in R_Q{}^*$. Hence $Q^* \subseteq R_Q{}^*$. Now let us consider $t \in R_Q{}^*$. For each $q \in Q$, we have $r \leq t$ for each $r \in P_{\text{Base}}(q)$, hence for Theorem 4.7 of [3], $q \leq t$ and $t \in Q^*$. Then $R_Q{}^* \subseteq Q^*$. In conclusion we have $R_Q{}^* = Q^*$ hence the thesis. □

Theorem 2.2. *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a rep-*

resentation P. The mapping $IRep : S^{\cap} \mapsto M(P)$ defined as

$$IRep(s) = (\{g \in P \mid g = Rep(r), r \in S_{\mathrm{Base}}(s)\}^*)_*$$

is an isomorphism. Hence $M(P)$ is a representation of $S^{\cap}$.

Proof. In the following for each $s \in S^{\cap}$ we denote as G_s the set $\{g \in P \mid g = Rep(r), r \in S_{\mathrm{Base}}(s)\}$. First of all we observe that the range of $IRep(.)$ is effectively $M(P)$, since for every $G \subseteq P$ we have (Lemma A.1 of [3]) $(G^*)_* = ((G^*)_*)^*)_*$.

Now we show that $IRep(.)$ is a surjective mapping. By Definition 3.9 of [3] for every $Q \in M(P)$ we have $Q \subseteq P$ and $Q = (Q^*)_*$. Proposition A.1 (p.90) tells us that for every $Q \in M(P)$, there exists $R_Q \subseteq U_P$ such that $(R_Q{}^*)_* = (Q^*)_* = Q$. Since P has a greatest element, $R_Q{}^* \neq \emptyset$. Moreover, since P is a representation of S, for each $p \in R_Q{}^*$ there exists $Rep^{-1}(p) \in S$. Let us consider $s = \bigcap_{p \in R_Q{}^*} Rep^{-1}(p)$. Clearly it is $s \in S^{\cap}$. We have $IRep(s) = (G_s{}^*)_*$. To prove the surjectivity of $IRep(.)$ we have to show that $Q = (R_Q{}^*)_* = (G_s{}^*)_*$. Let us consider $x \in R_Q$. For every $p \in R_Q{}^*$, we have $Rep^{-1}(x) \subseteq Rep^{-1}(p)$, hence by definition of s, $Rep^{-1}(x) \subseteq s$. Since $Rep^{-1}(x) \in U_S$, we have $Rep^{-1}(x) \in S_{\mathrm{Base}}(s)$, and so $x \in G_s$. We can conclude that $R_Q \subseteq G_s$ and hence, by Lemma A.1 of [3], $G_s{}^* \subseteq R_Q{}^*$.

Now let us consider $p \in R_Q{}^*$. The definition of s tells us that $s \subseteq Rep^{-1}(p)$. For every $g \in G_s$ we have by definition of G_s, $Rep^{-1}(g) \subseteq s$. Then for every $g \in G_s$, we have $g \leq p$, hence $p \in G_s{}^*$ and $R_Q{}^* \subseteq G_s{}^*$, so the surjectivity of $IRep(.)$ follows.

To complete the proof we have to show that $IRep(.)$ is an order-embedding. If $s_1 \subseteq s_2$ then $S_{\mathrm{Base}}(s_1) \subseteq S_{\mathrm{Base}}(s_2)$. This fact implies that $G_{s_1} \subseteq G_{s_2}$. Hence, by Lemma A.1 of [3], $(G_{s_1}{}^*)_* \subseteq (G_{s_2}{}^*)_*$. Conversely if $IRep(s_1) \subseteq IRep(s_2)$ then $(G_{s_1}{}^*)_* \subseteq (G_{s_2}{}^*)_*$. This fact implies (by facts 2 and 3 of Lemma A.1 of [3]) that $G_{s_2}{}^* \subseteq G_{s_1}{}^*$. From the definition of G_{s_1}, we have $Rep(s_2) \in G_{s_2}{}^* \subseteq G_{s_1}{}^*$. Then for each $g \in G_{s_1}$, we have $g \leq Rep(s_2)$, and for Theorem 4.7 of [3] $Rep(s_1) \leq Rep(s_2)$ so $s_1 \subseteq s_2$. □

Lemma 2.3. *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. For each $s \in S$, we have $\varphi(Rep(s)) = IRep(s)$.*

Proof. Let us consider $s \in S$. By definition of the mapping $IRep(.)$ we have $IRep(s) = (G_s{}^*)_*$ where $G_s = Rep(S_{\mathrm{Base}}(s))$. By Theorem 4.7 of [3] we have $Rep(s) = lub(G_s)$, hence we have $Rep(s) \in G_s{}^*$ and $Rep(s) \leq y$ for each $y \in G_s{}^*$. This fact implies that $G_s{}^* = \{Rep(s)\}^*$. Then we have $IRep(s) = (G_s{}^*)_* = (\{Rep(s)\}^*)_* = \{Rep(s)\}_*$, where the last equality follows from fact (5) of Lemma A.1 of [3]. By Definitions 3.3 and 3.9 of [3] we have

$$\{Rep(s)\}_* = \downarrow Rep(s) = \varphi(Rep(s)),$$

hence the thesis is proved. □

The following corollary follows trivially from the previous Lemma.

Corollary 2.4. *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. We have $\varphi(U_P) = U_{M(P)}$ and $\forall s \in S$, $\varphi(P_{\mathrm{Base}}(Rep(s))) = M(P)_{\mathrm{Base}}(IRep(s))$.*

Proof. The thesis follows trivially by applying the previous Lemma to the sets of U_S. □

Lemma 2.5. *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. Let us consider $t_o \in S^{\cap}$ and $V \subseteq S$ such that $t_o = \bigcap_{t \in V} t$. Then we have $IRep(t_o) = (C_V{}^*)_*$.*

Proof. By definition of $IRep(.)$ we have $IRep(t_o) = (G_{t_o}{}^*)_*$, where $G_{t_o} = Rep(S_{\mathrm{Base}}(t_o))$. We prove the thesis showing that $G_{t_o}{}^* = C_V{}^*$. Let us consider a generic $g \in G_{t_o}$. We have $Rep^{-1}(g) \subseteq t_o \subseteq t, \forall t \in V$, hence $g \in (Rep(V))_*$, where $Rep(V) = \{x \in P \mid x = Rep(r), r \in V\}$. Then $y \in C_V$ exists such that $g \leq y$. Now let us consider a generic $z \in C_V{}^*$. We have $g \leq y \leq z$, hence $z \in G_{t_o}{}^*$.

Conversely, let us consider $z \in G_{t_o}{}^*$. For each $g \in G_{t_o}$ we have $Rep^{-1}(g) \subseteq Rep^{-1}(z)$, then, since $\bigcup_{g \in G_{t_o}} Rep^{-1}(g) = t_o$, $t_o \subseteq Rep^{-1}(z)$. For each $y \in C_V, Rep^{-1}(y) \subseteq t, \forall t \in V$, hence $Rep^{-1}(y) \subseteq \bigcap_{t \in V} t = t_o \subseteq Rep^{-1}(z)$. Then for each $y \in C_V$ we have $y \leq z$, hence $z \in C_V{}^*$. □

Theorem 2.6. *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. Let $T \subseteq S$ be a subclass of S and $M(P)_T = \{(C_V{}^*)_* \mid V \subseteq T\}$. The restriction of the mapping $IRep(.)$ to the domain $T^{\cap}$ is an isomorphism between the posets $\langle T^{\cap}, \subseteq\rangle$ and $\langle M(P)_T, \subseteq\rangle$. Hence $M(P)_T$ is a representation of $T^{\cap}$.*

Proof. First of all we observe that $M(P)_T$ is a subset of $M(P)$, since for every $(C_V{}^*)_* \in M(P)_T$ we have $(C_V{}^*)_* \subseteq P$ and (by Lemma A.1 of [3]) $(C_V{}^*)_* = (((C_V{}^*)_*)^*)_*$. Now let us consider $t_o \in T^{\cap}$. There exists $V \subseteq T$ such that $t_o = \bigcap_{t \in V} t$. By Lemma 2.5 we have $IRep(t_o) = (C_V{}^*)_* \in M(P)_T$, hence the range of the restriction of the mapping $IRep(.)$ to the domain $T^{\cap}$ is $M(P)_T$. To show that $IRep(.)$ is a surjective mapping from $T^{\cap}$ to $M(P)_T$ let us consider a generic $l \in M(P)_T$. Then $l = (C_V{}^*)_*$ for a certain $V \subseteq T$. Now let us consider $\bigcap_{t \in V} t = t_o$. We have $t_o \in T^{\cap}$ and, by Lemma 2.5, $IRep(t_o) = (C_V{}^*)_* = l$. To complete the proof we observe that the fact that $IRep(.)$ is an order embedding is implied by Theorem 2.2 (p.84). □

Theorem 2.7. *Let S be a class of sets with a set-containment relation, a universal partition U_S, and a representation P. Let T be a subclass of S. The mapping $I(T,P)Rep : S \cup T^{\cap} \mapsto [P \cup M(P)_T]_{\cong_I}$ defined as*

$$I(T,P)Rep(s) = \begin{cases} [Rep(s)]_{\cong_I} & if\ s \in S \\ [IRep(s)]_{\cong_I} & if\ s \in T^{\cap}, \end{cases}$$

is an order isomorphism between the posets $\langle S \cup T^{\cap}, \subseteq \rangle$ *and* $\langle [P \cup M(P)_T]_{\cong_I}, \leq_I \rangle$. *Hence* $[P \cup M(P)_T]_{\cong_I}$ *is a representation of* $S \cup T^{\cap}$.

Proof. First of all, we show that $I(T,P)Rep(.)$ is a surjective mapping. Let us consider $\mathbf{A} \in [P \cup M(P)_T]_{\cong_I}$. There exists $x \in P \cup M(P)_T$ such that $x \in \mathbf{A}$. Let us suppose $x \in P$. Then $s = Rep^{-1}(x) \in S$. We have $I(T,P)Rep(s) = [Rep(s)]_{\cong_I} = [x]_{\cong_I} = \mathbf{A}$. We proceed analogously in the case $x \in M(P)_T$.

Now we show that $I(T,P)Rep(.)$ is an order embedding. For each $s_1, s_2 \in S \cup T^{\cap}$, we have $s_1 \subseteq s_2$ iff $S_{\text{Base}}(s_1) \subseteq S_{\text{Base}}(s_2)$ iff $Rep(S_{\text{Base}}(s_1)) \subseteq Rep(S_{\text{Base}}(s_2))$. For each s_i ($i = 1$ or $i = 2$) either $s_i \in S$, or $s_i \notin S$ and $s_i \in T^{\cap}$. In the former case, we have $I(T,P)Rep(s_i) = [A_i]_{\cong_I}$, where $A_i \in P$ is such that $A_i = Rep(s_i)$. Then, by Definition 4.9 of [3] we have $I(T,P)_{\text{Base}}(A_i) = P_{\text{Base}}(A_i) = Rep(S_{\text{Base}}(s_i))$. In the latter case, we have $I(T,P)Rep(s_i) = [A_i]_{\cong_I}$, where $A_i \in M(P)_T$ is such that $A_i = IRep(s_i)$. By Corollary 2.4 (p.84) we have $\varphi(P_{\text{Base}}(Rep(s_i))) = M(P)_{\text{Base}}(IRep(s_i))$. Then we have $I(T,P)_{\text{Base}}(A_i) = \varphi^{-1}(M(P)_{\text{Base}}(A_i)) = P_{\text{Base}}(Rep(s_i)) = Rep(S_{\text{Base}}(s_i))$. Therefore we can conclude that $Rep(S_{\text{Base}}(s_1)) \subseteq Rep(S_{\text{Base}}(s_2))$ iff $I(T,P)_{\text{Base}}(A_1) \subseteq I(T,P)_{\text{Base}}(A_2)$ iff $[A_1]_{\cong_I} \leq_I [A_2]_{\cong_I}$ (by Definition 2.3 (p.85)). □

Lemma 2.8. *The mapping* $Z : M(P) \mapsto S^{\cap}$ *is an order embedding between the posets* $\langle S^{\cap}, \subseteq \rangle$ *and* $\langle M(P), \leq \rangle$.

Proof. Let us consider $x_1, x_2 \in M(P)$ such that $x_1 \leq x_2$. Note that since P and $\varphi(P)$ have a greatest element, $\forall y \in M(P)$, $(\uparrow y)_{\varphi(P)} \neq \emptyset$. Moreover, $\forall y \in M(P)$, if $x_2 \leq y$ then $x_1 \leq y$, so we have $\uparrow x_2 \subseteq \uparrow x_1$ and $(\uparrow x_2)_{\varphi(P)} \subseteq (\uparrow x_1)_{\varphi(P)}$. Then $\forall t \in (\uparrow x_2)_{\varphi(P)}$ we have:

$$Z(x_1) = \bigcap_{y \in (\uparrow x_1)_{\varphi(P)}} Rep^{-1}(\varphi^{-1}(y)) \subseteq Rep^{-1}(\varphi^{-1}(t))$$

and so $Z(x_1) \subseteq \bigcap_{t \in (\uparrow x_2)_{\varphi(P)}} Rep^{-1}(\varphi^{-1}(t)) = Z(x_2)$. Consider now $x_1, x_2 \in M(P)$ such that $Z(x_1) \subseteq Z(x_2)$. The definition of $Z(.)$ tells us that for every $y_2 \in (\uparrow x_2)_{\varphi(P)}$, $Z(x_2) \subseteq Rep^{-1}(\varphi^{-1}(y_2))$. For each $y_1 \in (\downarrow x_1)_{\varphi(P)}, \forall t \in (\uparrow x_1)_{\varphi(P)}$ we have $y_1 \leq x_1 \leq t$. Since $Rep(.)$ is an order isomorphism and $\varphi(.)$ is an order embedding, $\forall y_1 \in (\downarrow x_1)_{\varphi(P)}, \forall t \in (\uparrow x_1)_{\varphi(P)}$ we have $Rep^{-1}(\varphi^{-1}(y_1)) \subseteq Rep^{-1}(\varphi^{-1}(t))$, hence $\forall y_1 \in (\downarrow x_1)_{\varphi(P)}, Rep^{-1}(\varphi^{-1}(y_1)) \subseteq \bigcap_{t \in (\uparrow x_1)_{\varphi(P)}} Rep^{-1}(\varphi^{-1}(t)) = Z(x_1) \subseteq Z(x_2)$. Then for each $y_1 \in (\downarrow x_1)_{\varphi(P)}, y_2 \in (\uparrow x_2)_{\varphi(P)}$ we have $Rep^{-1}(\varphi^{-1}(y_1)) \subseteq Rep^{-1}(\varphi^{-1}(y_2))$ and so $y_1 \leq y_2$. This fact implies that $lub((\downarrow x_1)_{\varphi(P)}) \leq glb((\uparrow x_2)_{\varphi(P)})$. Since $\varphi(P)$ is both join-dense and meet-dense in $M(P)$ (Lemma A.3 of [3]), we have by Lemma A.2 of [3] $x_1 = lub((\downarrow x_1)_{\varphi(P)})$ and $x_2 = glb((\uparrow x_2)_{\varphi(P)})$, so we can conclude that $x_1 \leq x_2$. □

From the previous lemma an important result follows immediately.

Lemma 2.9. *Let* S *be a class of sets with a set-containment relation and a representation* P. *We have* $|M(P)| \leq |S^{\cap}|$.

Proof. The mapping $Z(.)$ is an order embedding so it is an injective mapping. Hence $|M(P)| \leq |S^{\cap}|$. □

The two following propositions are used in the proof of the subsequent Theorem.

Proposition A.2 *Let* S *be a class of sets with a set-containment relation and a representation* P. *For every* $q \in P$ *we have* $Rep^{-1}(q) = Z(\varphi(q))$.

Proof. By the definition of $Z(.)$ we have $Z(\varphi(q)) = \bigcap_{y \in (\uparrow \varphi(q))_{\varphi(P)}} Rep^{-1}(\varphi^{-1}(y))$. As $\varphi(q) \in (\uparrow \varphi(q))_{\varphi(P)}$, we have $Z(\varphi(q)) \subseteq Rep^{-1}(\varphi^{-1}(\varphi(q)))$.

On the other end, $\forall y \in (\uparrow \varphi(q))_{\varphi(P)}$, $\varphi(q) \leq y$ hence $Rep^{-1}(\varphi^{-1}(\varphi(q))) \subseteq Rep^{-1}(\varphi^{-1}(y))$. This fact implies

$$Rep^{-1}(\varphi^{-1}(\varphi(q))) \subseteq \bigcap_{y \in (\uparrow \varphi(q))_{\varphi(P)}} Rep^{-1}(\varphi^{-1}(y)) = Z(\varphi(q)),$$

hence the thesis follows. □

Proposition A.3 *Let* S *be a class of sets with a set-containment relation and a representation* P. *Assume that* $s_o, s_1, s_2 \in S$ *exist such that* $Rep(s_o) = glb_P(Rep(s_1), Rep(s_2))$ *and* $s_o \subset s_1 \cap s_2$. *Then the mapping* $Z : M(P) \mapsto S^{\cap}$ *is not surjective.*

Proof. Since P is a representation of S, there exist $x_o, x_1, x_2 \in P$ such that $x_o = Rep(s_o), x_1 = Rep(s_1)$ and $x_2 = Rep(s_2)$. We have $x_o = glb_P(x_1, x_2)$, and (since the MacNeille completion preserves greatest lower bounds) $\varphi(x_o) = glb_{M(P)}(\varphi(x_1), \varphi(x_2))$. Proposition A.2 tells us that $Z(\varphi(x_o)) = s_o, Z(\varphi(x_1)) = s_1$ and $Z(\varphi(x_2)) = s_2$. Suppose now that $Z(.)$ is a surjective mapping. Then there is $k \in M(P)$ such that $Z(k) = s_1 \cap s_2$. Hence we have $Z(k) \subseteq s_1 = Z(\varphi(x_1))$ that implies $k \leq \varphi(x_1)$ since $Z(.)$ is an order embedding. Analogously, $k \leq \varphi(x_2)$, hence we have $k \leq \varphi(x_o) = glb_{M(P)}(\varphi(x_1), \varphi(x_2))$. But then, since $Z(.)$ is an order embedding, we have $Z(k) = s_1 \cap s_2 \subseteq s_o = Z(\varphi(x_o))$, which is a contradiction. □

Theorem 2.10. *Let* S *be a class of sets with a set-containment relation and a representation* P. *If* $S^{\cap}$ *is isomorphic to* $M(P)$, *then* $\forall s_o, s_1, s_2 \in S$, *if* $Rep(s_o) = glb_P(Rep(s_1), Rep(s_2))$ *then* $s_1 \cap s_2 = s_o$.

Proof. Suppose that $S^{\cap}$ is isomorphic to $M(P)$ but there exist $s_o, s_1, s_2 \in S$ such that $Rep(s_o) = glb_P(Rep(s_1), Rep(s_2))$ but $s_1 \cap s_2 \neq s_o$. Since $Rep(.)$ is an order isomorphism we have $s_o \subseteq s_1$ and $s_o \subseteq s_2$, hence $s_o \subseteq s_1 \cap s_2$. But then there exist $s_o, s_1, s_2 \in S$ such that $Rep(s_o) = glb_P(Rep(s_1), Rep(s_2))$ and $s_o \subset s_1 \cap s_2$, hence, by Proposition A.3 (p.92), the injective (by

Lemma 2.8 (p.85)) mapping $Z : M(P) \mapsto S^{\cap}$ is not surjective. This fact implies $|M(P)| \leq |S^{\cap}|$, hence it cannot exist an isomorphism between the posets $\langle S^{\cap}, \subseteq \rangle$ and $\langle M(P), \leq \rangle$, but this is a contradiction. □

Corollary 2.11. *Let S be a class of sets with a set-containment relation and a representation P. If $S^{\cap}$ is isomorphic to the Normal Completion of P, then for each $s_1 \in B_S$ and for each $s \in S$ it is $s_1 \cap s = s_1$ or $s_1 \cap s = \emptyset$.*

Proof. Let us consider $s_1 \in B_S$ and $s \in S$. If $s_1 \cap s = s_1$ then the thesis is true. Otherwise, since $s_1 \in B_S$, for each $s_o \in S$ such that $s_o \subseteq s_1$ but $s_o \neq s_1$ we have $s_o = \emptyset$. Then we have $Rep(\emptyset) = glb_P(Rep(s_1), Rep(s))$. Since $S^{\cap}$ is isomorphic to the Normal Completion of P, by Theorem 2.10 we have $s_1 \cap s = \emptyset$. □

Corollary 2.12. *Let S be a class of sets with a set-containment relation and a representation P. If $S^{\cap}$ is isomorphic to the Normal Completion of P, then for every $r_1, r_2 \in B_S$, $r_1 \cap r_2 = \emptyset$.*

Proof. Let us consider $r_1, r_2 \in B_S$ and $s \in S$ such that $s \subseteq r_1$ and $s \subseteq r_2$. Since r_1 and r_2 are distinct elements, by definition of B_S, we have $s = \emptyset$. Hence we have $Rep(\emptyset) = glb_P(Rep(r_1), Rep(r_2))$. Since $S^{\cap}$ is isomorphic to the Normal Completion of P, by Theorem 2.10 we have $r_1 \cap r_2 = \emptyset$. □

Theorem 2.13. *Let S be a class of sets with a set-containment relation and a representation P. If $S^{\cap}$ is isomorphic to the Normal Completion of P and S is consistent, then there exists a universal partition on S.*

Proof. We will show that B_S is a universal partition of S. Since $S^{\cap}$ is isomorphic to $M(P)$, Corollary 2.12 tells us that $r_1, r_2 \in B_S$, $r_1 \cap r_2 = \emptyset$. Now let us consider a generic $s \in S$. Since S is consistent there exists $T \subseteq B_S$ such that $s \subseteq \bigcup_{r \in T} r$. If $s \subset \bigcup_{r \in T} r$, then there is $r_o \in T$ such that $r_o \cap s \neq \emptyset$ and $r_o \cap s \neq r_o$, and so Corollary 2.11 tells us that $S^{\cap}$ is not isomorphic to $M(P)$, yielding a contradiction. Then necessarily $s = \bigcup_{r \in T} r$, hence B_S is a universal partition of S. □

Corollary 2.14. *Let S be a class of sets with a set-containment relation and a representation P. Let S be consistent. Then $S^{\cap}$ is isomorphic to the Normal Completion of P iff there exists a universal partition on S.*

Proof. It follows immediately from Theorems 2.2 (p.84) and 2.13. □

B Proofs of Section 3

Lemma 3.1. *Let S be a class of sets with a set-containment relation and let P be its representation. For each $B \subseteq P$, we have $A(P)_{\mathrm{Base}}(B^{\circ}) = \bigcup_{x \in B} P_{\mathrm{Base}}(x)$.*

Proof. By Definition 3.1 (p.87) we have $A(P)_{\mathrm{Base}}(B^o) = \bigcup_{x \in B^o} P_{\mathrm{Base}}(x)$. Since $B^o \subseteq B$ we have obviously $A(P)_{\mathrm{Base}}(B^o) \subseteq \bigcup_{x \in B} P_{\mathrm{Base}}(x)$. On the other side, for each $x \in B$ there exists $y \in B^o$ such that $x \leq y$. Then, since $P_{\mathrm{Base}}(.)$ is an order embedding (see Corollary 4.8 of [3]), we have

$$P_{\mathrm{Base}}(x) \subseteq P_{\mathrm{Base}}(y) \subseteq \bigcup_{t \in B^o} P_{\mathrm{Base}}(t) = A(P)_{\mathrm{Base}}(B^o),$$

hence we have $\bigcup_{x \in B} P_{\mathrm{Base}}(x) \subseteq A(P)_{\mathrm{Base}}(B^o)$. □

Following propositions show some fundamental consequences of Definition 3.1 (p.87) that are used in the rest of the section.

Proposition B.1 *Let S be a class of sets with a set-containment relation and let P be its representation. For each $A \in A(P)$ we have $A(P)_{\mathrm{Base}}(A) = B_P \cap {\downarrow} A$.*

Proof. The thesis follows from Definition 3.1 (p.87) and Lemma B.2 of [3]. □

Proposition B.2 *Let S be a class of sets with a set-containment relation and let P be its representation. Given $A_1, A_2 \in A(P)$, we have:*

1. *if $A_1 \tilde{\preceq} A_2$ then $A(P)_{\mathrm{Base}}(A_1) \subseteq A(P)_{\mathrm{Base}}(A_2)$;*
2. $A(P)_{\mathrm{Base}}(lub(A_1, A_2)) = A(P)_{\mathrm{Base}}(A_1) \cup A(P)_{\mathrm{Base}}(A_2)$;
3. $A(P)_{\mathrm{Base}}(glb(A_1, A_2)) = A(P)_{\mathrm{Base}}(A_1) \cap A(P)_{\mathrm{Base}}(A_2)$.

Proof. Fact 1 follows immediately from the definition of the relation $\tilde{\preceq}$ and Proposition B.1. By Lemma A.4 of [3] we have $lub(A_1, A_2) = ({\downarrow} A_1 \cup {\downarrow} A_2)^o$ and $glb(A_1, A_2) = ({\downarrow} A_1 \cap {\downarrow} A_2)^o$, hence ${\downarrow} lub(A_1, A_2) = {\downarrow} ({\downarrow} A_1 \cup {\downarrow} A_2)^o$ and ${\downarrow} glb(A_1, A_2) = {\downarrow} ({\downarrow} A_1 \cap {\downarrow} A_2)^o$. We now observe that for any $I \subseteq P$ such that $I = {\downarrow} I$ it is $I = {\downarrow} (I^o)$. In fact for any $x \in I$ there is $y \in I^o$ such that $x \leq y$, hence $x \in {\downarrow} (I^o)$. Conversely for any $x \in {\downarrow} (I^o)$ there is $y \in I^o \subseteq I$ such that $x \leq y$. Since $y \in I$, it is $x \in {\downarrow} I = I$. It is trivial to show that ${\downarrow} A_1 \cup {\downarrow} A_2 = {\downarrow} ({\downarrow} A_1 \cup {\downarrow} A_2)$ and ${\downarrow} A_1 \cap {\downarrow} A_2 = {\downarrow} ({\downarrow} A_1 \cap {\downarrow} A_2)$. Then using the above observation we have ${\downarrow} lub(A_1, A_2) = {\downarrow} A_1 \cup {\downarrow} A_2$ and ${\downarrow} glb(A_1, A_2) = {\downarrow} A_1 \cap {\downarrow} A_2$. Then facts 2 and 3 follow from Proposition B.1. □

Proposition B.3 *Given $X \in 2^{B_P}$, we have:*

1. *$X \in A(P)$;*
2. *$A(P)_{\mathrm{Base}}(X) = X$;*
3. *$\forall X, Z \in 2^{B_P}$, $X \subseteq Z$ if and only if $X \tilde{\preceq} Z$.*

Proof. Facts 1 and 2 follow from the definition of B_P. To show fact 3 observe that $\forall X, Z \in 2^{B_P}$ if $X \subseteq Z$, by Definition 3.4 of [3] follows immediately ${\downarrow} X \subseteq {\downarrow} Z$, hence

$X \tilde{\preceq} Z$. Conversely if $X \tilde{\preceq} Z$, then we have $A(P)_{\text{Base}}(X) \subseteq A(P)_{\text{Base}}(Z)$ by fact 1 of Proposition B.2 and $X \subseteq Z$ by fact 2. □

Corollary 3.2. *The mapping $A(P)_{\text{Base}}(.)$ is an order-preserving surjective mapping from the poset $\langle A(P), \tilde{\preceq}\rangle$ to the poset $\langle 2^{B_P}, \subseteq\rangle$.*

Proof. From fact 1 of Proposition B.2 follows that $A(P)_{\text{Base}}(.)$ is an order-preserving mapping. Surjectivity follows from facts 1 and 2 of Proposition B.3. □

Lemma 3.3. *Let $\langle P, \leq\rangle$ be a poset and let $A(P)$ be the set of its antichains. The equivalence relation introduced by Definition 3.2 (p.88) is a congruence.*

Proof. Recalling Definition 3.2 (p.88) and Definition A.2 of [3], the thesis follows from facts 2 and 3 of Proposition B.2. □

Theorem 3.4. *Let S be a class of sets with a set-containment relation and let P be its representation. Assume P has a universal partition U_P, and let $\langle A(P)_{\simeq}, \preceq\rangle$ be the U-completion of P. The U-completion of P is a completion of P via the mapping $\psi : P \mapsto A(P)_{\simeq}$ defined as $\psi(x) = [\{x\}]_{\simeq}$.*

Proof. We know by Lemma A.5 of [3] that $A(P)_{\simeq}$ is a complete lattice so we only have to show that the map $\psi(.)$ is an order embedding. Since P has a universal partition U_P, we know from Corollary 4.8 of [3] that $P_{\text{Base}}(.)$ is an order embedding. Then given $x_1, x_2 \in P$, we have $x_1 \leq x_2$ iff $P_{\text{Base}}(x_1) \subseteq P_{\text{Base}}(x_2)$ iff $A(P)_{\text{Base}}(\{x_1\}) \subseteq A(P)_{\text{Base}}(\{x_2\})$ iff $AC(P)_{\text{Base}}([\{x_1\}]_{\simeq}) \subseteq AC(P)_{\text{Base}}([\{x_2\}]_{\simeq})$ iff $[\{x_1\}]_{\simeq} \preceq [\{x_2\}]_{\simeq}$. □

The following three Propositions are used in the proof of the subsequent Theorem.

Proposition B.4 *Let S be a class of sets with a set-containment relation and let P be its representation. Let $\langle A(P)_{\simeq}, \preceq\rangle$ be the U-completion of P. For each $A \in A(P)_{\simeq}$, $AC(P)_{\text{Base}}(A) \in A$.*

Proof. For each $A \in A(P)_{\simeq}$, $AC(P)_{\text{Base}}(A) \in 2^{B_P}$. Then, by fact 2 of Proposition B.3, we have $AC(P)_{\text{Base}}(A) = A(P)_{\text{Base}}(AC(P)_{\text{Base}}(A))$. Now let us consider $a \in A$. We have, by definition of $AC(P)_{\text{Base}}(.)$, $A(P)_{\text{Base}}(a) = AC(P)_{\text{Base}}(A)$. Then $A(P)_{\text{Base}}(a) = A(P)_{\text{Base}}(AC(P)_{\text{Base}}(A))$, hence $a \simeq AC(P)_{\text{Base}}(A)$ and the thesis is proved. □

Proposition B.5 *Let S be a class of sets with a set-containment relation and let P be its representation. Let $\langle A(P)_{\simeq}, \preceq\rangle$ be the U-completion of P. For each $A, B \in A(P)_{\simeq}$, it is $A \preceq B$ if and only if $AC(P)_{\text{Base}}(A) \subseteq AC(P)_{\text{Base}}(B)$.*

Proof. Let us consider $A, B \in A(P)_{\simeq}$ such that $A \preceq B$. By definition there exist $a \in A$, $b \in B$ such that $a \tilde{\preceq} b$. Recalling Propositions B.4 and B.2 we have
$AC(P)_{\text{Base}}(A) = A(P)_{\text{Base}}(a) \subseteq A(P)_{\text{Base}}(b) = AC(P)_{\text{Base}}(B)$.
Conversely, suppose there exist $A, B \in A(P)_{\simeq}$ such that $AC(P)_{\text{Base}}(A) \subseteq AC(P)_{\text{Base}}(B)$. We have $AC(P)_{\text{Base}}(A), AC(P)_{\text{Base}}(B) \in 2^{B_P} \subseteq A(P)$ hence by Proposition B.3 (fact 3)

$$AC(P)_{\text{Base}}(A) \tilde{\preceq} AC(P)_{\text{Base}}(B).$$

Since $AC(P)_{\text{Base}}(A) \in A$ and $AC(P)_{\text{Base}}(B) \in B$ (Proposition B.4), by Definition 3.3 (p.88) we have $A \preceq B$. □

Proposition B.6 *Let S be a class of sets with a set-containment relation and let P be its representation. Assume that the base of S is a universal partition U_S. Let T be a subclass of S, and let $Q = Rep(T) \subseteq P$ be a representation of T. Consider $t \in T^{\cup}$ and define the set $X_t = \{y \mid y = Rep(r),\ r \in S_{\text{Base}}(t)\} \in 2^{U_P}$. Since $t \in T^{\cup}$ there exist $t_1, t_2, \ldots, t_n \in T$ such that $t = \bigcup_i t_i$. Then we have $[X_t]_{\simeq} = [\{q_1, q_2, \ldots, q_n\}^o]_{\simeq} \in A(Q)_{\simeq}$, where $q_i = Rep(t_i)$.*

Proof. Consider the antichain $\{q_1, q_2, \ldots, q_n\}^o \in A(Q)$. We have

$$A(Q)_{\text{Base}}(\{q_1, q_2, \ldots, q_n\}^o) = \bigcup_i P_{\text{Base}}(q_i)$$

by Lemma 3.1 (p.87). Then, by Definition 4.9 of [3] it is $\bigcup_i P_{\text{Base}}(q_i) = \bigcup_i \{y \mid y = Rep(r),\ r \in S_{\text{Base}}(t_i)\} = \{y \mid y = Rep(r),\ r \in \bigcup_i S_{\text{Base}}(t_i)\}$. By hypothesis we have $t = \bigcup_i t_i$. Then Corollary 4.3 of [3] implies $\bigcup_i S_{\text{Base}}(t_i) = S_{\text{Base}}(t)$ and consequently we have $\{y \mid y = Rep(r),\ r \in \bigcup_i S_{\text{Base}}(t_i)\} = \{y \mid y = Rep(r),\ r \in S_{\text{Base}}(t)\} = X_t$. In conclusion we have $X_t = A(Q)_{\text{Base}}(\{q_1, q_2, \ldots, q_n\}^o) = AC(Q)_{\text{Base}}([\{q_1, q_2, \ldots, q_n\}^o]_{\simeq})$ where the last equality follows from the definition of mapping $AC(Q)_{\text{Base}}(.)$. Hence by Proposition B.4, we have $X_t \in [\{q_1, q_2, \ldots, q_n\}^o]_{\simeq}$. □

Theorem 3.5. *Let S be a class of sets with a set-containment relation and let P be its representation. Assume that the base of S is a universal partition U_S. Let T be a subclass of S, and let $Q = Rep(T) \subseteq P$ be a representation of T. Let $A(Q)_{\simeq}$ be the U-completion of Q. The mapping $URep : T^{\cup} \mapsto A(Q)_{\simeq}$ defined as*

$$URep(t) = [\{y \mid y = Rep(r) \text{ and } r \in S_{\text{Base}}(t)\}]_{\simeq},$$

is an isomorphism. Hence $A(Q)_{\simeq}$ is a representation of $T^{\cup}$.

Proof. First of all, we show that $URep(.)$ is a surjective mapping. Let us consider $A \in A(Q)_{\simeq}$. There exists

$\{q_1, q_2, \ldots, q_n\} \in A(Q)$ such that $[\{q_1, q_2, \ldots, q_n\}]_{\simeq} = A$. Let us consider $t = \bigcup_i Rep^{-1}(q_i) \in T^{\cup}$. We have $URep(t) = [\{y \mid y = Rep(r),\ r \in S_{\mathrm{Base}}(t)\}]_{\simeq}$. Since $t = \bigcup_i Rep^{-1}(q_i)$, from Proposition B.6 we have $URep(t) = [\{q_1, q_2, \ldots, q_n\}^o]_{\simeq} = [\{q_1, q_2, \ldots, q_n\}]_{\simeq} = A$. Now we show that $URep(.)$ is an order embedding. For each $s_1, s_2 \in T^{\cup}$, we have $s_1 \subseteq s_2$ iff $S_{\mathrm{Base}}(s_1) \subseteq S_{\mathrm{Base}}(s_2)$ (Corollary 4.3 of [3]). Let us consider $A = \{y \mid y = Rep(r),\ r \in S_{\mathrm{Base}}(s_1)\}$ and $B = \{y \mid y = Rep(r),\ r \in S_{\mathrm{Base}}(s_2)\}$. Then for each $s_1, s_2 \in T^{\cup}$, we have $S_{\mathrm{Base}}(s_1) \subseteq S_{\mathrm{Base}}(s_2)$ iff $A \subseteq B$ and hence, by fact 3 of Proposition B.3 (p.93), iff $A \tilde{\preceq} B$. In conclusion, recalling Definition 3.3 (p.88), we have $s_1 \subseteq s_2$ iff $URep(s_1) = [A]_{\simeq} \preceq [B]_{\simeq} = URep(s_2)$. □

Theorem 3.6. *Let $Q \subseteq P$ be a representation of T and let $A(Q)_{\simeq}$ be the U-completion of Q. The mapping $U(T,P)Rep : S \cup T^{\cup} \mapsto [P \cup A(Q)_{\simeq}]_{\cong_U}$ defined as*

$$U(T,P)Rep(s) = \begin{cases} [Rep(s)]_{\cong_U} & if\ s \in S \\ [URep(s)]_{\cong_U} & if\ s \in T^{\cup}, \end{cases}$$

is an order isomorphism between the posets $\langle S \cup T^{\cup}, \subseteq \rangle$ and $\langle [P \cup A(Q)_{\simeq}]_{\cong_U}, \leq_U \rangle$. Hence $[P \cup A(Q)_{\simeq}]_{\cong_U}$ is a representation of $S \cup T^{\cup}$.

Proof. First of all, we show that $U(T,P)Rep(.)$ is a surjective mapping. Let us consider $A \in [P \cup A(Q)_{\simeq}]_{\cong_U}$. There exists $x \in P \cup A(Q)_{\simeq}$ such that $x \in A$. Let us suppose $x \in P$. Then, $s = Rep^{-1}(x) \in S$. We have $U(T,P)Rep(s) = [Rep(s)]_{\cong_U} = [x]_{\cong_U} = A$. We proceed analogously in the case $x \in A(Q)_{\simeq}$.

Now we show that $U(T,P)Rep(.)$ is an order embedding. For each $s_1, s_2 \in S \cup T^{\cup}$, we have $s_1 \subseteq s_2$ iff $S_{\mathrm{Base}}(s_1) \subseteq S_{\mathrm{Base}}(s_2)$ iff $Rep(S_{\mathrm{Base}}(s_1)) \subseteq Rep(S_{\mathrm{Base}}(s_2))$. For each s_i ($i = 1$ or $i = 2$) either $s_i \in S$, or $s_i \notin S$ and $s_i \in T^{\cup}$. In the former case, we have $U(T,P)Rep(s_i) = [A_i]_{\cong_U}$, where $A_i \in P$ is such that $A_i = Rep(s_i)$. Then, by Definition 4.9 of [3] we have $U(T,P)_{\mathrm{Base}}(A_i) = P_{\mathrm{Base}}(A_i) = Rep(S_{\mathrm{Base}}(s_i))$. In the latter case, we have $U(T,P)Rep(s_i) = [A_i]_{\cong_U}$, where $A_i \in A(Q)_{\simeq}$ is such that $A_i = URep(s_i)$. By definition of the mapping $URep(.)$ we have $Rep(S_{\mathrm{Base}}(s_i)) = \{y \mid y = Rep(r),\ r \in S_{\mathrm{Base}}(s_i)\} \in A_i$. Then we have $U(T,P)_{\mathrm{Base}}(A_i) = AC(Q)_{\mathrm{Base}}(A_i) = A(Q)_{\mathrm{Base}}(Rep(S_{\mathrm{Base}}(s_i)))$ where the last equality follows from the definition of the mapping $AC(Q)_{\mathrm{Base}}(A_i)$ and from the fact that $Rep(S_{\mathrm{Base}}(s_i)) \in A_i$.

By fact 2 of Proposition B.3 (p.93) we have $A(Q)_{\mathrm{Base}}(Rep(S_{\mathrm{Base}}(s_i))) = Rep(S_{\mathrm{Base}}(s_i))$, hence $U(T,P)_{\mathrm{Base}}(A_i) = Rep(S_{\mathrm{Base}}(s_i))$. Therefore we can conclude that $s_1 \subseteq s_2$ iff $Rep(S_{\mathrm{Base}}(s_1)) \subseteq Rep(S_{\mathrm{Base}}(s_2))$ iff $U(T,P)_{\mathrm{Base}}(A_1) \subseteq U(T,P)_{\mathrm{Base}}(A_2)$ iff $[A_1]_{\cong_U} \leq_U [A_2]_{\cong_U}$ (by Definition 3.5 (p.88)). □

The two following Propositions are used in the proofs of the two subsequent Theorems.

Proposition B.7 *Let S be a class of sets with a set-containment relation and a representation P such that $S^{\cup}$ is isomorphic to $A(P)_{\simeq}$ under a certain mapping $\mu(.)$. For each $r \in B_S$ we have $\mu(r) = [\{y\}]_{\simeq}$ where $y \in B_P$. Conversely, for each $y \in B_P$ there is $r \in B_S$ such that $\mu(r) = [\{y\}]_{\simeq}$*

Proof. We denote as s_B the least set of S and as $\perp$ the least element of P. Since $\mu(.)$ is an isomorphism it maps the least set of $S^{\cup}$ to the least element of $A(Q)_{\simeq}$, namely $\mu(s_B) = [\{\perp\}]_{\simeq}$. Let us consider $r \in B_S$. If $r = s_B$ the thesis is true. Hence we assume $s_B \subset r$ which implies $\perp \subset B_P$. Now let us consider $A \in A(P)$ such that $A \in \mu(r)$ and suppose $A = \{y_1, \ldots, y_k\}$ with $k > 1$. Since A is an antichain, for each y_j with $1 \leq j \leq k$ we have $\perp < y_j$. By Definition A.4 of [3] we have $\{y_1\} \tilde{\prec} A$, hence $[\{y_1\}]_{\simeq} \prec \mu(r)$ and $\mu^{-1}([\{y_1\}]_{\simeq}) \subset r$. But then, since $r \in B_S$, we have $\mu^{-1}([\{y_1\}]_{\simeq}) = s_B$ and this is a contradiction because $\perp < y_1$ implies $s_B = \mu^{-1}([\{\perp\}]_{\simeq}) \subset \mu^{-1}([\{y_1\}]_{\simeq})$. Then it is necessarily $A = \{y\}$. Suppose it is $y \notin B_P$. Then, since $\perp \subset B_P$, there is $z \in B_P$ such that $\perp < z < y$. By the definition of the relation $\prec$, it follows $[\{z\}]_{\simeq} \prec [\{y\}]_{\simeq} = \mu(r)$ and consequently $\mu^{-1}([\{z\}]_{\simeq}) \subset r$. But then $\mu^{-1}([\{z\}]_{\simeq}) = s_B$ which is a contradiction because $\perp \neq z$. Then we have $\mu(r) = [\{y\}]_{\simeq}$ with $y \in B_P$. The second part of the thesis, namely the fact that for each $y \in B_P$ there is $r \in B_S$ such that $\mu(r) = [\{y\}]_{\simeq}$, follows from the first part observing that, by definition, $|B_S| = |B_P|$ and that $\mu(.)$, being an isomorphism, is injective. □

Proposition B.8 *Let S be a class of sets with a set-containment relation and a representation P such that $S^{\cup}$ is isomorphic to $A(P)_{\simeq}$ under a certain mapping $\mu(.)$. For each $s \in S$ we have*

$$AC(P)_{\mathrm{Base}}(\mu(s)) = A_s,$$

where $A_s = \{z \mid [\{z\}]_{\simeq} = \mu(r)$ with $r \in S_{\mathrm{Base}}(s)\}$.

Proof. By Proposition B.7 we know $A_s \subseteq B_P$, hence $A_s \in A(P)$ (fact 1 of Proposition B.3). Now let us consider $r \in S_{\mathrm{Base}}(s)$. By Proposition B.7 there exists $z \in A_s \subseteq B_P$ such that $[\{z\}]_{\simeq} = \mu(r) \preceq \mu(s)$. Let us consider an arbitrary $B \in A(P)$ such that $B \in \mu(s)$. We have $\{z\} \tilde{\preceq} B$, hence (see Lemma A.4 of [3]) there exists $y \in B$ such that $z \leq y$. But then, since $z \in B_P$, we have $z \in P_{\mathrm{Base}}(y) \subseteq A(P)_{\mathrm{Base}}(B) = AC(P)_{\mathrm{Base}}(\mu(s))$. Hence $A_s \subseteq AC(P)_{\mathrm{Base}}(\mu(s))$. Conversely let us consider $w \in AC(P)_{\mathrm{Base}}(\mu(s))$. By Proposition B.7 there is $r \in B_S$ such that $[\{w\}]_{\simeq} = \mu(r)$. By definition of $A(P)_{\mathrm{Base}}(.)$, given $B \in \mu(s)$, there is $y \in B$ such that $w \in P_{\mathrm{Base}}(y)$, hence $w \leq y$. But then $\{w\} \tilde{\preceq} B$, hence $[\{w\}]_{\simeq} = \mu(r) \preceq \mu(s)$ which in turn implies $r \subseteq s$. Since $r \in B_S$, we have $r \in S_{\mathrm{Base}}(s)$, hence $w \in A_s$. Therefore $AC(P)_{\mathrm{Base}}(\mu(s)) \subseteq A_s$. □

Theorem 3.7. *Let S be a class of sets with a set-containment relation and a representation P such that $S^{\cup}$*

is isomorphic to $A(P)_{\simeq}$ *under a certain mapping* $\mu(.)$. *For each* $s \in S$ *there exist* $r_1, r_2 \ldots r_n \in B_S$ *such that* $s = \bigcup_i r_i$.

Proof. Let us suppose that there exists $s_o \in S$ for which the hypothesis is false. Then for each $r \in S_{\text{Base}}(s_o)$ we have $r \subset s_o$ and $\bigcup_{r \in S_{\text{Base}}(s_o)} r = s_1 \subset s_o$. Obviously $s_1 \in S^{\cup}$ and $S_{\text{Base}}(s_1) = S_{\text{Base}}(s_o)$ which implies $A_{s_o} = A_{s_1}$, where A_{s_o} and A_{s_1} are defined according to the hypothesis of Proposition B.8. But then, recalling Propositions B.4 and B.8, we have $\mu(s_o) = [AC(P)_{\text{Base}}(\mu(s_o))]_{\simeq} = [A_{s_o}]_{\simeq} = [A_{s_1}]_{\simeq} = [AC(P)_{\text{Base}}(\mu(s_1))]_{\simeq} = \mu(s_1)$, which is a contradiction because $s_1 \subset s_o$. □

Theorem 3.8. *Let* S *be a class of sets with a set-containment relation and a representation* P *such that* $S^{\cup}$ *is isomorphic to* $A(P)_{\simeq}$ *under a certain mapping* $\mu(.)$. *For each* $r_o \in B_S$ *do not exist* $r_1, r_2 \ldots r_n \in B_S$ *such that* $r_i \neq r_o$ *for* $1 \leq i \leq n$ *and* $r_o \subseteq \bigcup_i r_i$.

Proof. Suppose that the thesis is not true for some $r_o \in B_S$. By Proposition B.7, $\forall r_i$, $1 \leq i \leq n$, there exists $z_i \in B_P$ such that $[\{z_i\}]_{\simeq} = \mu(r_i)$. Consider $B = \{z_i,\ 1 \leq i \leq n\} \in A(P)$. There exists $s_1 \in S^{\cup}$ such that $\mu(s_1) = [B]_{\simeq}$. For each i, we have $\{z_i\} \tilde{\preceq} B$ (fact 3 of Proposition B.3) hence $\mu(r_i) = [\{z_i\}]_{\simeq} \preceq [B]_{\simeq} = \mu(s_1)$ which implies $r_i \subseteq s_1$. Then we have $r_o \subseteq \bigcup_i r_i \subseteq s_1$, hence $\mu(r_o) \preceq \mu(s_1)$. Since $r_o \in B_S$, by Proposition B.7, there exists $z_o \in B_P$ such that $[\{z_o\}]_{\simeq} = \mu(r_o) \preceq \mu(s_1)$. Since $\mu(.)$ is an isomorphism, $z_i \neq z_o$ for $1 \leq i \leq n$. But then $\{z_o\} \tilde{\preceq} B$, hence there exists $z_h \in B$ such that $z_o \leq z_h$, but this is a contradiction since $z_o, z_h \in B_P$ and $z_o \neq z_h$. □

ViCRO: An Interactive and Cooperative VideoRecording on-demand System over MBone

Giancarlo Fortino and Libero Nigro
Laboratorio di Ingegneria del Software
Dipartimento di Elettronica Informatica e Sistemistica
Università della Calabria, I-87036 Rende (CS), Italy
Phone: +39 0984 494748, Fax: +39 0984 494713
E-mail:{g.fortino, l.nigro}@unical.it

Keywords: Multimedia Modelling, VCRoD, Java, Internet MBone, RTSP, RTP, LRMP

Edited by: Branko Souček
Received: March 16, 1999 **Revised:** June 13, 1999 **Accepted:** July 17, 1999

This paper presents an interactive and cooperative VideoConference Recording On-demand system (ViCRO) designed for remote playback, recording and browsing multimedia sessions over the Internet MBone. It consists of Media Servers, possibly cluster-based, and Media Clients. The interaction multi-client/multi-server is based on the RTSP protocol for streaming control, on the RTP protocol for multimedia data streaming, on the LRMP protocol for reliable messaging and on the SAP protocol for group rendezvous. The time-critical and continuous components of the media clients and media servers are built by using Java and Actor based Framework (JAF), i.e., a variant of the Actor model specialised to multimedia requirements. JAF centres on timing predictability, customisable scheduling and a modular specification of QoS constraints. The main goal is to integrate MBone and WWW technologies towards media-on-demand and virtual collaborative work in heterogeneous environments. The paper introduces the Java-enabled ViCRO system and describes its application in several scenarios ranging from teleteaching to videoconferencing.

1 Introduction

The rapid growth, increasing bandwidth and the availability of low-cost multimedia end systems has made it possible to use Internet for multimedia applications ranging from telephony to conferencing, distance learning, media-on-demand and broadcast applications. Protocols for transporting real-time data, for reserving resources to guarantee quality of service, for initiating and controlling multimedia sessions have been developed, standardised and are being widely used. [1]

Today, Internet multimedia on-demand can fulfil the expectations that were not accomplished by the VoD systems (Rowe et al. 1995) whose trials were not exactly successful. The challenging and possibly winning key points of Internet Multimedia on demand solution (Schulzrinne 1997) are: re-use of existing infrastructure, flexible media (e.g., modem, wireless, cable, ATM, LAN), one service among many others, quality scalability, adaptive compression, easily integration with WWW, security through encryption, cheap authoring and lots of content. Internet Multimedia can support different delivery modes: on-demand (i.e., unicast), near-on-demand (programmed transmission on multicast), multicast. In addition, IP-multicast, deployed in the Internet as an overlay network, referred to as the MBone, facilitates a scaleable data delivery and stream control to large interacting user groups.

In this context, ViCRO, a Java-enabled on-demand system for playing and recording multimedia sessions (audio, video, text, graphics) on the Internet MBone (Kumar 1998), has been designed. The goal of the system is to create an environment where users could not only easily request playback and recording of electronic lecturers, seminars, meetings, entertainment events and so forth but also collaborate with each other as a virtual tightly-coupled group of workmates. Such an environment could directly fulfil the needs present in undergraduate courses, company and academic meetings, in which the group participants work by exchanging active messages. The ViCRO system consists of basic building blocks, Media Servers and Media Clients, subdivided into components. They deal with the requirements of media and control streaming on the Internet such as retrieval of media from a media server, management of live events, integration of conference, and enriching of contents. They are based on the Internet multimedia protocol stack: RTP/RTCP (Schulzrinne et al. 1996), RTSP (Schulzrinne et al. 1998), SAP (Handley 1997), SDP (Handley & Jacobson 1998) to guarantee interoperability toward other systems. The components of the media clients and media servers are achieved (Fortino & Nigro 1998; Fortino et al. 1998) by JAF, a Java and Actor-based Framework. It is a variant of the Actor model (Agha 1986) which centres on a modular approach to synchronisation, timing con-

[1] A preliminary version of this paper has been presented at the SCS Euromedia99 conference, Proceedings pp. 120-124.

straints and QoS specifications (Ren et al. 1997). In this context choosing Java ensures the system multi-platform portability, and heterogeneous environment interoperability.

The paper is structured as follows. Section 2 describes JAF and its use in the modelling of multimedia systems. Section 3 illustrates the design principles and the architecture of the ViCRO system. Section 4 shows some application scenarios for ViCRO. Section 5 summarises some related work. Section 6 highlights the implementation status of the prototype. Finally, the conclusion and some directions of further work are given.

2 Multimedia Systems based on JAF

A multimedia application is modelled as a collection of autonomous, distributed, concurrent and mobile multimedia and inter-media actors interacting one to another to achieve their common goal, e.g., real-virtual multimedia conferencing sessions, automatic contents generation or video on-demand services.

2.1 Java and Actor-based Framework

It is a variant of the Actor model (Agha 1986) (Nigro & Pupo 1998) which centres on light-weight actors and a modular approach to synchronisation and timing constraints. Actors are finite state machines. The arrival of an event (i.e., a message) causes a state transition and the execution of an atomic action. At the action termination the actor is ready to receive a next message and so forth. Actors do not have internal threads for message processing. At most one action can be in progress in an actor at a given time. Actors can be grouped into clusters (i.e., subsystems). A subsystem is allocated to a distinct physical processor. It is regulated by a control machine which hosts a time notion and is responsible of message buffering (scheduling) and dispatching.

The control machine can be customised by programming. For instance, in (Nigro & Pupo 1998, Fortino & Nigro 1998) a specialisation of the control machine for hard real-time systems is proposed, where scheduling is based on messages time-stamped by a time validity window [tmin, tmax] expressing the interval of admissible delivery times. Message selection and dispatching is based on an Earliest Deadline First strategy. Within a subsystem, actor concurrency rests on message processing interleaving. True parallelism is possible among actors belonging to distinct subsystems. A distinguishing feature of the actor framework is the modular handling of timing constraints. Application actors are developed according to functional issues only. They are not aware of when they are activated by a message. Timing requirements are responsibility of RTsynchronizers (Ren et al. 1997), i.e., special actors which capture "just sent messages" (including messages received from the network) and apply to them timing constraints affecting scheduling. Control machines of a distributed system can be interconnected by a network and real time protocol so as to fulfil system-wide timing constraints.

2.2 Modelling Multimedia Systems

2.2.1 Architecture of a single Multimedia Session

A multimedia session is assigned to an actor system split into two parts specialised to handle the requirements existing at both the server (transmitter) and client(s) (receiver(s)) sides of the application (Fortino et al. 1998). The transmitter side is typically devoted to achieving the multimedia data, e.g., from stored files, and to send them through a network binding to the client for the final presentation. Specific timing and synchronisation constraints exist and should be managed respectively at the server and client side to ensure quality-of-service parameters. To this purpose both server and client subsystems can be equipped with a distinct multimedia control machine with a QoSsynchronizer. Bindings, i.e., logical communication channels, are introduced for connecting transmitter/receiver subsystems. Bindings can be point-to-point (i.e., unicast) and point-to-multipoint (i.e., multicast). A binding is created by a bind operation originated from media-actors called Binders. A binder governs the on-going flow of data (e.g., continuous media or control messages) sent into the binding. It hides particular transmission mechanisms (e.g., network and transport protocols). It can also monitor the binding QoS so as to provide information such as throughput, jitter, latency and packet loss statistics. A QoSsynchronizer is an RTsynchronizer which captures and verifies QoS timing constraints. As an example, the QoSsynchronizer in a client subsystem can perform fine-grain inter-media synchronisation (e.g., lip sync).

The notion of multimedia presentation is encapsulated in a media-actor called a Manager (or Supervisor). According to the specification it orchestrates the media objects (time-dependent and time-independent) by interacting with media-actors called Streamers. A Streamer is a periodic actor that accesses to digital media information through media passive object (MediaFile, MediaDevice, MediaNetSource) and sends it to Binders or Presenters. Presenters are media-actors specialised to render media objects. Figure 1 portrays a multimedia system concerned with the remote and interactive playback on-demand of multimedia presentations (e.g., a teleteaching session composed of synchronised video and audio) over the Internet MBone.

The Transmitter and Receiver(s) are connected by two bindings: data streaming and control streaming. The former carries the data of the multimedia session according to the RTP/RTCP protocol. The latter makes the interactive control commands flow according to the RTSP protocol. In case the data streaming binding is multicast, receiver subsystems can arbitrarily join the on-going multimedia session requested by its initiator. Normally, only the initiator has the rights to control the session by the streaming control binding. In order to allow a group of receivers to perform

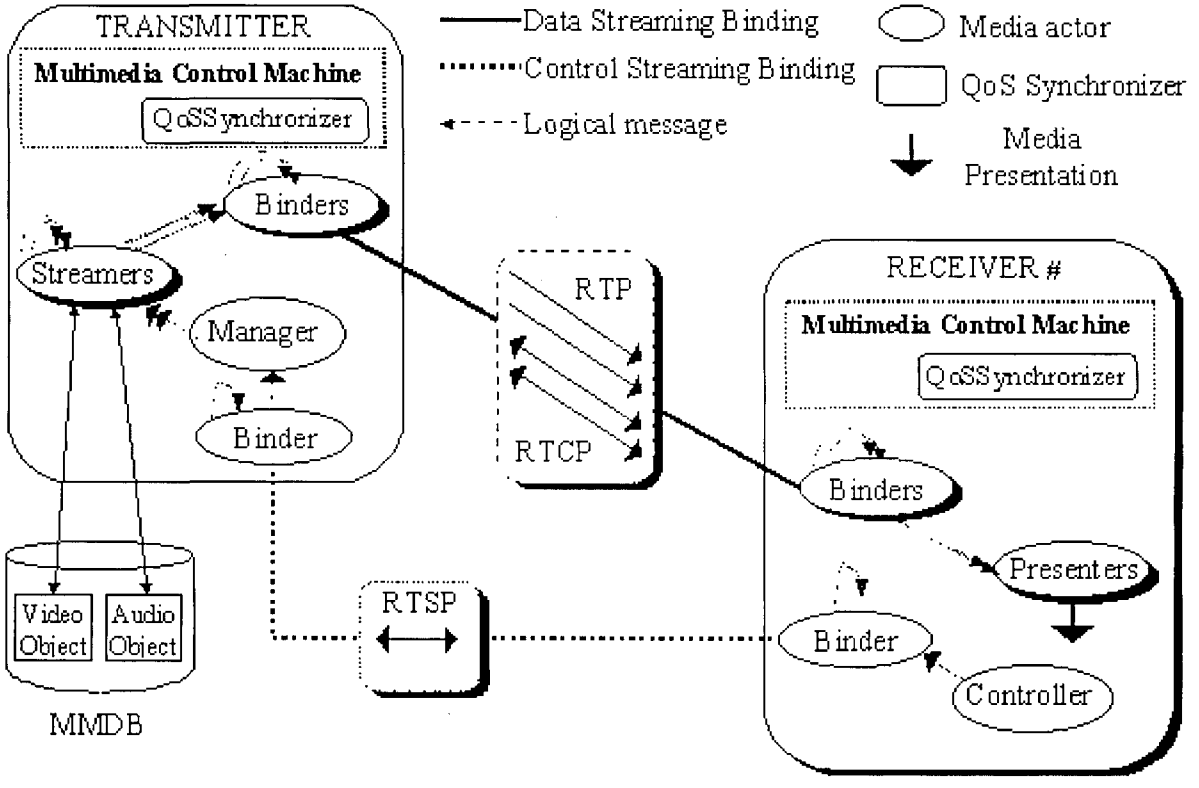


Figure 1: Playback on-demand of a multimedia session composed of two media stream(v/a)

control actions on the data binding, a floor control policy should be introduced.

Transmitter subsystem is responsible of the reading process and the enforcement (by using the RTP timestamps) of timing constraints upon the media streams to fulfil the requirements of the multimedia presentation. In addition, it responds to commands (e.g., play, pause) from the receiver site through the control streaming binding so as to perform an interactive service. On the remote site, Receiver(s) subsystem(s) control and render the requested multimedia session. The video and audio objects are RTP files previously recorded and archived. The multimedia session is described by the Session Description Protocol (SDP). The presentation description contains information about the media streams within the presentation, such as the set of encodings, network addresses, inter-stream synchronisation relations and information about the content. At the transmitter site, an actor pair, Streamer and Binder, is instantiated for each media stream. The Streamers read the media files (audio, video) and send them, as messages, to the Binders. At the receiver site, a mirrored situation exists. The Binders poll the bindings and deliver the read messages to the Presenter for rendering purposes.

2.2.2 Multiple Sessions and QoS Broker

Multiple session support is the responsibility of a system-agent called a Broker (or QoSBroker) (Steinmetz & Nahrstedt 1995). It acts as a coordinator for all the on-going sessions and performs admission control for new incoming ones. More specifically, the QoSBroker manipulates resources at the end-points, coordinating resource management across layer boundaries. The various components of a multimedia system can logically be organised according to four different abstraction layers: media, computational, timing, brokering. Media layer encompasses passive objects that model media information sources such as media files (e.g., RTP, AVI, MOV, AU), media devices (e.g., video capture boards, sound boards) and network streams. Computational layer consists of media-actors such as Binders, Streamers, Managers, Presenters. Media-actor Controllers are introduced to handle events generated by the user through a graphical interface. The QoSsynchronizers are located in the timing layer. They manipulate timing parameters such as jitter intra-stream and skew inter-stream. The brokering layer have to do with filtering of user QoS requirements, admission control of multiple sessions and resource management. The system-agent QoSBroker performs the mentioned tasks.

3 The ViCRO System

ViCRO is a distributed system over the Internet MBone and WWW for collaborative recording, playback and browsing of multimedia sessions. It provides VoD interactive presentation, cooperative playback and recording. The users of an interactive presentation should have control at least over

the following "degree of customisation": the time when the presentation starts, the order in which the various information items are presented, the speed at which they are displayed, the form of presentation. Since the ViCRO system lives over MBone, the following are some additional degrees of customisation: the location in the network where the recording or playback takes place, the scope for recordings and playbacks (e.g., local, global), the transmission mode for a playback (unicast or multicast), the users or groups that may access a recorded session and the modality of access (e.g., public, private with keyword). The system leverages the evolving infrastructure of the World-Wide-Web and also provides it with a major missing ingredient, namely interoperable continuous media services. It is based on Internet multimedia protocols. This section presents the system design goals, the system architecture (Figure 2), and the media client/server interactions.

3.1 Design Goals

The basic design goals of ViCRO are:

- interactive distributed client-server architecture where multiple clients can access and request recording and playback services from multiple servers;
- Internet standard-based multimedia communication protocols;
- off-line playing and recording services where users can schedule requests to be fulfilled at later point in time (e.g., a student programs the media server to record a teleteaching session at 12.30 of the day after; a teacher, after recording a lesson, schedules multiple playbacks for the next days, etc.);
- near-on demand features (staggered playbacks);
- floor control mechanisms for collaborative joint work (cooperative control of a playback session);
- multi-user capabilities with personalised access to media resources;
- platform-independency.

3.2 System Architecture

The system architecture (Figure 2) consists of Media Servers, Media Clients and (optionally) Web Servers that are the basic building blocks in the large. RTSP is employed as the service access and stream control protocol.

3.2.1 Media Server

The media server is the network entity that provides playback, recording, and browsing services for multimedia sessions. It consists of an RTSP Server, Recorders, Players and Session Directory interface (Fortino 1997).

The service entry point is the **RTSP Server** which allows a media client to request a service according to the RTSP protocol. It is composed of a Manager and Front-Ends. The former performs load monitoring and admission control. For each connection, the Manager spawns a Front-End thread which directly dialogs with the media client. The Front-End starts, manages and terminates the service agent (or servent, see below) under the media client's control.

The servent **Recorder** can record IP-packets on UDP and RTP level. Recording on RTP-level means that it parses the RTP-header of each incoming packet, checks for duplicates and out-of-order packets, synchronises them, and stores them on the local file system. It records data arriving both in unicast and in multicast way. Each recorded data-stream is stored in two files, a data-file and an index-file. The data-file contains the raw RTP packets. The index-file provides indexing on the data-file to make random and fast access to the data. A third file called option-file can be used to store important points or markers in the multimedia session both during a registration and during successive playbacks.

The servent **Player** plays packets back according to the sender schema of the Architecture of a Single Multimedia Session (see section 2.2).

The **Session Directory** interface implements an SDR (Session DiRectory tool) using the Session Announcement Protocol (SAP) and the Session Description Protocol (SDP) to listen to and announce multimedia sessions over MBone. It delivers current session information at the Manager that updates the live archive of the Media Server.

3.2.2 Media Client

The media client is the network entity that requests continuous media data from the media server. It consists of the RTSP client, the QoS Filter (Fortino 1997, Fortino & Nigro 1998), a VCR Controller, and Media Presentation tools.

The **RTSP Client** implements the client part according to the RTSP protocol specification (Schulzrinne et al. 1998).

The **QoS Filter** is developed according to the receiver schema of the Architecture of a single Multimedia Session (see section 2.2) and was applied to the lip synchronisation problem (Fortino & Nigro 1998, Kouvelas et al. 1996). It pre-filters RTP packets so as to recover intra- and inter-stream synchronisation.

The **VCR Controller** allows the user to start, control (e.g., by issuing pause and seek commands), and tear down remote sessions.

The **Media Presentation** tools currently used are VIC for video and VAT for audio.

3.3 Media Client/Server Interaction

Media Server and Media Client interact to one another by exchanging two kind of streams: media data and con-

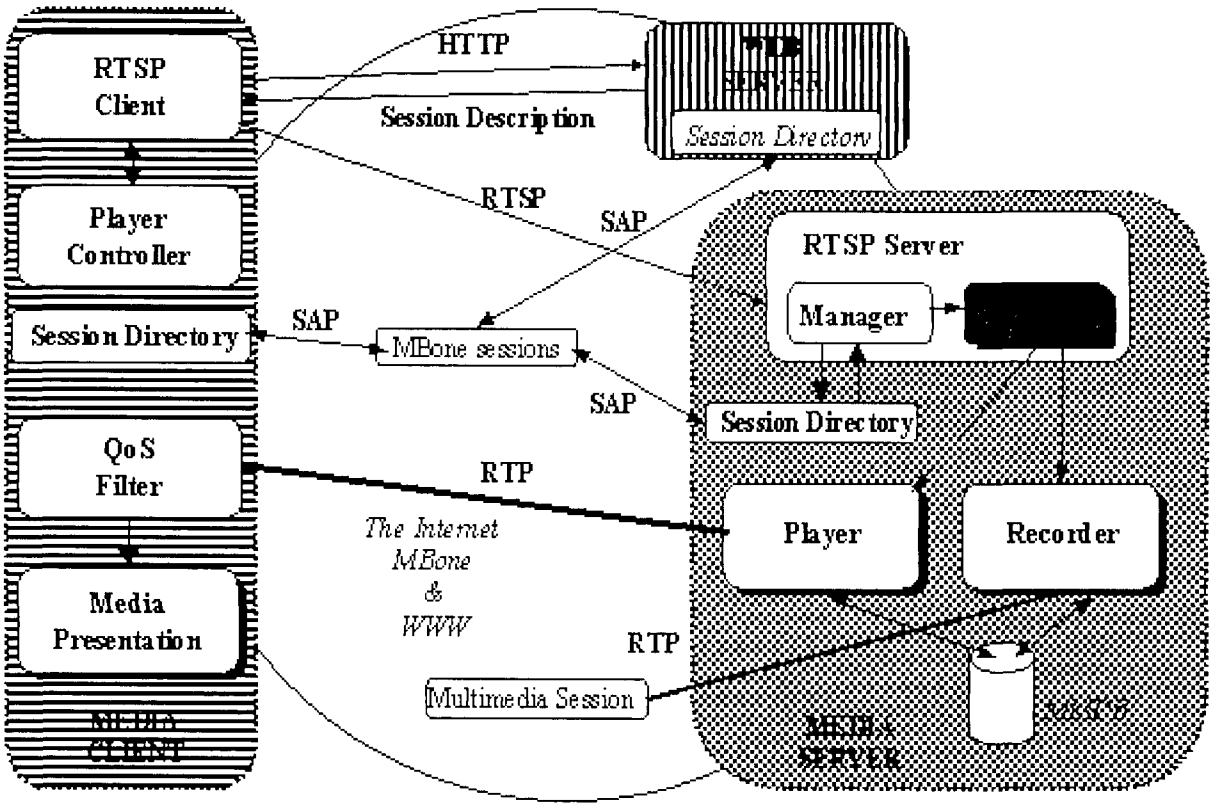


Figure 2: ViCRO system architecture

trol. Data Streaming is based on the RTP protocol. Media Streaming Control is based on the RTSP protocol. RTSP is a text-based application level protocol developed for on-demand delivery control of media streams both live and pre-recorded. The control stream carries information about set-up (e.g., multicast addresses, media-type) and control operations (e.g., play, pause) of the multimedia session being sent from the Media Server to the Media Client/s under the form of one or more media streams (e.g., audio, video). The media stream can be sent both in unicast (VoD) and multicast way (Near VoD, MBone model). In Figure 3 is portrayed an example of media C/S interaction during a playback session.

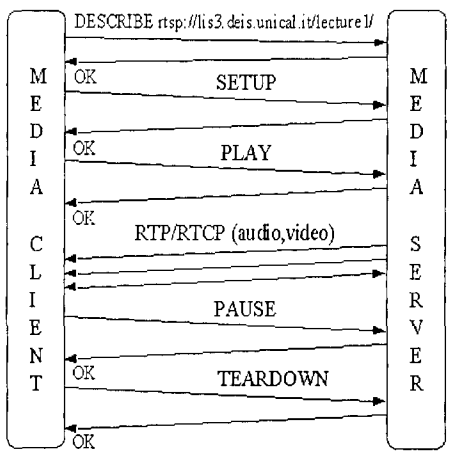


Figure 3: RTSP playback session example

The media client that has already picked up a formatted description (SDP, RTSL, etc.) of the session (e.g., via RTSP describe method or WEB server) issues a Setup message. Setup causes the server to allocate resources for a stream and start an RTSP session. In this phase the media client proposes a unicast (or multicast) <address, port> pair for each media within a presentation. The media server receives the request and processes it. If the media client proposal is unacceptable to the media server, perhaps because the address is already in use in a different session, the media server replies with a different address. This negotiation continues until the media client and the media server agree on the media <address, port> pair of the session. Afterwards, the media client can send the message Play to start the data transmission of the presentation allocated via Setup. The message Pause temporarily halts a stream without freeing the media server resources. The message "Play range:npt | smpte | clock= ts-te" performs seeks, delayed playbacks, etc. That is, it positions the normal play time to the beginning of the temporal range (normal, relative and absolute) specified and delivers data till the end of the range is reached. The Teardown message closes the multimedia session and frees resources, on the media server, associated with the session streams.

3.3.1 Cooperative Playback of Sessions

In the context of a media-on demand system, interaction among clients means both sharing of the view and control of a multimedia session playback, and collaboration (i.e., questioning and book-marking) on its contents. A set of interacting clients, in a sense given above, is called "explicit" cooperative group. It has the property to be tightly-coupled and strictly-collaborative. Using a VCR metaphor, each participant has a remote control, which operates on a single and shared playback. If one group member performs a seek/pause operation, that operation must be propagated, according to certain rules, to all other members of the group. Besides, they can collaborate with each other by marking specific instants within a session (e.g., for question proposals) and questioning through a question-board. In order to hand off control among collaborators and enrich their inter-communication semantics, global coordination protocols that manage resources or resolve conflicts between different users' actions are needed. Thus, control policies (e.g., moderated playback control, floor or token based and lock control schemes, or laissez-faire cooperative control) have to be implemented. In a floor-controlled cooperative session, a member can issue a command (e.g., a pause) only if he/she is entitled to do so according to the adopted control policy, i.e., if he/she has got the floor. The other members should have the same view as the one that raised the command. It is the concept of "What You See Is What I See" (WYSIWIS).

On the other hand, a group is implicit or loosely-coupled when independent clients simply happen to be viewing the same playback session at the same point in time. The goal of those clients is not to collaborate with other clients. Servers may coalesce these clients into a single group to improve service efficiency and scalability (Almeroth & Ammar 1998). If any of the members of such an implicit group performs a pause/seek operation, the group is automatically split up.

In order to support explicit cooperative groups, an interactive and cooperative playback system should provide at least the following functionality:

- Group Organisation (formation and management);
- Shared State Maintenance (streaming control protocol);
- Fault Tolerance (detection and recovery);
- Cooperative Work (message exchanging);
- Control Policies.

How the media C/S interaction is designed and implemented has a deep impact on the performance of the system. The use of multicast vs. unicast can boost efficiency and scalability.

Three different approaches have been envisaged and are being developed in ViCRO system:

- Unicast or Multiconnected. Each media client involved in the cooperative session establishes a separate connection (unicast) with the media server. The media server works as a router. The solution is expensive, i.e., the server must maintain per-client connection states, and not scalable, i.e., the server must stay connected to the clients and update them, if any state changes, all session duration long;

- Hybrid or Reflected. Only a media client originator of the collaborative playback session connects to a media server. The other members of the explicit group communicate with the originator on a multicast channel and with the media server through the originator. The originator performs routing and filtering of control messages. He/she is a sort of reflector. The solution scales well but has a weak point: the unique connection between the originator and the media server. If this connection fails all the other media clients lose control on the playback session;

- Fully Multicast. The archive service is multicast-enabled (Shuett et al. 1998). The media server sets up a multicast address allowing media clients to share a control session. That is, both media server and media clients send and receive messages on a common multicast control channel. This solution doesn't suffer the previous drawbacks and so it is more robust, cost-effective and scalable.

The multiconnected approach can be based on the RTSP protocol with no substantial changes.

The reflected one can be based on the RTSP for the interaction media server / media client originator and on the LRMP for allowing the other media clients to send control messages. RTSP has been enhanced in this work in order to support fault-tolerance. In particular, two new methods were introduced: PASS and CONTINUE. The PASS method is issued by the media client originator in the case he/she wants to pass the session control to another client and leaves. When the server receives the PASS request, it disconnects the originator and waits for a new connection by freezing the thread which was handling the interrupted connection. The thread continues to stay alive till a timeout expires. The CONTINUE method is sent from the new originator to take control of the session. When the server receives the CONTINUE, it connects the new originator to the frozen thread. If the originator fails (e.g., leaving without sending the PASS, so losing the connection with the server), the server behaves the same as it receives a PASS method.

The fully multicast approach can use RTSP on top of LRMP. Obviously, RTSP, designed for unicast connection, has to be upgraded in order to be enabled for multicast semantics.

4 Application Scenarios

4.1 Tele-Teaching and Cooperative Learning

In this scenario, recorded lessons, seminars and talks can be transmitted on a regular basis over an "ad-hoc" MBone (e.g., University, Institute, private network). The instructors can use ViCRO to augment their instructional materials by mixing live and recorded data. Students can remotely access archived lessons by connecting to ViCRO services. They can choose and navigate a particular lesson. The system allows students to work in groups where members cooperatively watch a teleteaching session, feeding back each other with questions and hints.

4.2 Videoconferencing

In a typical videoconferencing scenario, users send live audio/video streams from their desktops to the other conference participants to discuss particular problems. On-line videoconferencing can be greatly enhanced if users are enabled to access archived material (e.g., slides, video clips) and bring it into the on-going conference. ViCRO also favours the off-line videoconferencing. In this scenario, a user that wants to participate to a videoconference being transmitted from the USA over MBone could not be able to do so because he/she is in a different timezone (e.g., Europe) or lacks direct access to the MBone. In this case, recording the videoconference and sending it at a later point in time would resolve the problem. Optionally this retransmission could be sent to a different multicast address with a different scope (i.e., local Time To Live) or to a unicast address. In order to maximise the quality of the recorded session it would be convenient to connect to a media server, if any, as close to the sender site as possible.

5 Related Work

5.1 mMOD

The multicast Media-on-Demand system (Parnes et al. 1997), mMOD, developed at the Luleă(University of Technology, is a system for recording and playing back MBone sessions. mMOD can be controlled by using a command-line interface or a WWW based interface. The system consists of three separate parts, the VCR, the data translater and the Web-controller. The VCR is a stand-alone program for recording and playing back IP-packets on either UDP or RTP level. The Data-Translater translates the traffic in various ways (recoding, mixing and switching techniques) to allow users with different bandwidth to access the service. The Web-controller is a program that acts as a Web-interface of mMOD. It is through this interface that a new session can be started and controlled and information about running sessions viewed so that a user can join them. mMOD is completely written in Java 1.1. However, it doesn't support remote, interactive recording and isn't based on an open and standard protocol for the media client/server interaction.

5.2 MVoD

MBone Video Conference Recording on Demand (Holfelder 1997), MVoD, which is being developed at the University of Mannheim, is a client-server based architecture for interactive remote recording and playback of MBone sessions. It is based on open standards (e.g., CORBA), making it possible for other applications to interface it. The MVoD service consists of three basic components: the Manager and the VideoPump, that form the logical unit called MVoD Server, and the MVoD Client. The interaction between the system components is regulated by four protocols: the VCR Service Access Protocol (VCRSAP), the VCR Stream Control Protocol (VCRSCP), the VCR Announcement Protocol (VCRSAP), and the VCR Client Message Protocol (VCRCMP). The Manager and the Client are implemented in Java. The VideoPump is implemented in C++. The Client can be started either as a Java applet within a Java-enabled browser, or as a stand-alone Java application. The interfaces of the MVoD service are specified by using the standardised Interface Definition Language (IDL). Though the system is very well designed, the use of proprietary protocols limits the system openness obtained by CORBA and the implementation in C++ of the VideoPump reduces its portability.

5.3 MARS

The MASH Remote Player (Shuett et al.1998), MARS, designed at UCB (University of California, Berkeley), is a client/server system for remote media browsing. The media server is implemented by using the TACC (Transformation, Aggregation, Customisation and Control) toolkit. RTSP is used to control real-time media streams. Mechanisms of advertising and discovering new contents are implemented such as rich-description hyperlinks to the archived sessions and automatic detection of significant instants during a session. The client part consists of the MASH streaming player that can operate as media browser or as helper application within a Web browser. It allows the user to bookmark specific sessions and specific instants within a session. The bookmark file can be shared among several participants. The client is implemented in C++ and TCL/TK. To date, the system doesn't provide a remote recording service.

5.4 IMJ

The Interactive Multimedia Jukebox (Almeroth & Ammar 1998), IMJ, is a research effort which is exploring the viability of grouping users together into implicit group. The IMJ system uses the WEB as a means to gather requests

and present play schedules. Users may request playback, but have no means to interact with the server to control the resulting schedule, cancel playback or perform seek or pause operations. Control is performed only on buffered, per-client replicated data of the session.

6 Implementation Status

The implementation of the ViCRO system is based on the Java Multimedia Studio tool (Fortino 1997) developed at the International Computer Science Institute. The Player, Recorder, QoS Filter, Session Directory interface and multimedia protocols (RTP/RTCP, LRMP, SAP, SDP) had been already implemented and tested over MBone. The RTSP protocol has been developed according to the latest version (Schulzrinne et al. 1998). In addition, RTSP has been adapted to run on top of LRMP. Currently, the RTSP client is implemented as both a standalone application and a helper application within a Web browser. The main advantage to invoking the client as a helper application, instead of a Java applet, is that it can open multicast channels on behalf of the user. The media presentation tools are VIC for the video and VAT for the audio. JMF (Java Media Framework) is also under experimentation. The mechanisms supporting collaborative playbacks are being developed.

In Figure 4 the GUI of the RTSP Client (VCR Controller and RTSP Browser) is portrayed. The components of the RTSP Client are divided into 5 areas: Menubar, VCR Controller, State Field, URL Field, Session ID Label. **Menubar** consists of File, Session, and Bookmarks Menu.

The menu File allows to open and close recording and/or playback sessions with a chosen media server. The menu Session makes it available options for retrieving a session description form (remote, user, local), for obtaining info about the on-going sessions, etc. Through Bookmarks, the user can bookmark RTSP URLs. **State Field** displays the current state (INIT, READY, PLAYING) of the session. URL Field portrays the **RTSP URL** of the on-going session. **Session ID Label** highlights the session identifier from the Client side. **VCR Controller** is composed of buttons (START, PLAY, PAUSE, STOP, DISCONNECT) to control the session, a temporal slide bar to randomly seek, and a display reporting the Total time and the Elapsed time. DISCONNECT breaks the C/S connection without freeing resources at server site (see the RTSP PASS method in section 3.3).

7 Conclusions

This paper presents the ViCRO videoconference recording on-demand system which permits to playback, record and browse multimedia sessions over MBone. The system has been tested on top of Windows95, NT, and Solaris. It has been successfully used for playing teleteaching sessions and recording multimedia sessions over MBone. Prosecution of the research aims at:

- completing and testing the cooperative playback features in a collaborative teleteaching scenario;
- decentralising the components of a Media Server on a cluster to improve efficiency, fault-tolerance and service availability;
- comparing performance with the MoD systems described in §5.

8 Acknowledgments

The authors are grateful to Andres Albanese and Vincenzo Ciminelli for their contribution to the described multimedia system. Work partially funded by the Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST) in the framework of the Project "Methodologies and Tools of High Performance Systems for Distributed Applications (MOSAICO)".

References

[1] Agha G. (1986) *Actors: A model for concurrent computation in distributed systems.* MIT Press.

[2] Almeroth K.C. & Ammar M.H. (1998) The Interactive Multimedia Jukebox (IMJ): A new Paradigm for the On-Demand Delivery of Audio/Video. *Proceedings of the Seventh International World Wide Web Conference (WWW7).*

[3] Fortino G. (1997) Java Multimedia Studio v1.0. *ICSI Technical Report*, No. TR-97-043.

[4] Fortino G. & Nigro L. (1998) QoS centred Java and actor based framework for real/virtual teleconferences. *Proceedings of SCS EuroMedia 98.* Leicester, UK, p. 129-133.

[5] Fortino G., Nigro L. & Albanese A. (1998) A Java Middleware for the development of actor-based multimedia systems over Internet MBone. *Proceedings of ISAS'98.* Orlando (FL), USA.

[6] Handley M. (1997) SAP: Session Announcement Protocol. *Internet Draft, IETF, Multiparty Multimedia Session Control Working Group.* Work in progress.

[7] Handley M. & Jacobson V. (1998) SDP: Session Description Protocol. *Request for Comments (Proposed Standard) 2327, Internet Engineering Task Force.*

[8] Holfelder W. (1997) Interactive remote recording and playback of multicast videoconferences. *Proceedings of the 4th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97).* Darmstadt, Germany.

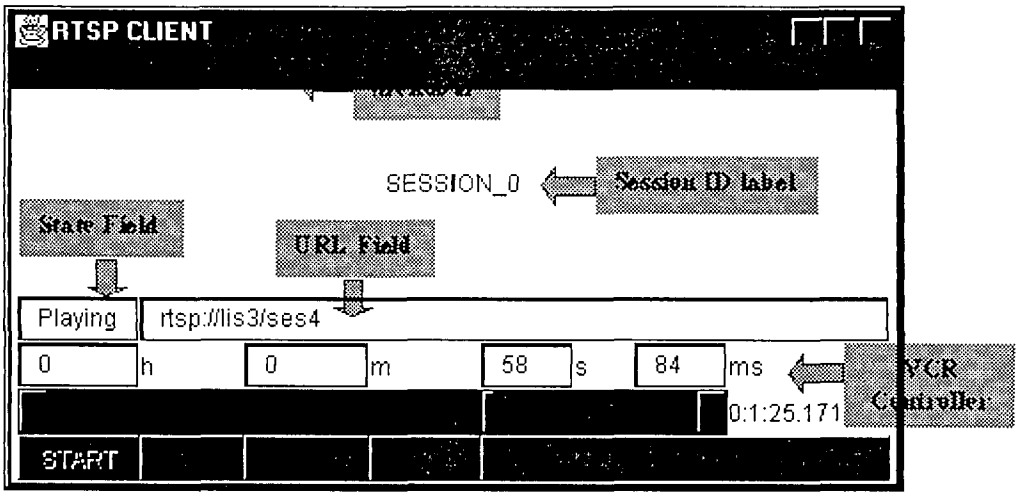


Figure 4: RTSP Client's GUI during a playback session

[9] Kouvelas I., Hardman V. & Watson A. (1996) Lip Synchronisation for use over the Internet: Analysis and Implementation. *Proceedings of IEEE Globecom'96*. London, UK.

[10] Kumar V. (1998) *Mbone: Multicast Multimedia for the Internet*. McMillan Technology Series.

[11] Liao T. (1997) Lightweight Reliable Multicast Protocol version 1. *http://webcanal.inria.fr/lrmp/*.

[12] Nigro L. & Pupo F. (1998) A modular approach to real-time programming using actors and Java. *Control Engineering Practice*, 6, 12, p. 1485-1491.

[13] Parnes P., Mattsson M., Synnes K. & Schefstrom D. (1998) mMOD: the multicast Media-On-Demand system. *http://www.cdt.luth.se/peppar/progs/mMOD/*.

[14] Ren S., Venkatasubramanian N. & Agha G. (1997) Formalising multimedia QoS constraints using actors. *Formal Methods for Open Object-based Distributed Systems*, 2, Bowman H. and Derrik J. (eds), Chapman & Hall, 139-153.

[15] Rowe L. A., Berger D. A. & Baldeschwieler J. E. (1995) The Berkeley Distributed Video-on-Demand System. *Proceedings of the Sixth NEC Research Symposium*.

[16] Schulzrinne H. (1997) A comprehensive multimedia control architecture for the Internet. *Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*. St. Louis, Missouri.

[17] Schulzrinne H., Casner S., Frederick R. & Jacobson V. (1996) RTP: a Transport Protocol for Real-Time Applications. *Request for Comments (Proposed Standard) 1889, Internet Engineering Task Force*.

[18] Schulzrinne H., Rao A. & Lanphier R. (1998) Real Time Streaming Protocol (RTSP). *Request for Comments (Proposed Standard) 2326, Internet Engineering Task Force*.

[19] Shuett A., Raman S., Chawathe Y., McCanne S. & Katz R. (1998) A Soft State Protocol for Accessing Multimedia Archives. *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV98)*. Cambridge, UK.

[20] Steinmetz R. & Nahrstedt K. (1995) *Multimedia: computing, communications and applications*. Prentice-Hall.

Computing Multidimensional Aggregates in Parallel

Weifa Liang
Department of Computer Science
The Australian National University
Canberra, ACT 0200, Australia
E-mail: wliang@cs.anu.edu.au
AND
Maria E. Orlowska
Distributed Systems Technology Centre
Department of Computer Science and Electrical Engineering
The University of Queensland
St. Lucia, QLD 4072, Australia
E-mail: maria@csee.uq.edu.au

Keywords: data cube, parallel algorithms, OLAP, aggregation computation, data warehousing

Edited by: Rudi Murn
Received: February 5, 1999 **Revised:** September 14, 1999 **Accepted:** October 29, 1999

Computing multiple related group-bys and aggregates is one of the core operations of On-Line Analytical Processing (OLAP) applications. This kind of computation involves a huge volume of data operations (megabytes or treabytes). The response time for such applications is crucial, so, using parallel processing techniques to handle such computation is inevitable. In this paper we present several parallel algorithms for computing a collection of group-by aggregations based on a multiprocessor system with sharing disks. We focus on a special case of the aggregation problem–"Cube" operator which computes group-by aggregations over all possible combinations of a list of attributes. The proposed algorithms introduce a novel processor scheduling policy and a non-trivial decomposition approach for the problem in the parallel environment. Particularly, we believe the proposed hybrid algorithm has the best performance potential among the four proposed algorithms. All the proposed algorithms are scalable.

1 Introduction

Aggregation is a predominant operation in decision support database systems. On-Line Analytical Processing (OLAP) often needs to summarize data at various levels of detail and on various combinations of attributes. Recently [8] introduced the *data cube* operator for the convenient support of such aggregates in OLAP. In technical terms, the data cube is a redundant multidimensional projection of a relation on subsets of its schema. It computes all possible group-by SQL operators and aggregates their results into a n-dimensional space for answering OLAP queries. In general, if we consider an n-attribute relation R, there are 2^n group-by aggregations. In this paper we refer to the computation of the 2^n group-by aggregations as the *data cube* computation of R. [10] suggests a search lattice structure for computing the data cube. The lattice is a directed acyclic graph (DAG) $\mathcal{L} = (V, E)$, where each node $v \in V$ represents a group-by of the data cube. Each directed edge $\langle u, v\rangle \in E$ from group-by u to group-by v illustrates that v can be generated from u. u is called the *parent* of v, also v has exactly one attribute less than u. Thus, the outgoing degree of any node with d attributes is d. Let n be the number of attributes in a base relation R and $l \leq n$. Define

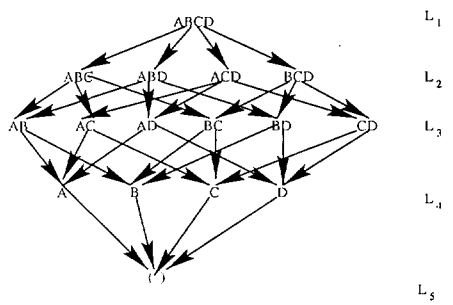


Figure 1: A lattice for the cube based on R.

level l of the search lattice as the level where all group-bys contain exactly $n - l + 1$ attributes. Each group-by v in the lattice is also called a *cuboid* in the data cube. The size of a cuboid is the number of tuples in a relation defined by the corresponding projection. In this paper we will use the terms *node*, *group-by*, or *cuboid* interchangeably. Figure 1 is an example of a search lattice of a relation R with four attributes A, B, C and D, and each node is labeled by its attribute subset. There are $2^4 = 16$ different group-by aggregations for R, and each group-by corresponds to a table derived from R.

The response time is a primary concern in OLAP applications [3]. Interactive analysis (response time in seconds) is hard to achieve if each of the aggregates is computed "on the fly" at query execution time. Therefore, most solutions for the OLAP application precompute the aggregates and run interactive sessions against the precomputed data. It must be mentioned that efficiency is also important for the precomputation time, since this is a lower bound on the recency of the data available to analysts and decision makers. Furthermore, the cost of precomputation influences how frequently the precomputed data is brought up-to-date.

There are two fundamental problems related to the cube operator. One is related to the efficient computation of all dimensional aggregates (group-bys). Another focuses on the selection of a subset of aggregates for materialization in order to reduce the response time of the queries. This occurs when it is impossible to store all aggregates in a data warehouse, due to a limited storage space and other constraints. In this paper we shall focus on the efficiency issues by presenting parallel computational solutions. Following [1, 4, 14], we assume that the aggregating functions are *distributive* [8]. Examples of distributive functions include `max`, `min`, `count` and sum, etc.

1.1 Related work

Basically there are two approaches to compute a group-by: (1) the sort-based method and (2) the hash-based method [7]. Methods of computing single group-bys in both sequential and parallel environments have been well studied (see [7] for a survey, also see [15]). However, little work has been done on optimizing the total operations for a collection of related aggregates. The problem was first introduced by [8]. In their seminal paper they give some rules of thumb to be used for an efficient implementation of the data cube operator. These include the `smallest parent optimization` (a node is generated from a parent who has the minimum number of tuples), and the partitioning of data by attribute values. Since then, several works have been carried out along this direction. For example, [1, 14] embed five optimization strategies: `smallest-parent`, `cache-results`, `amortize-scans`, `share-sorts` and `share-partitions`, into a single algorithmic framework, and suggest algorithms `pipesort` and `pipehash` (based on sort-related and hash-related grouping methods) to implement the cube operator. [1, 4] give overlap algorithms for the data cube computation. Their algorithms seek to minimize the number of sorting steps required to compute many sub-aggregates that comprise a multidimensional aggregate, and make efficient use of the available memory to reduce the number of I/Os, i.e, by overlapping the computation of the various cuboids, this thereby reduces the number of sorting steps required. [13] considers the sparse relation by giving a recursive algorithm for the cube operator. Their solution is based on two fundamental ideas that have been successfully used for performing complex operations (e.g. sorting and joins) over large relations: (1) partition the large relations into fragments that fit in memory; and (2) perform the complex operation over each memory-sized fragment independently. [17] gives an algorithm for computing the cube by using a multidimensional array rather than a table and compression techniques. All of the above mentioned papers only consider computing all dimensional group-by aggregations of R, while [10, 9] present algorithms for deciding what group-bys to precompute and index if not all aggregates are needed. [12] presents efficient algorithms for the online maintenance of data cubes.

1.2 Contributions

So far all known algorithms for computing cubes are designed for the sequential uniprocessor computing environment. As it is well understood, the cube computation usually involves large data operations. Since the response time for such computation is crucial for the successful deployment of OLAPs, it seems that applying parallel processing techniques to this problem is interesting possibility. In [8], they mention the possibility of applying parallelism for the data cube computation. However, they do not exploit this concept. In this paper we shall propose several parallel algorithms to compute the data cube efficiently, based on a multiprocessor system with sharing disks. The proposed algorithms introduce a novel processor scheduling policy and a non-trivial decomposition approach for the problem in the parallel environment. We shall first introduce in detail a naive algorithm for the problem and then present an optimization algorithm for it based on the cost estimation using the minimum spanning tree (MST) method. We shall also give another optimization algorithm for the problem based on the cost estimation using the minimum weighted matching method. We shall finally propose a hybrid algorithm which makes use of the advantages of the previous two optimization algorithms. We believe that the hybrid algorithm has the best performance potential among the four algorithms introduced in this paper. All the proposed algorithms are scalable.

The rest of the paper is organized as follows. In Section 2 we shall introduce the parallel computational model, and then present some results for the single group-by aggregation in the parallel computational environment. In Section 3 we give three parallel algorithms for the cube operator, while in Section 4 we present a hybrid parallel algorithm for the problem by incorporating the two algorithms discussed in Section 3. In Section 5 we extend our result to a generalization in which we compute only some group-by aggregates in a data cube by optimizing the total operations on these aggregates. We conclude our discussion in Section 6.

2 Preliminaries

2.1 The parallel computational model

We assume that a multiprocessor system consists of p processors, and each processor has its own local main memory. All processors are assumed to be identical and connected with an array of disks which is used to store the initial data (the base relation table) and the final data (all dimensional group-by aggregations) through a high-speed interconnection network. We further assume that the sizes of the base relation table R and these group-bys are very large and cannot fit in the main memory.

2.2 Parallel algorithms for a single aggregate

There are a number of parallel algorithms for processing a single aggregate [2, 7, 15]. The generic framework for this kind of processing consists of two phases. In the first phase each processor executes aggregation on its local partition of the relation. In the second phase, all processors merge these partial results to produce the final result. For example, [2] discusses two sorting-based algorithms for aggregate processing on a shared disk cache architecture. [7] gives an optimization for dealing with bucket overflow for the two phase algorithm based on the hash-joining method. Recently, [15] considers a range of grouping selectivities by presenting an adaptive parallel algorithm for the problem.

Now let us consider a single aggregate v on a table R'. Assume that there are p' processors available for computing v. >From the proposed algorithms above, we can derive that the time t_v used for computing v is inversely proportional to the number of processors used, i.e., the more processors used, the less time is needed. In general, t_v can be expressed as follows.

$$t_v = a \cdot |R'|/p' + b \cdot p' \quad (1)$$

where $|R'|$ is the number of tuples in R', a and b are the parameters dependent on the computing environment in which a is related to the time for read/write an I/O page and b is related to the communications overhead among the p' processors. In this paper we assume that the parallel algorithm for the single aggregate v is available [15], and refer to it as `Aggregate_multiprocessor(`v, p'`)`.

3 Parallel Algorithms

In this section we introduce three parallel algorithms for the efficient computation of data cubes, and provide an analysis of these algorithms in terms of the response time.

3.1 A naive parallel algorithm

Without loss of generality, let us assume that the base relation table R has n attributes and $|R|$ tuples. Assume that the multiprocessor system consists of p processors, and each processor P_i is responsible for a horizontal fragment R_i of R, where $R_i \cap R_j = \emptyset$, $\bigcup_{i=1}^{p} R_i = R$ and $|R_i| \approx |R|/p$, $i \neq j$ and $1 \leq i, j \leq p$.

The algorithm presented below is rather natural and consists of two phases. In the first phase, each processor computes the data cube of R_i, and stores the data cube back to the disks. Note that all processors execute their tasks independently. In the second stage, we merge the data cubes of the fragments of R to form the original data cube of R through a number of rounds of parallel merging. The detailed algorithm is described as follows.

Algorithm 1:

1. Horizontally partition R into p equal parts roughly. Let processor P_i be responsible for the fragment R_i of R.
2. **for** each processor $P_i : 1 \leq i \leq p$ **pardo** compute the data cube of R_i and write the results back to the disks using an efficient sequential algorithm **endfor**;
3. **for** $k := 1$ **to** $\lceil \log p \rceil$ **do**
   Let G_i be a cluster of processors indexed from $2^k(i-1)+1$ to $i2^k$.
   for each cluster $i : 0 \leq i \leq p/2^k$ **pardo**
   processors in G_i merge the data cubes obtained by them in the previous iteration and write the results back to the disks
   endfor

endfor.

Now, let us analyze the response time of the proposed algorithm. The time complexity T_p in Step 2 is approximately $1/p$ of the sequential time if carrying such a computation by a uniprocessor. This is because each processor only handles a fragment whose size is about $1/p$ of the size of the base relation table. Before running Step 3, there must be a synchronization at the end of Step 2. Now we consider the implementation of Step 3, assuming that every group-by has been sorted already during the first stage. Then, we merge these group-bys involved. So, the total of $\lceil \log p \rceil$ rounds is necessary. Since we assume that there is a high-speed interconnection network available, the read/write conflicts from and to the disks between different processors are negligible and ignored in our consideration afterwards.

`Algorithm 1` has some drawbacks when the data distribution on some dimensional aggregate is heavily skewed. The reason is that the synchronization at the end of Step 2 takes $\Delta = t' - t$ unit times, where $t = \min_{1 \leq i \leq p}\{t_i\}$ is the earliest finishing time by some processor and $t' = \max_{1 \leq i \leq p}\{t_i\}$ is the latest finishing time by some processor, t_i is the time of processor P_i for computing the data cube of R_i. As a result, the more skew the data distribution among processors is, the larger the Δ becomes. Meanwhile, the synchronization is also required between different rounds in the second stage.

In order to reduce the synchronization time for skewed data distribution, we propose the following approach. That is, when we compute a group-by aggregation or a merging (in the second stage), we first run a sub-routine to check whether the data distribution among the processors is skewed. If this is the case, we first re-distribute the data among the processors in order to make the processors involved are well load balanced. We then run through the steps of `Algorithm 1`. Certainly this will lead to a better response time for this case. However the load-balancing computation and data re-distribution in a multiprocessor system involve communications and data transfer among the processors which are also time consuming. Therefore, there must be a trade-off in terms of overheads between synchronization and load-balance.

3.2 An MST-based parallel algorithm

`Algorithm 1` gives some optimization for the cube operator in the total operations required for each fragment. But this local optimization is not necessarily the global optimization. In this section we introduce our second parallel algorithm `Algorithm 2` which in some sense presents global optimization in terms of the total operations for the cube operator of R. Actually, `Algorithm 2` is a parallelization of the smallest parent method [8].

Basically, the algorithm consists of two phases. In phase one we produce a scheduling for the cube computation by establishing an estimated cost tree which will be defined later. And in the second phase we assign processors to the nodes in the tree and implement the cube computation.

As mentioned in the introduction, the cube computation can be implemented through building a search lattice. Now we assign a weight for each directed edge in the lattice which represents the computation cost of a node from its parent. We then produce such a directed minimum weighted spanning tree (also called optimal branching) in the lattice that each node has in-degree one except the root with in-degree zero. Then, the weighted sum of the edges in the tree represents the minimum total operations for the cube computation.

However, before the actual computation of each node (a group-by aggregate in the data cube) in the tree, there is no way of knowing its size exactly. Therefore, in practice we can randomly withdraw a small fraction of pages from the disk, which stores the base relation R_i as random samples, and use them to estimate the size of every node in the cube by using some efficient estimating methods (e.g. uniform distribution estimation). Thus, in the first phase, in fact, we build an *estimated* directed minimum spanning tree T for the lattice since the size of every node has an estimated value. The detailed procedure for this phase is introduced as follows.

Phase one: compute an estimated directed MST.

1. **for**each processor P_i: $1 \leq i \leq p$ **pardo**
   Let P_i be responsible for fragment R_i.
   P_i builds the lattice $\mathcal{L}_i$ for R_i and
   assigns the weights to the edges of $\mathcal{L}_i$.
   The weights are obtained by randomly
   fetching a small fraction of the pages of R_i
   from the disks and estimating the size
   $S_{i,k}$ of each node $\text{cuboid}_{i,k}$ in $\mathcal{L}_i$ [8].
   endfor;
2. Construct the lattice $\mathcal{L}$ of R
   and assign weights to its edges.
   The estimated size S_k of a node cuboid_k in $\mathcal{L}$
   thus is $\lceil \sum_{i=1}^{p} S_{i,k}/p \rceil$. Assign all outgoing
   edges from cuboid_k in $\mathcal{L}$ with weight S_k.
3. Find a directed minimum spanning tree T on $\mathcal{L}$.

Step 1 can be done easily because only a couple of pages are withdrawn from R_i to estimate the size of every node in $\mathcal{L}_i$. Step 2 involves the construction of $\mathcal{L}$ and the weight assignment of the edges in $\mathcal{L}$. Note that the size estimate of every node in $\mathcal{L}$ is now more accurate than it shows in Step 1 because p samples are used this time. Step 3 can be implemented as follows. For each node, except the root, it finds an incoming edge with the minimum weight, then another endpoint of the edge is the parent of the node in T. It is also straightforward to prove that the tree generated by this method is a directed minimum spanning tree for this special case.

In phase two we assign processors to the nodes in T such that the response time for computing all nodes in T is minimized, i.e., we aim to achieve the minimum response time for the data cube computation. However, how to assign processors to and how many processors are assigned to a node are very crucial to the entire response time. In this phase we present a processor allocation and deallocation scheduling procedure for the nodes in T.

Before we proceed, let us consider a node v in T. Assuming that each node in T has been assigned a weight which is the cost of generating the node. This assignment is carried out as follows. The root of T is assigned the cost for sorting its tuples by some order of all attributes. Now let $s(v)$ be the weight of the edge from the parent of v in T to v. We assign v by weight $s(v)$. T becomes a *node-weighted* tree afterwards.

Now we consider the processor allocation for T. Let W_T be the total estimated operations for the cube computation by T. Then,

$$W_T = \sum_{k=1}^{2^n} s(v_k). \quad (2)$$

>From equation 2, a lower bound of the response time for the cube computation based on T is obtained as follows.

$$T_{\min} = \lceil W/p \rceil. \quad (3)$$

Basically our processor allocation algorithm for T is based on the estimation of the time lower bound $T_{\min}$. The basic idea behind the algorithm is that the number of processors allocated for each subtree is proportional to the total operations needed to compute all nodes in that subtree.

Thus, all the subtrees whose roots share a parent can finish their computation in the same time approximately.

Recall that T is a node-weighted tree, and there are p processors available. In what follows, we first assign p processors to compute the root r of T. Let r have d children, $v_1, v_2, \ldots, v_d$, $d \leq n$, T_i the subtree rooted at v_i and W_{T_i} the weighted sum of the nodes in T_i, $1 \leq i \leq d$. It is obvious that v_i can be computed only after r (a table) is available, and the time lower bound for computing the nodes in these subtrees thus is

$$t_{\min} = \lceil \sum_{i=1}^{d} W_{T_i}/p \rceil. \quad (4)$$

Now we allocate the p processors to the subtrees. Assume that $W_{T_1} \geq W_{T_2} \geq \ldots \geq W_{T_d}$. We scan the subtrees by this order. Let T_i be the subtree which we are considering. If $\lfloor \frac{W_{T_i}}{t_{\min}} \rfloor \neq 0$, we assign $\lfloor \frac{W_{T_i}}{t_{\min}} \rfloor$ processors to T_i; otherwise, we merge T_i with T_{i+1} and repeat the above procedure until all subtrees have been assigned processors. In the end we claim that the number of processors used for this allocation is no more than the actual available number of processors. In the following, we show that this claim is true. Let p' be the actual number of processors used for this assignment, then,

$$\begin{aligned} p' &= \sum_{i=1}^{d} \lfloor \frac{W_{T_i}}{t_{\min}} \rfloor \leq \frac{\sum_{i=1}^{d} W_{T_i}}{t_{\min}} \\ &= \frac{\sum_{i=1}^{d} W_{T_i}}{\lceil \sum_{i=1}^{d} W_{T_i}/p \rceil} \leq \frac{\sum_{i=1}^{d} W_{T_i}}{\sum_{i=1}^{d} W_{T_i}/p} = p \quad (5) \end{aligned}$$

We then turn to the processor allocation inside each subtree or a subset of subtrees by applying our allocation approach recursively. The detailed algorithm is outlined as follows.

Procedure `Processor_Allocation(`T, r, p`)`
/* T is a weighted-node tree, r is the root of T, */
/* and p is the number of processors. */
/* Compute r by assigning p processors to it,*/
/* e.g., [15] algorithm can be applied. */
`Aggregate_multiprocessor(`v, p`);`
$t_{\min} := \lceil \frac{W_T - s(r)}{p} \rceil$; /* the time lower bound. */
Let $v_1, v_2, \ldots, v_d$ be the children of r,
and T_i be a subtree rooted at v_i, $1 \leq i \leq d$.
$i := 0$;
repeat
  $i := i + 1$;
  $p' := \lfloor \frac{W_{T_i}}{t_{\min}} \rfloor$;
  if $p' \neq 0$ **then**
    allocate p' processors to T_i;
    if $p' > 1$ **then**
      `Processor_Allocation(`T_i, v_i, p'`);`
    endif;
  else
    $i_0 := i$;
    $W' := W_{T_i}$;
    while $(p' = 0)$ & $(i \neq d)$
      $i := i + 1$;
      $W' := W' + W_{T_i}$;
      $p' := \lfloor \frac{W'}{t_{\min}} \rfloor$;
    endwhile;
    allocate a processor to a group $\mathcal{G}$ of subtrees $\{T_{i_0}, \ldots, T_i\}$; run an efficient sequential algorithm to compute all the nodes in $\mathcal{G}$;
  endif
until $i = d$.

Lemma 1 *If the* `while` *loop in procedure* `Processor_Allocation` *is executed, then either* $p' = 1$ *or* $p' = 0$ *after the loop termination, and* $p' = 0$ *only if* $i = d$.

Proof The proof for the case of $p' = 0$ is straightforward according to the `while` condition. Here we only show $p' = 1$ by considering $k = i - i_0 + 1$ subtrees are included after the `while` loop termination.

When $k = 1$, $p' = \lfloor W'/t_{\min} \rfloor = \lfloor (W_{T_{i_0}} + W_{T_{i_0+1}})/t_{\min} \rfloor \leq \lfloor 2W_{T_{i_0}}/t_{\min} \rfloor$ since $W_{T_{i_0}} \geq W_{T_{i_0+1}}$ by the initial assumption. Then we can derive that $0 < W'/t_{\min} < 2$ since $0 < W_{T_{i_0}}/t_{\min} < 1$. Therefore, $p' = \lfloor W'/t_{\min} \rfloor = 1$ because $p' \neq 0$.

Now consider the case of $k > 1$. Clearly $0 < \sum_{j=i_0}^{i-1} W_{T_j}/t_{\min} < 1$ by the `while` condition, and $W_{T_i} \leq W_{T_{i-1}}$. Then $0 < W'/t_{\min} = \sum_{j=i_0}^{i} W_{T_j}/t_{\min} < 2\sum_{j=i_0}^{i-1} W_{T_j}/t_{\min} < 2$, i.e., $0 \leq p' = \lfloor W'/t_{\min} \rfloor = 1$. Since $p' \neq 0$, so, $p' = 1$. □

Compared with `Algorithm 1`, in some sense `Algorithm 2` optimizes the total operations for the data cube computation globally. Moreover, this algorithm almost doesn't need any global synchronization (a synchronization by all processors involved), thereby reduces the entire response time. However, the processor allocation in this algorithm is more complicated. The efficiency of the algorithm is based on the size estimation for each node in $\mathcal{L}$, whereas the accuracy of the estimation depends on what fraction of the base table is used for samples, as well as which estimation method is used. If we only use a very small sample and the time used for this estimation takes a very small proportion of the entire computation, this may lead to an inaccurate estimation for the total operations for the data cube computation. As a result, the response time for the computation may be much worse than expected.

3.3 A matching-based parallel algorithm

In `Algorithm 2` we assumed that the size estimation of the nodes in $\mathcal{L}$ and the weight assignment of the edges in $\mathcal{L}$ are accurate. If not, the weighted sum of the directed minimum spanning tree T in $\mathcal{L}$ does not represent the real total optimal operations for the data cube computation. Therefore, the actual response time based on T is not the best one because the response time fully depends on the weight of

T as well the processor allocation to the nodes in T. Actually `Algorithm 2` first gives a *static* estimate of the total operations for the data cube computation, then schedules processors to the nodes in T for implementing the computation. In the following we present an algorithm which gives a more accurate size estimation to each node in $\mathcal{L}$. The algorithm is inspired by the sequential algorithm due to [14].

Their algorithm proceeds level-by-level, starting from level $l = 1$ to level $l = n + 1$. For each level l it finds the best way of computing the nodes at level $l + 1$ from level l by reducing the problem to a minimum weighted matching problem on a weighted bipartite graph. Here we should stress that in the sequential environment, the subsequent computation for the nodes in the matching is easy after finding the matching. However, in the parallel environment the computation for the nodes is more complicated which reflects issues such as how to allocate the processors to the nodes in the matching, and how to determine the number of processors which should be assigned to a node, etc. In the following we introduce a processor allocation strategy for this purpose. We refer to the processor allocation and node computation as *the second phase* of the proposed algorithm. The detailed algorithm is outlined as follows.

Initially we sort the base relation table by some order of its attributes at level one (root). Now we assume that all group-by aggregations at level l are available and these group-by aggregations are sorted in some order by their attributes. Initially $l = 1$.

Following [14], a bipartite graph $G = (X, Y, E)$ is constructed as follows: X is the set of nodes at level l, and Y is the set of nodes at level $l + 1$. There is a directed edge from $x \in X$ to $y \in Y$ if y can be generated from x, and the weight $C_{scan}(x, y)$ associated with this edge is the size of x. We then construct another weighted bipartite graph $G' = (X' \cup X, Y, E' \cup E)$ from $G(X, Y, E)$ as follows: $X' = \{x_1, x_2, \ldots, x_{n-l} \mid x \in X\}$, i.e., there are $n - 1$ new nodes in G' for each node $x \in X$. $\langle x_i, y\rangle \in E'$ if $\langle x, y\rangle \in E$, and it is assigned the cost $C_{sort}(x, y)$ of sorting x. The purpose of creating $n - 1$ corresponding edges of each edge (x, y) in G is that each node $y \in Y$ except x can be derived from the other $n - 1$ nodes.

Having the weighted bipartite graph G', find a minimum weighted matching $\mathcal{M}$ of G' which corresponds to a scheduling for computing the nodes in Y. An edge $\langle x_i, y\rangle$ in $\mathcal{M}$ means that y can be obtained by sorting x, and an edge $\langle x, y\rangle$ in $\mathcal{M}$ means that y can be obtained by scanning x without sorting x. Then the total operations for the node computation related to $\mathcal{M}$ is

$$W_{\mathcal{M}} = \sum_{\langle u,v\rangle \in \mathcal{M}, u \in X', \text{ and } v \in Y} C_{sort}(u, v) + \sum_{\langle u,v\rangle \in \mathcal{M}, u \in X, \text{ and } v \in Y} C_{scan}(u, v) \quad (6)$$

>From $\mathcal{M}$, a weighted forest $\mathcal{F} = (V', E')$ of trees is constructed as follows. $V' \subseteq V$, $E' \subseteq E$ and $\mathcal{L} = (V, E)$. Let $T \in \mathcal{F}$, if $\langle x, y\rangle \in \mathcal{M}$, then x is the root of T and y is a son of x and the weight of this edge is $C_{scan}(x, y)$; if $\langle x_i, y\rangle \in \mathcal{M}$, then x is the root of a tree and y is a son of x and the weight of this edge is $C_{sort}(x, y)$.

In the following we consider the processor allocation for the node computation at level $l + 1$ by $\mathcal{M}$. Let $t_{\mathcal{M}}$ be the time lower bound for computing all nodes at level $l + 1$, then

$$t_{\mathcal{M}} = a \cdot W_{\mathcal{M}}/p + b \cdot p \quad (7)$$

The remaining is to allocate processors to the trees in $\mathcal{F}$. We adapt the approach as following. We scan the elements in $\mathcal{F}$ one by one. Let $T \in \mathcal{F}$ be the current scanning element and let W_T be the weight of T. Then, the number of processors is assigned to T as follows: if $\lfloor W_T/t_{\mathcal{M}} \rfloor \neq 0$, we assign $\lfloor W_T/t_{\mathcal{M}} \rfloor$ processors to T, and remove T from $\mathcal{F}$; otherwise, we merge T with the next tree T', and repeat the above procedure until $\mathcal{F} = \emptyset$.

After finishing the processor allocation, the processors assigned to a tree or a subset of trees are dedicated to the computation of those (leaf) nodes in the tree or the subset of trees. In summary, the detailed algorithm is presented as follows.

Algorithm 3:
$l := 1$; /* Starting from level $l = 1$ */
repeat
1. Construct the bipartite graph $G(X, Y, E)$.
2. Construct the bipartite graph $G'(X' \cup X, Y, E' \cup E)$.
3. Find a minimum weighted matching $\mathcal{M}$ in G'.
4. Construct a weighted forest $\mathcal{F}$ from $\mathcal{M}$.
5. Allocate processors to the trees in $\mathcal{F}$.
6. Proceed group-by aggregations by each cluster of processors independently.

$l := l + 1$;
until $l = n + 1$.

`Algorithm 3` can compute the size of every node accurately, but the cost incurred for this purpose means that a synchronization must be carried out at each level in order to obtain the computation cost for the nodes at the next level. Meanwhile, the estimated running time $t_{\mathcal{M}}$ for processor allocation is also important. A rough estimation of $t_{\mathcal{M}}$ will result in a considerable synchronization overhead. Finally, it should be mentioned that the local optimization carried out at each level does not mean global optimization with regard to the total operations for the data cube computation.

4 A Hybrid Parallel Algorithm

In this section we introduce a hybrid algorithm for the data cube computation. >From `Algorithm 3` we can see that at the end of generating all group-by aggregations at each level, a synchronization is used to collect the size of each node at that level. As we know, if the estimated time $t_{\mathcal{M}}$

is correct, the overhead of the synchronization is not too much. Otherwise, we spend a lot of time for synchronization. In this section we combine Algorithms 2 and 3 by presenting a hybrid parallel algorithm to overcome some disadvantages incurred by each of them individually.

The basic idea is that we partition the lattice $\mathcal{L}$ into several subgraphs horizontally. The synchronization is only carried out at the bottom levels of these subgraphs. Let H be such a subgraph with the starting level l and the ending level $l+h$. Assume that the computation for all nodes, as well their sizes at level l, has been done. Then for every node v at level l, we use its information and its size to estimate the sizes of those nodes in H which are reachable from v. We assign an estimated size for every node in H by combining all estimates to it, and as a result, H becomes a node-weighted graph. We then find a directed minimum spanning forest of a weighted graph H, and allocate the processors to the trees in the forest according to their weights. Finally we use Algorithm 2 to compute the nodes in each subtree independently by using the processors allocated to it. In the following we illustrate these in detail.

Let N_l be the set of nodes at level l in $\mathcal{L}$. A subgraph H of $\mathcal{L}$ is an induced subgraph by the nodes from level l to level $l+h$, i.e., $H = (V_1, (V_1 \times V_1) \cap E)$ where $V_1 = \cup_{i=l}^{l+h} N_i$ and $\mathcal{L} = (V, E)$.

Define $den_H(v) = \{u \mid u \in V_1,\ and\ u\ is\ reachable\ from\ v\}$ for every $v \in N_l$; and $source_H(u) = \{v \mid v \in N_l,\ and\ u\ is\ reachable\ from\ v\}$ for every $u \in V_1 - N_l$. Now for every $v \in N_l$, it generates an estimated size $S_H(v,u)$ for every node $u \in den_H(v)$. As results, the estimated size of a node $u \in V_1 - N_l$ in H is

$$S_H(u) = \lceil \frac{\sum_{v \in N_i} S_H(v,u)}{|source_H(u)|} \rceil \quad (8)$$

In order to reduce the time for estimating the sizes of nodes, the estimation is actually carried out during generating the nodes in N_l.

A directed weighted graph H' from H then is obtained where $H' = (V_1 \cup \{s\}, (V_1 \times V_1) \cap E \cup \{\langle s, v\rangle \mid v \in N_l\})$, where s is a virtual node which can reach every other node in V_1. The weight associated with each edge in H' is defined as follows. For every edge $\langle s, v\rangle$, it is assigned weight zero; for every edge $\langle x, y\rangle$ with $x \neq s$, it is assigned weight $S_H(x)$.

Having H', find a directed minimum spanning tree $T_{H'}$ rooted at s in H'. After that, a forest $\mathcal{F}$ of trees is obtained by removing s and all its adjacent edges from $T_{H'}$. Now the second phase of Algorithm 2 can be applied to $\mathcal{F}$, i.e, allocate processors to each tree in $\mathcal{F}$ and implement the computation for those nodes which are not the tree roots in $\mathcal{F}$. For the sake of integrity, we present the entire algorithm below.

Algorithm 4:
$l := 1$; /* Starting from level $l = 1$ */
repeat
1. Construct a subgraph H induced by the nodes from level l to level $l+h$;
2. Estimate the size of every node in H by the information of the nodes in N_l;
3. Construct the subgraph H' from H;
4. Find a directed minimum spanning tree $T_{H'}$ rooted at s in H';
5. Construct a weighted forest $\mathcal{F}$ from $T_{H'}$;
6. Allocate processors to the trees in $\mathcal{F}$;
7. Proceed group-bys by each cluster of processors independently;

$l := l + h$;
until $l = n$.

Compared with the two previous algorithms Algorithms 2 and 3, Algorithm 4 reduces the number of synchronizations in Algorithm 3, and therefore reduces the entire response time. However, the size estimate of each node is not as accurate as the result we achieved in Algorithm 3. As a result, the total number of operations is not optimal. However this algorithm gives a much more accurate estimation of total operations than Algorithm 2 because we use the actual information and the sizes of the nodes at level l to estimate the sizes of the nodes in the subsequent h levels.

Here we give a variant of the hybrid parallel algorithm. Let P be a directed path from the root to a node v in the lattice $\mathcal{L}$, and the node sequence in P be $v_0, v_1, v_2 \ldots, v_k = v$. It is obvious that $s(v_0) \geq s(v_1) \geq \ldots \geq s(v_k)$ where $s(v_k)$ is the actual size of v. >From this observation, we have the following variant of the proposed algorithm. That is, at the first l levels, we use Algorithm 3 to compute the cuboids level by level. After that, we repeatedly use the hybrid algorithm for a subgraph consisting of the nodes by h_i levels ($h_i \geq 1$), and h_i is an ascending sequence, i.e, $h_j > h_i$ if $j > i$.

5 A Generalized Algorithm

In the previous two sections we have considered computing all dimensional group-by aggregations for a base relation table, or computing all nodes in the lattice. In practice, not all dimensional aggregates are interesting to analysts and decision makers. Instead only some of them are required quite often. Therefore, we only need to compute some nodes in the lattice.

Recall that R is an n-attribute relation and $\mathcal{L} = (V, E)$ is the lattice of R. Assume that each directed edge $\langle u, v\rangle \in E$ is assigned a cost which is the estimated operation cost of generating v from u.

We now construct a weighted directed graph $\mathcal{L}^* = (V^*, E^*)$ from $\mathcal{L}$ as follows: $V^* := V$. For an edge $\langle u, v\rangle \in E^*$, it is assigned the same weight as the edge $\langle u, v\rangle \in E$ in $\mathcal{L}$. If there is a directed path from u to v in $\mathcal{L}$, then $\langle u, v\rangle \in E^*$, and the weight associated with this edge

is the cost of generating v from u directly. Actually $\mathcal{L}^*$ is the transitive closure of $\mathcal{L}$.

Let $S \subset V$ be a proper subset of the nodes in $\mathcal{L}$ in which we are interested. The problem is to compute all group-by aggregations in S with minimizing the total operations for these aggregates. In order to optimize the total operations, we need to find a directed tree T_S rooted at R in $\mathcal{L}^*$ which contains all nodes in S such that the weighted sum of the edges in T_S is minimized. Clearly, T_S is a directed Steiner tree in $\mathcal{L}^*$. It is well known that the directed Steiner tree problem on a general directed graph is NP-complete [6], and even for this special Steiner tree problem, it is also NP-complete [14]. It is also well known that the minimum set cover (MSC) problem can be reduced to a directed Steiner tree problem in polynomial time, whereas [5] has shown that there is no polynomial approximation algorithm for the MSC problem which delivers a solution better than $(1 - \epsilon')\ln q$ times optimum unless there is an $O(n^{\log\log n})$ time algorithm for NP-complete problems for any fixed ϵ' with $0 < \epsilon' < 1$, where q is the problem size. Therefore, we have

Lemma 2 *Given a directed DAG $G(V, E)$ ($\mathcal{L}$ is a special DAG) with $|V| = N$, and each edge is assigned a weight, let $S \subset V - \{s\}$ be subset of the nodes in V where s is the unique source and every node $v \in S$ is reachable from s. Then there is a polynomial approximation algorithm for finding an approximate directed Steiner tree rooted at s including all nodes in S such that the weight of the tree is within $O(|S|^{\epsilon})$ times optimum if the running time is $O((N|S|)^{1/\epsilon}/N + N^3)$, where $0 < \epsilon \leq 1$.*

Proof By the result of [5], it is unlikely that there is a polynomial algorithm for the directed Steiner tree problem which gives an approximation solution better than $(1 - \epsilon')\ln N$ times the optimum. However, by using the algorithm due to Zelikovsky [16], the claim in the lemma is straightforward.□

Basically `Algorithm 2` can be applied here with minor modifications, i.e, replace the subroutine for finding a minimum directed spanning tree by the subroutine for finding an approximately Steiner tree T_S. Now we sketch the steps for computing all aggregates in S as follows.
Step 1. Construct the lattice $\mathcal{L}$ of R and assign the edges in $\mathcal{L}$ with weights.
Step 2. Construct $\mathcal{L}^*$ from $\mathcal{L}$ and assign weights to the edges in $\mathcal{L}^*$.
Step 3. Find an approximation Steiner tree T_S in $\mathcal{L}^*$, using the directed Steiner tree algorithm [16].
Step 4. Allocate processors to the nodes in T_S and carry out the aggregates for all corresponding nodes using the processors allocated to them. This can be done by using the second phase of `Algorithm 2`.

Actually the approach above can be applied to any DAG and not just the lattice only.

6 Conclusions

In this paper we have developed four parallel algorithms to compute the data cube efficiently in the parallel computational environment. The proposed algorithms introduce a novel processor scheduling policy and a non-trivial decomposition approach for the problem. Particularly, the hybrid algorithm has the best performance potential among the four proposed algorithms. Partial contents of this paper also appeared in [11].

Appendix: Procedure for estimating the size of group-bys. As suggested in [8], we first convert the raw data by mapping each attribute value into a unique integer before starting to compute the cube. We can use this data conversion step to get the number of distinct values, D_i for each attribute, A_i of the cube. A starting estimate of the size of group-by $A_1A_2\ldots A_k$ consisting of attributes 1 through k can then be taken to be $\min\{D_1 \times D_2 \times \ldots \times D_k, N'\}$ where N' is the number of the tuples in the raw table.

These initial estimates are then refined in the following ways. (1) Once a group-by is computed, then the estimated size of all group-bys derivable from it can be strictly smaller than this group-by. (2) Once we have covered two levels we can get better estimates as follows.

$$\frac{|ABCD|}{|ABC|} \leq \frac{|ABD|}{|AB|} \tag{9}$$

therefore,

$$|AB| \leq \frac{|ABD||ABC|}{|ABCD|} \tag{10}$$

References

[1] Agawal S., Agrawal R., Deshpande P. M., Gupta A., Naughton J. F., Ramakrishnan R. & Sarawagi S. (1996) On the computation of multidimensional aggregates. *Proc. of the 22nd VLDB Conf.*, Mumbay, India, p. 506-521.

[2] Bitton D., Boral H., et al (1983) Parallel algorithms for the execution of relational database operations. *ACM Trans. on Database Systems*, 8(3), p. 324–253.

[3] Codd E. F. (1993) Providing OLAP: an IT mandate. *Unpublished manuscript*, E. F Codd and Associates.

[4] Deshpande P. M., Agarwal S., Naughton J. F., Ramakrishnan R. (1996) Computation of multidimensional aggregates. *Technical Report*, No: 1314, Dept. of CS, Univ. of Wisconsin-Madison.

[5] Feige U. (1996) A threshold of $\ln n$ for approximating set cover. *Proc. of 28th ACM Symp. on Theory of Computing*, p. 314–318.

[6] Garey M. R. & Johnson D. S. (1979) *Computers and Intractability*. W. H. Freeman, San Francisco.

[7] Graefe G. (1993) Query evaluation techniques for large databases. *ACM Computing Surveys*, 25(2), p. 73–170.

[8] Gray J., Bosworth A., Layman A. & Prahesh H. (1995) Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub-total. *Microsoft Technical Report*, MSR-TR-95-22, 1995.

[9] Gupta H., Harinarayan V., Rajaraman A. & Ullman J. D. (1997) Index selection for OLAP. *Proc. of Int'l Conf. on Data Engineering*, Birmingham, UK, p. 208-219.

[10] Harinarayan V., Rajaraman A. & Ullman J. D. (1996) Implementing data cubes efficiently. *Proc. of the 1996 ACM-SIGMOD Conf.*, p. 205–216.

[11] Liang W. & Orlowska M. E. (1998) Computing multidimensional aggregates in parallel. *Proc. of the 1998 Int'l Conf. on Parallel and Distributed Systems*, Taiwan, IEEE Computer Society Press, p. 92–99.

[12] Mumick I. S., Quass D. & Mumick B. S. (1997) Maintenance of data cubes and summary tables in a warehouse. *Proc. of the 1997 ACM-SIGMOD Conf.*, p. 100–111.

[13] Ross K. A. & Srivastava D. (1997) Fast computation of sparse datacubes. *Proc. of the 23rd VLDB Conf.*, Athens, Greece, p. 116-125.

[14] Sarawagi S., Agrawal R. & Gupta A. (1996) On the computing the data cube. *Research Report*, IBM Almaden Research Center, San Jose, CA.

[15] Shatdal A. & Naughton J. F. (1995) Adaptive parallel aggregation algorithms. *Proc. of the 1995 ACM-SIGMOD Conf.*, p. 104–114.

[16] Zelikovsky A. (1997) A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, vol.18, p. 99-110.

[17] Zhao Y., Deshpande P. M. & Naughton J. F. (1997) An array-based algorithm for simultaneous multidimensional aggregates. *Proc. of the 1997 ACM-SIGMOD Conf.*, p. 159–170.

Fractal Geometry for Natural-Looking Tree Generation

I.A.R. Moghrabi and Farid Raidan
Natural Science Division,
Lebanese American University,
P.O. Box 13-5053 Beirut, Lebanon
e-mail: imoghrbi@lau.edu.lb

Keywords: topological model, poset, spatial relations, lattice completion.

Edited by: Marcin Paprzycki
Received: October 5, 1999 **Revised:** November 11, 1999 **Accepted:** January 10, 2000

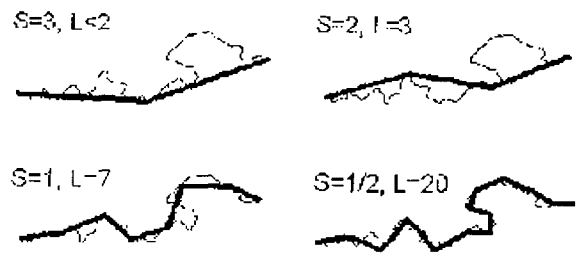
Of the most known natural looking objects generated by Fractal Geometry are trees. We address in this work several modified popular techniques for generating randomized naturally looking trees with non-uniformity in the pattern of generation to faithfully model irregularities in shape. In particular, we focus on L-System models for the generation of random natural phenomena. We stress statistical rather than absolute self-similarity.

1 Introduction

Sharp boundaries or smooth shapes for real entities reflect a map model or geometric bias rather than an appropriate model for nature. Fractal geometry is used to represent those models that are not smooth in shape as in the case of natural phenomena, such as coastlines. Binoit Mandelbrot recognized that the relationship between large scale structure and small scale detail is an important aspect of natural phenomena. He gave the name fractals to objects that exhibit increasing details as one zooms in closer. Mandelbrot raised an example question "How long is the coast-line of Great Britain?". At first sight this question may seem trivial. Given a map one can sit down with a ruler and soon come up with a value for the length. The problem is that repeating the operation with a larger scale map yields a greater estimate of the length (Fig. 1). If one actually went to the coast Fig 1. Using sticks of different size S to estimate the length L of the coast-line and measured them directly, then still greater estimates would result. It turns out that as the scale of measurement decreases the estimated length increases without limit.

In discussing measurement, scale can be characterized in terms of a measuring stick of a particular length: the finer the scale, the shorter the stick. Thus at any particular scale, we can think of a curve as being represented by a sequence of sticks (Fig. 1), all of the appropriate length, joined end-to-end. Clearly, any feature shorter than the stick will vanish from a map constructed this way. Of course no one actually makes map by laying sticks on the ground, but the stick analogy reflects the sorts of distortions that are inevitably produced by the limited resolution of the photographs, or by the thickness of the pen being used in drafting.

An important difference between fractal curves and idealized curves that are normally applied to natural processes is that fractal curves are nowhere differentiable. Fractals



can be characterized by the way in which representation of their structure changes with changing scale. So it is important to realize that true fractals are an idealization. No curve or surface in the real world is a true fractal; real objects are produced by processes that act only over a finite range of scales only.

Some terminology used with fractals:

Iterate: to repeat an operation, generally using the last result of that operation as the input.

Self-similarity: a similar appearance at all scales. More about this is shown later.

Fractal dimension: the dimension of a fractal in general is not a whole number, not an integer. Also, more on this after some more examples of fractals.

Replacement rule: in going from one stage of construction of a fractal to the next, one graphical object is replaced with another.

Fractals can be generated using recursive algorithms as well as other algorithms such as the grammar based model which is called L-system.

2 Self-similarity:

Self-similarity is symmetry across different scales; there are patterns within patterns. Fractals are geometric shapes

that are equally complex in their details as in their overall form. That is, if the small scale detail resembles the large scale detail, the object is said to be self-similar. An appreciation for these concepts is best approached by seeing how particular types of objects can be created.

(i) Cantor dust:

Consider a line segment split into 3 equal parts with the middle section discarded. An indefinitely long continued process of splitting the line segments will in the end produce a set of very small aligned segments or points (whose length is tending to zero) which are called Cantor dust. (Fig 2)

Such a process of pattern generation has two components:

A. An initiator (for cantor dust this is the line segment).

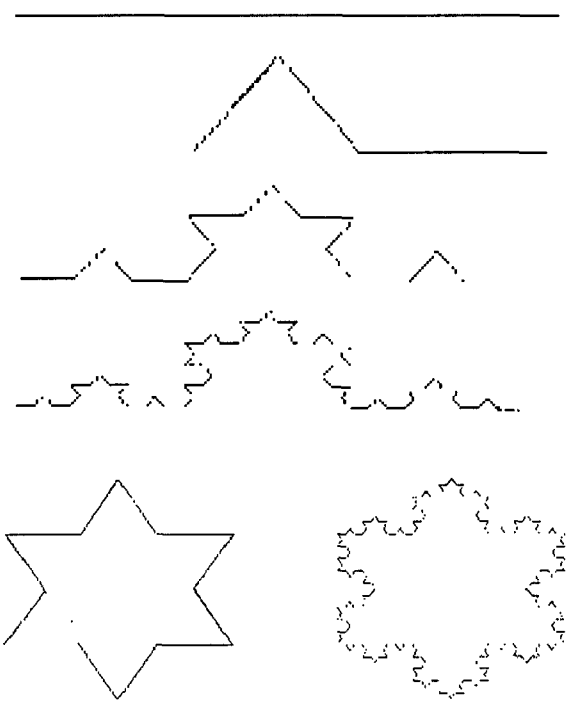
B. A replacement rule (two other segments) or pattern generator.

At step n, the set consists of 2n segments. The segment length is "l3-n", where l is the unit length of the initiator. Thus, for the third level of application of the recursive process, if the original line was 4 units, then each segment piece will be 4/9. For the replacement rule, its self-similarity ratio is 1/3, identifying the subdivision of the original line into 3 pieces.

(ii) Von Koch curve:

Starting with a segment with a bump on it as shown in fig 3, replace each segment of the line by a figure exactly like the original one. This process is repeated: Each segment in part (b) in the figure is replaced by a shape exactly like the entire figure. It is easy to imagine that at infinity, there will be a continuous curve formed by the succession of

small angles. If the line segment of the initiator of the Von Kock curve is replaced by an equilateral triangle, a snowflake is produced by the recursive process. At step one, a six-pointed star corresponding to a polygon with 12 sides replaces the initiator. At step 2, there are 48 sides; at each step the number of sides is multiplied by 4. So, for the initiator perimeter of length "l", the perimeter become "l(4/3)n" at step n which tends to infinity although the area tends to a finite limit. The self-similarity ratio is 1/3 (Fig 4, 5).



iii) Sierpinski triangle:

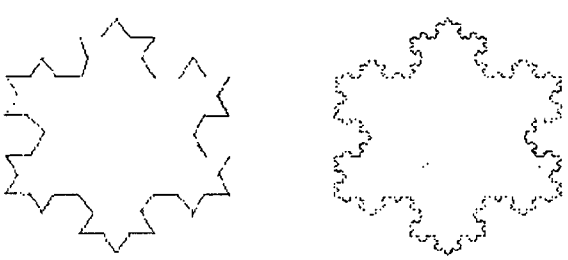
To make a fractal from a triangle, draw lines connecting the midpoints of the sides and

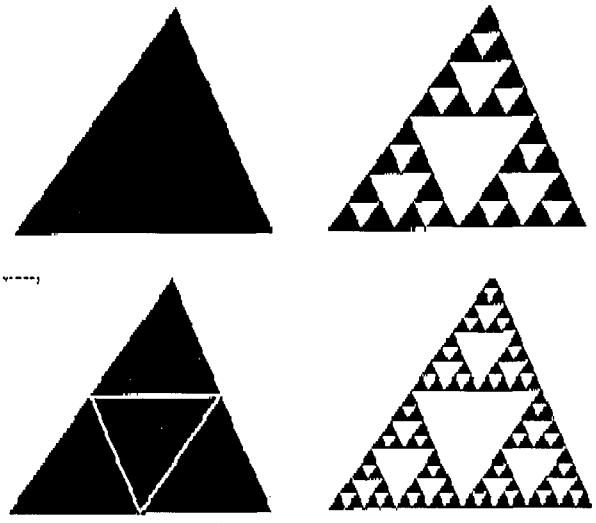
cut out the center. Take the result and do again and again (Fig. 6, 7).

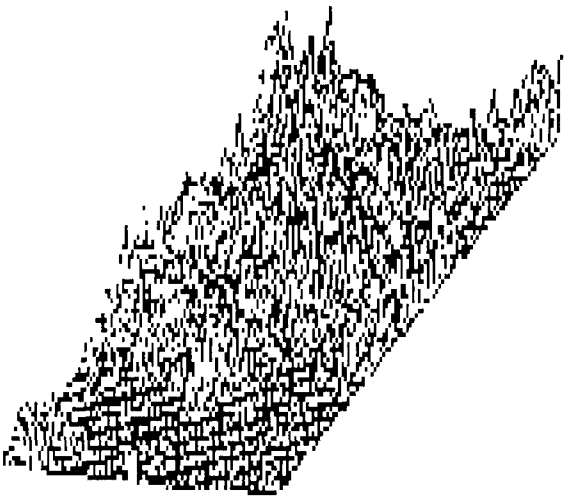
The above fractals look the same over all ranges of scale. This is called "self-similarity".

3 Fractal dimension:

Associated with the notion of self-similarity is the notion of fractal dimension. The notion of fractal dimension provides a way to measure how rough fractal curves are. The dimension of a fractal is not an integer. So, a fractal curve (a one-dimensional object in a plane that has two dimensions) has a fractal dimension that lies between 1 and 2. Likewise, a fractal surface has a dimension between 2 and 3. The value depends on how the fractal is constructed. The closer the dimension of a fractal is to its possible upper limit which is the dimension of the space in which it is embedded, the rougher, i.e. the more filling of that space it is (Fig.s 8 and 9).





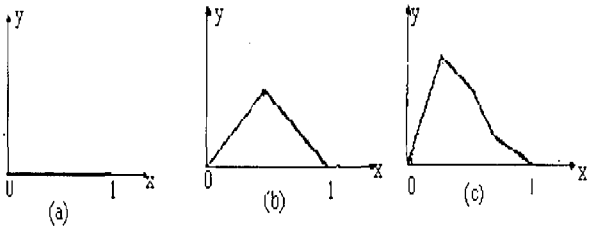


To compute fractal dimension, examine first some properties of objects whose dimension is known. For example: a line segment is 1D; if it is divided into N equal parts, each part looks like the original scaled down by a factor of $N = N^{1/1}$. A square is 2D: if it's divided into N parts, each part looks like the original scaled down by a factor of $\sqrt{N} = N^{1/2}$. For example, a square divides nicely into 9 subsquares, each looks like the original scaled down by a factor of 1/3.

Consider now the fractal, the von Koch curve. When it's divided into 4 pieces (the pieces associated with original 4 segments in fig.s 3 and 4), each resulting piece looks like the original scaled down by a factor of 3. Let d be the dimension of the von Koch curve where 41/d=3 Þ d= log(4)/log(3) = 1.26 which is not an integer.

4 Statistical self-similarity

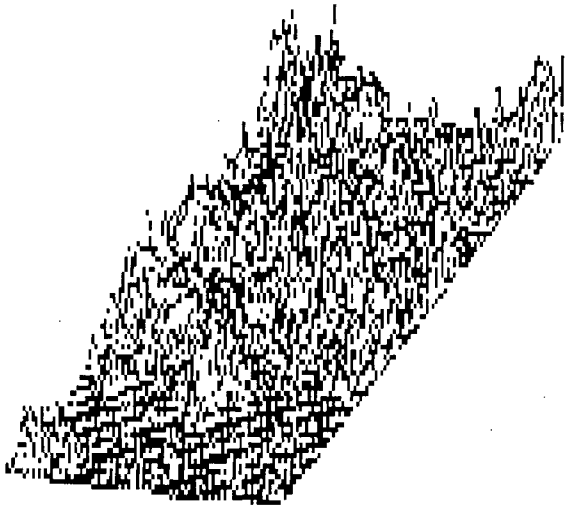
To create more natural looking shapes, that is involving variety, use randomization. This is known as stochastic fractals.

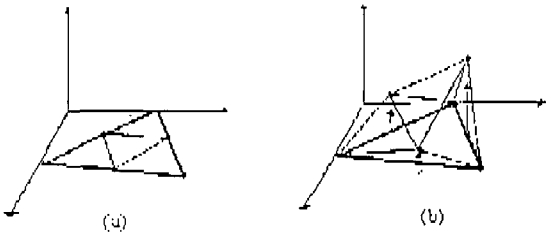


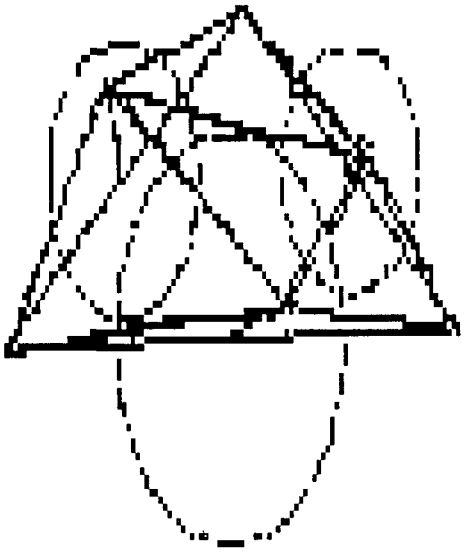
For example: starting with a line segment lying on the x-axis (Fig 10a). Subdivide the line into 2 halves and then move the mid-point some distance in the y direction (Fig 10b). To continue subdividing each segment, compute a new value for the mid-point of the segment from (Xj , Yj) to (Xj+1, Yj+1) as follows: Xnew=1/2(Xj + Xj+1), Ynew=1/2(Yj + Yj+1) + P(Xj+1- Xj)*R(Xnew); where P is a function determining the extent of the perturbation in terms of the size of the line being perturbed (P(s) could be s or sa or 2-s), and R( ) is a random number between 0 and 1 selected on the basis of Xnew (Fig 10b).

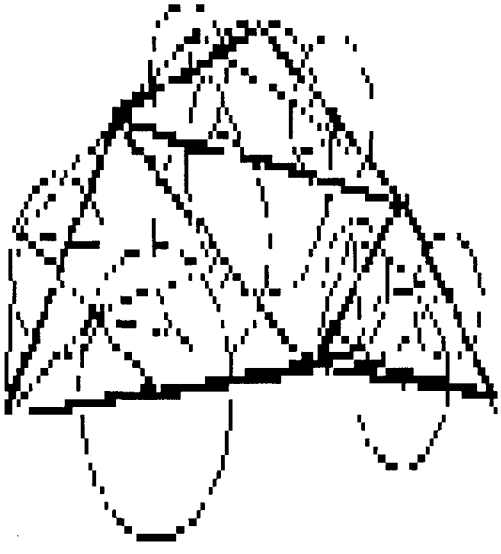
This process can be used to modify 2D shapes in the following fashion. Starting with a triangle, mark the midpoints of each edge in fig 11a. The y coordinate is then modified in the manner described above, so that the result looks like in fig 11b. This process when iterated produces realistic-looking mountains.

Example 2: adding a little randomness to the Sierpinski triangle can produce something very natural looking. Change the rule so that instead of using the exact midpoints









of the sides of the triangle, take a point at random around the midpoint. The self-similarity will be statistical rather than absolute. Constrain the randomness by making the point some where within the circle centered on the midpoint whose diameter is half the length of the side (see fig 12). Connect these points and the corners. The heavy lines are the starting triangle (Fig. 12). In fig 13 the heavy lines are the result of the last iteration. Fig. 14 is the result of clearing the construction and making the corners black. Fig 15 is the result after 8 more iterations, and enlarging.

These results are extremely suggestive for modeling natural forms, since many natural objects seem to exhibit striking self-similarity. Mountains have peaks and smaller peaks and rock which all look similar; trees have limbs and branches, which again look similar. Hence modeling self-similarity or statistical self-similarity at some scale seems to be a way to generate appealing-looking models of natural phenomena. When an object has been generated recursively through enough steps that all further changes happen at well below pixel resolution, there is no need to continue.

5 L-systems:

L-systems are sets of rules and symbols (also known as "formal grammars") that model growth processes. The name "L-system" is short for "Lindenmayer System". A simple L-system contains four elements:

1. VARIABLES are symbols denoting elements that are replaced.

2. CONSTANTS are symbols denoting elements that remain fixed.

e.g. The expression <subject> <verb> <predicate>

consists of grammatical variables. Each variable may be replaced by constants (English words or phrases) to produce sentences in english, such as "The cat sat on the mat" or "The dog ate the bone".

3. RULES ("syntax") define how the variables are to be replaced by constants or other variables. e.g. in the above example

<subject> →the cat

would be one such rule.

4. START words are expressions defining how the system begins. e.g. the above

examples from english might start from a single variable <sentence>

Example - Fibonacci numbers:

Consider the simple grammar, defined as follows

Variables : A B

Constants : none

Start : A

Rules : A →B

B →AB

This L-system produces the following sequence of strings .

Stage 0 : A

Stage 1 : B

Stage 2 : AB

Stage 3 : BAB

Stage 4 : ABBAB

Stage 5 : BABABBAB

Stage 6 : ABBABBABABBAB

Stage 7 : BABABBABABBABBABABBAB

If the length of each string is counted, the famous Fibonacci sequence of numbers will be obtained:

1 1 2 3 5 8 13 21 34

The power of L-systems comes when we assign meaning to the symbols and rules. For instance the symbols might represent branches of a growing tree and the rules denote the way these symbols are to be used.

These languages are described by the grammar consisting of a collection of productions, all of which are applied at once. A typical example is the grammar with variables A and B, constants "[" and "]", and two production rules:

1. A →AA

2. B →A[B]AA[B]

Starting from axiom A, the first few generations are A, AA, AAAA, Starting from axiom B, the first few generations are:

1. B

2. A[B]AA[B]

3. AA[A[B]AA[B]]AAAA[A[B]AA[B]]

4.

Suppose that a word in the language represents a sequence of segments in a graph structure and that the bracketed portions represents portions that branch from the symbol preceding them; then the associated pictures are shown in fig16.

Adding two more constants "(" and ")", and changing the second production to:

2. B→ A[B]AA(B)

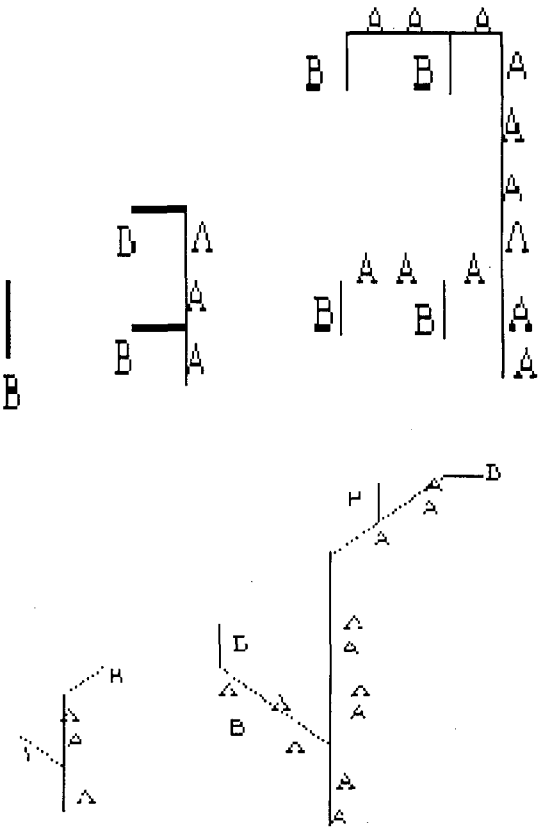
Then starting from axiom B, the first few generations are:

1. B

2. A[B]AA(B)

3. AA[A[B]AA(B)]AAAA(A[B]AA(B))

4.



Now suppose that square brackets denote a left branch and parenthesis denote a right branch, then the associated pictures are shown in fig 17.

By progressing to later generations in such a language, graph structures representing complex patterns will be produced. These graph structures have a sort of self similarity, in that the pattern described by the nth-generation is contained (repeatedly, in this case) in the (n + 1)th-generation word.

6 A New Variant of the L-System

Choosing varying branching angles for different depth branches, and varying thickness for lines (or even cylinders) representing the segments gives different results; drawing a 'leaf' or 'flower' at each terminal node of the tree further enhances the picture. The grammar itself has no inherent geometric content, so using a grammar-based model requires both a grammar and a geometric interpretation of the language. At some point, additional features should be added to the grammar or to the interpretation of a word in it.

So, in order to handle the above requirements, the L-system should be modified as follows:

First, rather than having only one replacement rule for each letter, modify the productions as follows. E.g.

B→ A[B](B) | A[B]A(B) |A[B](AB) | C

Here B could be replaced by one of the three productions or the constant C at the right hand side of the arrow. In or-

der to decide which one of these productions is to be used, a random number "r" between 1 and 4 is generated and according to the value of "r", the corresponding production is used.

e.g. if r = 1 B→ A[B](B)
Else if r = 2 B→ A[B]A(B)
Else if r = 3 B→ A[B](AB)
Else B→ C

Doing this, the productions will be applied probabilistically rather than deterministically so that not to have exact self-similarity. The letter B here corresponds to the bud that could be replaced by different shapes (i.e. auxillary or apical buds) or it can be replaced by a constant that corresponds to leaves. However, replacing B by the constant should be done at stages closer to the level of the tree, otherwise we cannot have further substitutions. Also, the interpretation of the word generated from the grammar should be modified according to the order computed. For example, if the order is 1, then a thick black line should be selected by the turtle, and if the order is close to the level (i.e. the final stage), a different line should be selected with less thickness (and may be with different color e.g. green). Also, leaves should be displayed when the order increases and this is done by ensuring that the constant C could be selected when the order is around its upper limit.

The order of the word is computed as follows:

First the order is 1, then after each constant that indicates the start of a branch (e.g. '(' or '['), the order is incremented and at each constant that indicates the end of the branch (e.g. ')' or ']'), the order is decremented.

Varying the values for the probabilities and the angles can produce a wide variety of extremely convincing tree models. The correct choices for these parameters depend on knowledge of plant biology or on the modeler's artistic eye; by using the wrong values, they can also generate plants bearing no resemblance at all to anything real.

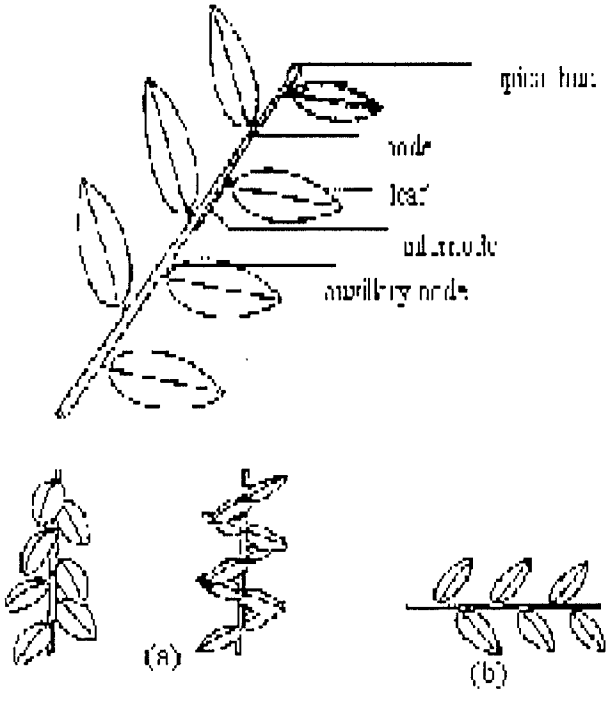
The productions of the grammar are applied probabilistically rather than deterministically. In this model, start as before with a single stem. At the tip of this stem is a bud, which can undergo several transitions: it may die, it may flower and die, it may sleep for some period of time, or it may become an internode, a segment of the plant between buds. The process of becoming an internode is added, and the end of the internode becomes an apical bud (a bud at the very end of the sequence of internodes) see fig18.

Each of the buds in the resulting object can then undergo similar transitions. Suppose the initial segment of the tree is of order-1, define the order of all other

internodes inductively: Internodes generated from the apical bud of an order-i internode are also of order-i; those generated from auxiliary buds of an order-i internode are of order (i + 1).

Thus the entire trunk of a tree is of order-1, the limbs are order-2, the branches on those limbs are of order-3, and so on (Fig19). The placement of auxiliary buds on a sequence of order-i internodes may occur in different

rways (Fig19a), and the angles at which the order (i + 1)



internodes (if any) branch out from order-i auxiliary buds also determine the shape of the plant (Fig 19b). There are some anomalies in the tree growth, in which the behavior of a collection of order (i + 1) internodes is not standard, but instead resembles that of some lower order (called reiteration), and this is too must be modeled. Finally, converting this description into an actual image requires a model for the shapes of its various components: an order-1 internode may be large tapered cylinder, and an order-7 internode may be a small green line, for example. The sole requirement is that there must be a leaf at each auxiliary node.

Varying the values for the probabilities and the angles can produce a wide variety of extremely convincing tree models. The correct choices for these parameters depend on knowledge of plant biology or on the modeler's artistic eye; by using the wrong values, they can also generate plants bearing no resemblance at all to anything real.

7 Conclusion

Fractal geometry proves to be more adequate than Euclidean geometry for generating naturally looking objects modeling natural phenomena. Stochastic fractals (statistical self-similarity) create more 'natural looking' shapes than exact self-similarity since the former involves randomization which is closer to nature because nature is not exactly self similar. Fractals are recursive in nature; however, in producing trees for example, a similar approach could be used which is called graftals (L-grammar). This method can be used in any domain in which the object being modeled exhibit sufficient regularity.

References

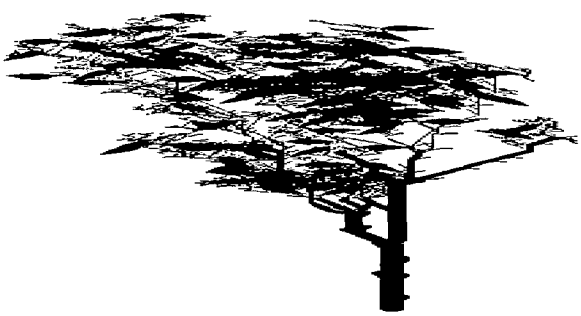
[1] J.A. Ford, and P.J. Cooper "Chaos and Fractals in Numerical Computation," Science and Engineering No. 5 (1991).

[2] Foley, Vandam, Feiner, and Hughes "Computer Graphics: Principles and Practice," 2nd ed. (1993) pp. 1020-1030.

[3] Michael McGuire, "In Eye For Fractals," (1991).

[4] David G. Green, "L-systems," Environmental and Information Sciences, Charles Sturt University (1993).

[5] David G. Green, "Fractals and Scale," Environmental and Information Sciences, Charles Sturt University (1993).



Appendix

This appendix shows some sample trees as generated by the new approach (fig.s 20-24)

Elementary sets and declarative biases in a restricted gRS–ILP model

Arul Siromoney,
School of Computer Science and Engineering,
Anna University, Chennai – 600 025, India
asiro@vsnl.com
AND
Katsushi Inoue,
Department of Computer Science and Systems Engineering,
Faculty of Engineering,
Yamaguchi University, Ube 755–8611, Japan
inoue@csse.yamaguchi-u.ac.jp

Keywords: Rough Set Theory, Inductive Logic Programming, Machine Learning, Knowledge Discovery from Databases

Edited by: Xindong Wu

Received: April 15, 1999 **Revised:** October 4, 1999 **Accepted:** December 10, 1999

Rough set theory is a powerful model for imprecise information. Inductive logic programming (ILP) is a machine learning paradigm that learns from real world environments, where the information available is often imprecise. The rough setting in ILP describes the situation where the setting is imprecise and any induced logic program will not be able to distinguish between certain positive and negative examples. The gRS-ILP model (generic Rough Set Inductive Logic Programming model) provides a framework for ILP in a rough setting. The formal definitions of the gRS–ILP model and the theoretical foundation for definitive description in a rough setting are presented. Definitive description is the description of data with 100% accuracy and is of use in the context of Knowledge Discovery from Databases. Several declarative biases and the formation of elementary sets in a restricted gRS–ILP model are then studied. An illustrative experiment of the definitive description of mutagenesis data using the ILP system Progol is presented.

1 Introduction

Inductive Logic Programming (in the example setting) [,] uses background knowledge (definite clauses), and positive and negative examples (ground facts) to induce a logic program that describes the examples, where the induced logic program consists of the original background knowledge along with an induced hypothesis (as definite clauses).

Rough set theory [,] defines an indiscernibility relation, where certain subsets of examples cannot be distinguished. A concept is rough when it contains at least one such indistinguishable subset that contains both positive and negative examples. It is inherently not possible to describe the examples accurately, since certain positive and negative examples cannot be distinguished.

The gRS–ILP model [] introduces the rough setting in ILP that describes the situation where the background knowledge, declarative bias and evidence are such that it is not possible to induce any logic program from them that is able to distinguish between certain positive and negative examples. Any induced logic program will either cover both the positive and the negative examples in the group, or not cover the group at all, with both the positive and the negative examples in this group being left out. (The declarative bias minimally restricts the presence of examples in the hypothesis, since otherwise a hypothesis that is the same as the positive examples will exactly distinguish the positive examples.)

This paper is an extended version of the paper presented at []. The formal definitions of the gRS–ILP model and the theoretical foundation of definitive description in a rough setting are presented. Several declarative biases and the formation of elementary sets in a restricted gRS–ILP model are then studied. A brief comparison is then made with other learning paradigms, including the fuzzy–set rough–set ILP system EAGLE []. An illustrative experiment of the definitive description of mutagenesis data using the ILP system Progol is reported.

Definitive description is one of the several possible application areas of the gRS–ILP model. Description focuses on finding human–interpretable patterns describing the data. Definitive description is the description of the data with full accuracy. In a rough setting, it is not possible to definitively describe the entire data, since some of the positive examples and negative examples (of the concept being described) inherently cannot be distinguished from each other.

Conventional systems handle a rough setting by using various techniques to induce a hypothesis that describes the evidence as well as possible. They aim to maximize the correct cover of the induced hypothesis by maximizing the number of positive examples covered and negative exam-

ples not covered. This is usually done by allowing a certain amount of coverage of negative examples. This means that most of the positive evidence would be described, along with some of the negative evidence. The induced hypothesis cannot say with certainty whether an example definitely belongs to the evidence or not. However, the gRS–ILP model lays a firm theoretical foundation for the definitive description of data in a rough setting. A part of the data is definitively described. The rest of the data can then be described using conventional methods, but not definitively.

The highlights of this paper are

- the formal definition of the gRS–ILP model,
- the theoretical foundation for definitive description in a rough setting,
- a study of various declarative biases and the formation of elementary sets in a restricted gRS–ILP model, and
- the report of an illustrative experiment using Progol on mutagenesis data.

2 Inductive Logic Programming

Inductive Logic Programming [] is the research area formed at the intersection of logic programming and machine learning. The semantics of ILP systems are discussed in []. In ILP systems, background (prior) knowledge B and evidence E (consisting of positive evidence E^+ and negative evidence E^-) are given, and the aim is then to find a hypothesis H such that certain conditions are fulfilled.

In the *normal semantics*, the background knowledge, evidence and hypothesis can be any well-formed logical formula. The conditions that are to be fulfilled by an ILP system in the normal semantics are

Prior Satisfiability: $B \wedge E^- \not\models \Box$
Posterior Satisfiability: $B \wedge H \wedge E^- \not\models \Box$
Prior Necessity: $B \not\models E^+$
Posterior Sufficiency: $B \wedge H \models E^+$

However, the *definite semantics*, which can be considered as a special case of the normal semantics, restricts the background knowledge and hypothesis to being definite clauses. This is simpler than the general setting of normal semantics, since a definite clause theory T has a unique minimal Herbrand model $\mathcal{M}^+(T)$, and any logical formula is either true or false in the minimal model. The conditions that are to be fulfilled by an ILP system in the definite semantics are

Prior Satisfiability: all $e \in E^-$ are false in $\mathcal{M}^+(B)$
Posterior Satisfiability: all $e \in E^-$ are false in $\mathcal{M}^+(B \wedge H)$
Prior Necessity: some $e \in E^+$ are false in $\mathcal{M}^+(B)$
Posterior Sufficiency: all $e \in E^+$ are true in $\mathcal{M}^+(B \wedge H)$

The Sufficiency criterion is also known as *completeness* with respect to positive evidence and the Posterior Satisfiability criterion is also known as *consistency* with the negative evidence.

The special case of definite semantics, where evidence is restricted to true and false ground facts (examples), is called the *example* setting. The example setting is thus the normal semantics with B and H as definite clauses and E as a set of ground unit clauses. The example setting is the main setting of ILP employed by the large majority of ILP systems.

3 Rough set theory

The basic notions of rough set theory are defined in [], and in [], which is an excellent reference for the fundamentals of rough set theory.

Let U be a certain set called the *universe*, and let R be an equivalence relation on U. The pair $A = (U, R)$ is called an *approximation space*. R is called an *indiscernibility relation*. If $x, y \in U$ and $(x, y) \in R$ we say that x and y are indistinguishable in A.

Equivalence classes of the relation R are called *elementary sets*, and every finite union of elementary sets is called a *composed set*.

Let X be a certain subset of U. The greatest composed set contained in X is the *best lower approximation* (or *lower approximation*) of X (known as $\underline{R}(X)$), i.e., $\bigcup_{[x]_R \subseteq X}[x]_R = \underline{R}(X)$ where for each $x \in U$, $[x]_R = \{y \in U \mid (x, y) \in R\}$. $\underline{R}(X)$ is also known as the *R–positive region* of X ($Pos_R(X)$). The lower approximation is the collection of those elements that can be classified with full certainty as members of set X using R. In other words, elements of $Pos_R(X)$ surely belong to X.

The least composed set containing X is the *best upper approximation* (or *upper approximation*) of X (known as $\overline{R}(X)$), i.e., $\bigcup_{[x]_R \cap X \neq \phi}[x]_R = \overline{R}(X)$. The *upper approximation* of X consists of elements that could possibly belong to X. In other words, R does not allow us to exclude the possibility that they may belong to X.

The *R–negative region* is the complement of the upper approximation with respect to the universe U ($Neg_R(X) = U - \overline{R}(X)$). The R–negative region is the collection of elements that can be classified without any ambiguity using R, that they do not belong to the set X. In other words, elements of $Neg_R(X)$ surely belong to the complement of X, that is, elements of the R–negative region surely do not belong to X.

The *R–borderline region* of X (or *boundary* of X), $Bnd_R(X) = \overline{R}(X) - \underline{R}(X)$, is the undecidable area of the universe. None of the elements in the boundary region can be classified with certainty into X or $U - X$ using R.

If the R–borderline region of X is empty, X is *crisp* in R (or X is *precise* in R); and otherwise, if the set X has some non–empty R–borderline region, X is *rough* in R (or X is *vague* in R).

4 Formal definitions and fundamental facts of the gRS–ILP model

The generic Rough Set Inductive Logic Programming (gRS–ILP) model introduces the basic definition of elementary sets and the rough setting in ILP []. The essential feature of an elementary set is that it consists of examples that cannot be distinguished from each other by any induced logic program in that ILP system. The essential feature of a rough setting is that it is inherently not possible for the consistency and completeness criteria to be fulfilled together, since both positive and negative examples are in the same elementary set.

4.1 The RSILP system

We first formally define the ILP system in the example setting of [] as follows.

Definition 1. An *ILP system in the example setting* is a tuple $S_{es} = (E_{es}, B)$, where
(1) $E_{es} = E^+_{es} \cup E^-_{es}$ is the *universe*, where E^+_{es} is the set of positive examples (true ground facts), and E^-_{es} is the set of negative examples (false ground facts), and
(2) B is a background knowledge given as definite clauses, such that
(i) for all $e^- \in E^-_{es}$, $B \not\vdash e^-$, and
(ii) for some $e^+ \in E^+_{es}$, $B \not\vdash e^+$.

Let $S_{es} = (E_{es}, B)$ be an ILP system in the example setting. Then let $\mathcal{H}(S_{es})$ (also written as $\mathcal{H}(E_{es}, B)$) denote the set of all possible definite clause hypotheses that can be induced from E_{es} and B, and be called the *hypothesis space* induced from S_{es} (or from E_{es} and B). Further, let $\mathcal{P}(S_{es})$ (also written as $\mathcal{P}(E_{es}, B)) = \{P = B \wedge H \mid H \in \mathcal{H}(E_{es}, B)\}$ denote the set of all the programs induced from E_{es} and B, and be called the *program space* induced from S_{es} (or from E_{es} and B).

Our aim is to find a program $P \in \mathcal{P}(S_{es})$ such that the next two conditions hold:
(iii) for all $e^- \in E^-_{es}$, $P \not\vdash e^-$,
(iv) for all $e^+ \in E^+_{es}$, $P \vdash e^+$.

The following definitions of Rough Set ILP systems in the gRS–ILP model (abbreviated as *RSILP systems*) use the terminology of [].

Definition 2. An *RSILP system in the example setting* (abbreviated as RSILP–E system) is an ILP system in the example setting, $S_{es} = (E_{es}, B)$, such that there does not exist a program $P \in \mathcal{P}(S_{es})$ satisfying both the conditions (iii) and (iv) above.

Definition 3. An *RSILP–E system in the single–predicate learning context* (abbreviated as RSILP–ES system) is an RSILP–E system, whose *universe* E is such that all examples (ground facts) in E use only one predicate, also known as the *target predicate.*

A *declarative bias* [] biases or restricts the set of acceptable hypotheses, and is of two kinds: *syntactic bias* (also called *language bias*) that imposes restrictions on the form (syntax) of clauses allowed in the hypothesis, and *semantic bias* that imposes restrictions on the meaning, or the behaviour of hypotheses.

Definition 4. An *RSILP–ES system with declarative bias* (abbreviated as RSILP–ESD system) is a tuple $S = (S', L)$, where
(i) $S' = (E, B)$ is an RSILP–ES system, and
(ii) L is a declarative bias, which is any restriction imposed on the hypothesis space $\mathcal{H}(E, B)$.
We also write $S = (E, B, L)$ instead of $S = (S', L)$.

For any RSILP–ESD system $S = (E, B, L)$, let
$\mathcal{H}(S) = \{H \in \mathcal{H}(E, B) \mid H$ is allowed by $L\}$, and
$\mathcal{P}(S) = \{P = B \wedge H \mid H \in \mathcal{H}(S)\}$.
$\mathcal{H}(S)$ (also written as $\mathcal{H}(E, B, L)$) is called the *hypothesis space* induced from S (or from E, B, and L). $\mathcal{P}(S)$ (also written as $\mathcal{P}(E, B, L)$) denotes the set of all the programs induced by S, and is called the *program space* induced from S (or from E, B, and L).

4.2 Equivalence relation, elementary sets and composed sets

We now define an equivalence relation on the universe of an RSILP–ESD system.

Definition 5. Let $S = (E, B, L)$ be an RSILP–ESD system. An indiscernibility relation of S, denoted by $R(S)$, is a relation on E defined as follows: $\forall x, y \in E$, $(x, y) \in R(S)$ iff
$(P \vdash x \Leftrightarrow P \vdash y)$ for any $P \in \mathcal{P}(S)$ (i.e. iff x and y are inherently indistinguishable by any induced logic program P in $\mathcal{P}(S)$).

The following fact follows directly from the definition of $R(S)$.

Fact 4.1 *For any RSILP–ESD system S, $R(S)$ is an equivalence relation.*

Definition 6. Let $S = (E, B, L)$ be an RSILP–ESD system. An *elementary set* of $R(S)$ is an equivalence class of the relation $R(S)$. For each $x \in E$, let $[x]_{R(S)}$ denote the elementary set of $R(S)$ containing x. Formally,
$[x]_{R(S)} = \{y \in E \mid (x, y) \in R(S)\}$.
A *composed set* of $R(S)$ is any finite union of elementary sets of $R(S)$.

Definition 7. An RSILP–ESD system $S = (E, B, L)$ is said to be in a *rough setting* iff
$\exists e^+ \in E^+ \, \exists e^- \in E^- \, [\, (e^+, e^-) \in R(S) \,]$.

4.3 Rough declarative biases

By (E, B, ϕ), we denote an RSILP–ESD system whose universe and background knowledge are E and B, respectively, and which does not have any declarative bias. We also write (E, B, ϕ) as (S, ϕ) where $S = (E, B)$.

For any RSILP–ES system $S = (E, B)$, let $\mathcal{H}^{se}(S) = \{ \{e\} \mid e \in E\}$ and $\mathcal{P}^{se}(S) = \{P = B \wedge H \mid H \in \mathcal{H}^{se}(S)\}$. Let $E_B = \{e \in E \mid B \vdash e\}$.

Fact 4.2 *Let $S' = (S, L)$ be any RSILP–ESD system such that $\mathcal{H}^{se}(S) \subseteq \mathcal{H}(S')$. Every elementary set of $R(S')$, other than E_B, is singleton.*

Proof. Let $S = (E, B)$. For each $P \in \mathcal{P}^{se}(S)$, $P \vdash e \wedge P \not\vdash e'$ for all $e' (\neq e)$ in E, where $P = B \wedge H$, $H = \{e\}$, $e \in E - E_B$. Hence the fact follows. □

Fact 4.3 *Let $S^{\phi} = (S, \phi)$ for any RSILP–ES system $S = (E, B)$. Every elementary set of $R(S^{\phi})$, other than E_B, is singleton.*

Proof. We note that $\mathcal{H}(S^{\phi}) = \mathcal{H}(S)$ and therefore $\mathcal{H}^{se}(S) \subseteq \mathcal{H}(S^{\phi})$, since $\mathcal{H}(S)$ is the set of all possible hypotheses that can be induced from E and B. Using Fact 4.2 we get this fact. □

Some declarative bias $L_{R'}$ is needed to be able to have an RSILP–ESD system in a rough setting. In other words, E and B could be such that a rough setting is possible for some $L_{R'}$, but without some such $L_{R'}$, S is not in a rough setting. E and B are what we would normally consider input or data in the system. So the input or data could be 'rough', but the system will still not be in a rough setting without some declarative bias $L_{R'}$.

We illustrate these with a simple illustration. Let $S = (E, B, \phi)$, with $E = \{p(x), p(y)\}$ and $B = \{data(x, a), data(y, a)\}$. It is to be noted that S is not in a rough setting, even though the input or data appears to be 'rough'.

In section 5.1 later, we study several such $L_{R'}$s. The following definitions are useful in that study.

Definition 8. Let L_R be a declarative bias such that, for some RSILP–ESD system $S = (E, B, L_R)$, at least one elementary set of $R(S)$ is not singleton. L_R is called a *rough declarative bias.*

It is to be noted in the above definition that if the non–singleton elementary set contains $e^+ \in E^+$ and $e^- \in E^-$, then S is in a rough setting.

Definition 9. Let L_G be a declarative bias such that for all E with $|E| \geq 2$ (where for any set A, $|A|$ denotes the cardinality of A), there is some B such that at least one elementary set of $R(S)$ is not singleton, where $S = (E, B, L_G)$ is an RSILP–ESD system. L_G is called a *globally rough declarative bias.*

We now define a combination of declarative biases.

Let $S = (E, B)$ be an RSILP–ES system. Let L_1, L_2 and L_3 be declarative biases. $L_1 \wedge L_2$ (resp., $L_1 \vee L_2$) denotes the declarative bias such that $\mathcal{H}(S') = \mathcal{H}(S_1) \cap \mathcal{H}(S_2)$ (resp., $\mathcal{H}(S'') = \mathcal{H}(S_1) \cup \mathcal{H}(S_2)$), where $S' = (E, B, L_1 \wedge L_2)$, $S'' = (E, B, L_1 \vee L_2)$, $S_1 = (E, B, L_1)$ and $S_2 = (E, B, L_2)$ are RSILP–ESD systems.

$L_1 \wedge L_2 \wedge L_3$ (resp., $(L_1 \wedge L_2) \vee L_3$) denotes the declarative bias such that $\mathcal{H}(S''') = \mathcal{H}(S_1) \cap \mathcal{H}(S_2) \cap \mathcal{H}(S_3)$ (resp., $\mathcal{H}(S'''') = (\mathcal{H}(S_1) \cap \mathcal{H}(S_2)) \cup \mathcal{H}(S_3)$), where $S''' = (E, B, L_1 \wedge L_2 \wedge L_3)$, $S'''' = (E, B, (L_1 \wedge L_2) \vee L_3)$, $S_1 = (E, B, L_1)$, $S_2 = (E, B, L_2)$ and $S_3 = (E, B, L_3)$ are RSILP–ESD systems. $L_1 \vee L_2 \vee L_3$, $(L_1 \vee L_2) \wedge L_3$, ..., etc. are defined similarly.

4.4 An illustrative example

We use the following simple example to illustrate these definitions. Consider the ILP system in the example setting as defined in Definition 1. Let $S = (E, B)$ where
$E = E^+ \cup E^-$,
$E^+ = \{p(d1), p(d2), p(d3)\}$,
$E^- = \{p(d4), p(d5), p(d6)\}$ and
$B = \{atom(d1, c), atom(d2, c), atom(d3, o), atom(d4, o), atom(d5, n), atom(d6, n)\}$. It is seen that for all $e^- \in E^-$, $B \not\vdash e^-$, and for some $e^+ \in E^+$, $B \not\vdash e^+$. (Two conditions (i) and (ii) of an ILP system in the example setting hold.) Let $H = \{p(d1), p(d2), p(d3)\}$. Then for all $e^- \in E^-$, $B \wedge H \not\vdash e^-$, and for all $e^+ \in E^+$, $B \wedge H \vdash e^+$. (Two conditions (iii) and (iv) in the paragraph just below definition 1 also hold.) It is seen that the ILP system can exactly describe the set of positive examples, but in a manner that is not very useful, since the hypothesis is the same as the positive example ground facts.

If $d1, \ldots, d6$ are not allowed in H, then with $H = \{p(A) \leftarrow atom(A, c)\}$, for all $e^- \in E^-$, $B \wedge H \not\vdash e^-$. However it is not true that for all $e^+ \in E^+$, $B \wedge H \vdash e^+$, since $B \wedge H \not\vdash p(d3) \in E^+$. (Condition (iii) holds, but not condition (iv).)

With $H = \{p(A) \leftarrow atom(A, c), p(A) \leftarrow atom(A, o)\}$, for all $e^+ \in E^+$, $B \wedge H \vdash e^+$. However it is not true that for all $e^- \in E^-$, $B \wedge H \not\vdash e^-$, since $B \wedge H \vdash p(d4) \in E^-$. (Condition (iv) holds, but not condition (iii).)

This is formalised in the definition of the RSILP–ESD system (Definition 4). Let $S = (E, B, L)$ where E and B are as given above, and L is the declarative bias such that $d1, \ldots, d6$ is not a term in $q(\ldots)$ for any $H \in \mathcal{H}(S)$, any $C \in H$, and any predicate $q(\ldots) \in C$.

The equivalence relation $R(S)$ is defined in Definition 5 and we see that $R(S) = \{ (p(d1), p(d2)), (p(d2), p(d1)), (p(d3), p(d4)), (p(d4), p(d3)), (p(d5), p(d6)), (p(d6), p(d5)) \}$.

The elementary sets of $R(S)$ (Definition 6) are $\{p(d1), p(d2)\}$, $\{p(d3), p(d4)\}$, $\{p(d5), p(d6)\}$.

The composed sets of $R(S)$ are
$\{\}$, $\{p(d1), p(d2)\}$, ..., $\{p(d1), p(d2), p(d3), p(d4)\}$, ...,
$\{p(d1), p(d2), p(d3), p(d4), p(d5), p(d6)\}$.

S is in a rough setting (Definition 7) since $p(d3) \in E^+$, $p(d4) \in E^-$ and $(p(d3), p(d4)) \in R(S)$.

In the study of the formation of these elementary sets, one needs to show that for some $x, y \in E$, for any $P \in \mathcal{P}(S)$, $P \vdash x \Leftrightarrow P \vdash y$, so that $(x, y) \in R(S)$ (for a special case, it is shown in Section 5.2 that this will be done by an equivalent *operational* check).

4.5 Consistency and completeness in the gRS–ILP model

Let $S = (E, B, L)$ be an RSILP–ESD system, and $\mathcal{P}(S)$ the program space induced by S, as defined earlier.

Definition 10. The *upper approximation* of S, $Upap(S)$, is defined as the least composed set of $R(S)$ such that $E^+ \subseteq Upap(S)$.

Definition 11. The *lower approximation* of S, $Loap(S)$, is defined as the greatest composed set of $R(S)$ such that $Loap(S) \subseteq E^+$.

The set $Bndr(S) = Upap(S) - Loap(S)$ is known as the *boundary region* of S (or the *borderline region* of S). The lower approximation of S, $Loap(S)$, is also known as $Pos(S)$, the *positive region* of S. The set $Neg(S) = E - Upap(S)$ is known as the *negative region* of S.

Definition 12. The *consistent program space* $\mathcal{P}_{cons}(S)$ of S is defined as
$\mathcal{P}_{cons}(S) = \{P \in \mathcal{P}(S) \mid P \nvdash e^-, \forall e^- \in E^-\}$.
A program $P \in \mathcal{P}(S)$ is *consistent* with respect to S iff $P \in \mathcal{P}_{cons}(S)$.
The *reverse–consistent program space* $\mathcal{P}_{rev-cons}(S)$ of S is defined as
$\mathcal{P}_{rev-cons}(S) = \{P \in \mathcal{P}(S) \mid P \nvdash e^+, \forall e^+ \in E^+\}$.
A program $P \in \mathcal{P}(S)$ is *reverse–consistent* with respect to S iff $P \in \mathcal{P}_{rev-cons}(S)$.

Consistency is useful with respect to a positive region and its dual, reverse–consistency, is useful with respect to a negative region.

Definition 13. The *complete program space* $\mathcal{P}_{comp}(S)$ of S is defined as
$\mathcal{P}_{comp}(S) = \{P \in \mathcal{P}(S) \mid P \vdash e^+, \forall e^+ \in E^+\}$.
A program $P \in \mathcal{P}(S)$ is *complete* with respect to S iff $P \in \mathcal{P}_{comp}(S)$.

Definition 14. The *cover* of a program $P \in \mathcal{P}(S)$ in S is defined as
$cover(S, P) = \{e \in E \mid P \vdash e\}$.

The following facts follow directly from the definitions.

Fact 4.4 $\forall P \in \mathcal{P}_{cons}(S)$, $cover(S, P) \subseteq Loap(S)$.

Fact 4.5 $\forall P \in \mathcal{P}_{comp}(S)$, $cover(S, P) \supseteq Upap(S)$.

Fact 4.6 $\forall P \in \mathcal{P}_{comp}(S)$, $(E - cover(S, P)) \subseteq (E - Upap(S))$.

Fact 4.7 $\forall P \in \mathcal{P}_{rev-cons}(S)$, $cover(S, P) \subseteq (E - Upap(S))$.

For a program $P \in P_{cons}(S)$, the closer to $Loap(S)$ P is, the better P is. P is *best* when $cover(S, P) = Loap(S)$. Similarly, for a program $P \in P_{rev-cons}(S)$ (resp., $P \in P_{comp}(S)$), the closer to $U - Upap(S)$ (resp., $Upap(S)$) P is, the better P is, and P is *best* when $cover(S, P) = U - Upap(S)$ (resp., $cover(S, P) = Upap(S)$).

Fact 4.8 $\forall P \in \mathcal{P}_{cons}(S)$, $P \vdash e \Rightarrow e \in E^+$.

Fact 4.9 $\forall P \in \mathcal{P}_{comp}(S)$, $P \nvdash e \Rightarrow e \in E^-$.

Fact 4.10 $\forall P \in \mathcal{P}_{rev-cons}(S)$, $P \vdash e \Rightarrow e \in E^-$.

These facts are used in the definitive description of data in a rough setting. Definitive description involves the description of the data with 100% accuracy. In a rough setting, it is not possible to definitively describe the entire data, since some of the positive examples and negative examples (of the concept being described) inherently cannot be distinguished from each other. These facts show that definitive description is possible in a rough setting when an example is covered by a consistent program (the example is then definitely positive), covered by a reverse–consistent program (the example is then definitely negative), or not covered by a complete program (the example is then definitely negative). In many practical implementations, it is easy to find a consistent program (and therefore, also a reverse–consistent program), whereas it is not so easy to find a complete program. So in practical applications of definitive description, consistent and reverse–consistent programs are easier to use than consistent and complete programs.

Let $S = (E, B, L)$ be the same as in the illustrative example of Section 4.4. We see that
$Upap(S) = \{p(d1), p(d2), p(d3), p(d4)\}$ and
$Loap(S) = \{p(d1), p(d2)\}$.

The following are examples of consistent, complete and reverse–consistent programs.
$P = B \wedge H$ is a consistent program, when $H = \{p(A) \leftarrow atom(A, c)\}$.
$P = B \wedge H$ is a complete program, when $H = \{p(A) \leftarrow atom(A, c), p(A) \leftarrow atom(A, o)\}$.
$P = B \wedge H$ is a reverse–consistent program, when $H = \{p(A) \leftarrow atom(A, n)\}$.

5 A restricted RSILP system

A *restricted* RSILP–ES system is defined by placing certain restrictions on an RSILP–ES system.

Definition 15. A *restricted RSILP–ES system* (abbreviated as R–RSILP–ES system) is an RSILP–ES system $S = (E, B)$, where
(i) the target predicate p used in the universe E is a unary predicate, and
(ii) B is a background knowledge that (a) has only ground unit clauses and (b) has no example from E.
An *R–RSILP–ES system with declarative bias* (abbreviated as R–RSILP–ESD system) is a tuple $S' = (S, L)$ where
(i) $S = (E, B)$ is an R–RSILP–ES system, and
(ii) L is a declarative bias.
$S' = (S, L)$, where $S = (E, B)$, is also written as $S' = (E, B, L)$.

In an RSILP–ESD system, it is quite difficult to determine when a rough setting occurs and when any two given examples in the system are together in the same elementary set. The R–RSILP–ESD is introduced to enable the study of these issues in an easier manner.

The R–RSILP–ESD system, despite the restrictions, is powerful enough to model several practical ILP applications, including a discrete version of the classic *mutagenesis* application [], used in the experimental illustration described later in Section 7.

5.1 Various declarative biases

We consider different declarative biases in R–RSILP–ESD systems and study whether each of them is a globally rough declarative bias.

The following fact follows directly from Fact 4.2.

Fact 5.1 *Let $S' = (S, L)$ be any R–RSILP–ESD system such that $\mathcal{H}^{se}(S) \subseteq \mathcal{H}(S')$. Every elementary set of $R(S')$ is singleton.*

We here define two underlying declarative biases that are used in the following sections.
Let L_{pi} be the declarative bias such that for any R–RSILP–ESD system $S = (E, B, L_{pi})$,
$H \in \mathcal{H}(S) \Rightarrow$ head predicate of C is the target predicate, for any $C \in H$ (predicate invention is not allowed), and
let L_{rd} be the declarative bias such that for any R–RSILP–ESD system $S = (E, B, L_{rd})$,
$H \in \mathcal{H}(S) \Rightarrow$ head predicate of C is not in the body of C, for any $C \in H$ (directly recursive definition is not allowed).

5.1.1 The declarative bias L_{eu}

Let L_{eu} be a declarative bias such that for any R–RSILP–ESD system $S = (E, B, L_{eu})$, $H \in \mathcal{H}(S) \Rightarrow e \notin C$ for any $e \in E$ and any $C \in H$, i.e. no example is used in H.

Fact 5.2 *The declarative bias $L_{eu} \wedge L_{pi} \wedge L_{rd}$ is a globally rough declarative bias.*

Proof. Consider any R–RSILP–ESD system $S = (E, B, L_{eu} \wedge L_{pi} \wedge L_{rd})$, such that $B = \phi$. We see that $R(S) = E^2$ for any E, i.e. for any H, either $H \vdash$ all e or $H \nvdash$ any e. Therefore, by the definition of globally rough declarative bias, we have the fact. □

5.1.2 The declarative bias L_{te}

Let L_{te} be a declarative bias such that for any R–RSILP–ESD system $S = (E, B, L_{te})$, $H \in \mathcal{H}(S) \Rightarrow x$ is not a term in $q(\ldots)$ for any $C \in H$, any $q(\ldots) \in C$, and any x such that $p(x) \in E$, where p is the target predicate of S.

Fact 5.3 *The declarative bias L_{te} is a globally rough declarative bias.*

Proof. The proof is the same as the proof of Fact 5.2. □

The biases studied in this section (L_{pi}, L_{rd}, L_{eu} and L_{te}) are more fundamental biases than traditional biases such as i, j–determinacy [] since they deal with more fundamental restrictions such as restrictions on predicate invention, recursive definition, and the presence of examples in the hypotheses.

We see that the RSILP–ESD system $S = (E, B, L)$ used in the illustrative example of Section 4.4 is an R–RSILP–ESD system according to Definition 15. L is the declarative bias L_{te}.

5.2 Formation of elementary sets under different declarative biases

In this section we consider R–RSILP–ES systems with different declarative biases and study the conditions under which two examples are in the same elementary set.

Definition 16. Let x and y be ground terms. Let $a = q(t_1, \ldots, t_n)$ and $b = q(u_1, \ldots, u_n)$ be ground atoms, where q is an n–arity predicate for some $n \geq 1$. For each $i \in \{1, \ldots, n\}$, we refer to the tuple (t_i, u_i) as $place\_pair(a, b, i)$. The two ground atoms a and b are called *(x,y)–paired* (also called *(y,x)–paired*) iff for any $i \in \{1, \ldots, n\}$, $(t_i, u_i) = place\_pair(a, b, i)$ is such that $(t_i = x) \Leftrightarrow (u_i = y)$.
The two ground atoms a and b are called *(x,y)–equivalent* (also called *(y,x)–equivalent*) iff for any $i \in \{1, \ldots, n\}$, $(t_i, u_i) = place\_pair(a, b, i)$ is such that (i) $(t_i = x) \Leftrightarrow (u_i = y)$ and (ii) $t_i = u_i$ when $t_i \neq x$.

A set V of ground atoms is called *(x,y)–paired* (also called *(y,x)–paired*) iff for every ground atom a ($\in V$) that has x *or* y as a term, there exists a ground atom b ($\in V$) such that a and b are (x,y)–paired. A set V' of ground atoms is called *(x,y)–equivalent* (also called *(y,x)–equivalent*) iff for every ground atom a ($\in V'$) that has x *or* y as a term, there exists a ground atom b ($\in V'$)

such that a and b are (x,y)–equivalent.

For example, let $V = \left\{ \begin{array}{c} q(a,x,b,x,c), \\ q(d,y,e,y,f), \\ q(g,z,h,z,i) \end{array} \right\}$. V is (x,y)–paired, (y,z)–paired and (x,z)–paired. V is also (a,d)–paired, (b,e)–paired, (c,f)–paired, (d,g)–paired, (e,h)–paired ....

Let $V = \left\{ \begin{array}{c} q(a,x,b,x,c), \\ q(a,y,b,y,c), \\ q(a,z,b,z,c) \end{array} \right\}$. V is (x,y)–equivalent, (y,z)–equivalent and (x,z)–equivalent.

5.2.1 Elementary sets when the declarative bias is $L_{pi} \wedge L_{rd} \wedge L_{eu}$

Proposition 5.1 *Let $P \in \mathcal{P}(S)$ for an R–RSILP–ESD system $S = (E, B, L_{pi} \wedge L_{rd} \wedge L_{eu})$, and $p(x), p(y) \in E$. (i) If B is (x,y)–equivalent, then in the resolution tree that shows that $P \vdash p(x)$, every occurrence of x can be replaced with y to result in a resolution tree that shows $P \vdash p(y)$, using only clauses in P. (ii) If B is (x,y)–equivalent, then in the resolution tree that shows that $P \vdash p(y)$, every occurrence of y can be replaced with x to result in a resolution tree that shows $P \vdash p(x)$, using only clauses in P.*

Proof. We first prove (i) above.
We consider the different clauses that can occur in P and therefore be used in the resolution tree. $P = B \wedge H$ consists of all the clauses from B and all the clauses from H. Consider the different types of clauses that can occur in the hypothesis H. Clauses in H can be unit clauses or non–unit clauses.
We first consider unit clauses. Unit clauses can either be ground or non-ground. Ground unit clauses are not allowed in H, due to L_{pi} and L_{eu}. Non–ground unit clauses in H use only the unary target predicate, due to L_{pi}. The presence of a non–ground unit clause in H using the unary target predicate implies that $P \vdash p(x) \wedge P \vdash p(y)$.
We then consider non–unit clauses. Only one clause of H will appear in the resolution tree due to L_{pi} and L_{rd}. This clause will not have the target predicate in its body due to L_{rd}. This clause will not have a ground target predicate as its head due to L_{eu}. Hence this clause can be used both in the resolution tree of $P \vdash p(x)$ and in the resolution tree of $P \vdash p(y)$.

We now consider the clauses in the resolution tree belonging to B. These are ground unit clauses. Since B is (x,y)–equivalent, any clause in B that has x as some of its terms will have an identical clause, with y replacing x. Hence, any clause in the resolution tree with x as some of its terms can be replaced by the corresponding clause with y in the places of x. Any clause in the resolution tree that does not have x as any of its terms can be used in both resolution trees.

A similar discussion proves (ii) above. □

Fact 5.4 *Let $S = (E, B, L_{pi} \wedge L_{rd} \wedge L_{eu})$ be an R–RSILP–ESD system. Then, for any $p(x), p(y) \in E$, $(p(x), p(y)) \in R(S) \iff B$ is (x,y)–equivalent.*

Proof. We first prove the necessary condition $\Longrightarrow$:
Let us assume that B is not (x,y)–equivalent. Then, for some $q(t_1, \ldots, t_n) \in B$ such that $t_i = x$ for any $i \in I$, and $t_j \neq x$ for $j \in \{1, \ldots, n\} - I$, for some non–empty $I \subseteq \{1, \ldots, n\}$,
there does not exist a $q(u_1, \ldots, u_n) \in B$ such that $u_i = y$ for any $i \in I$ and $u_j = t_j$ for any $j \in \{1, \ldots, n\} - I$.
Consider the hypothesis $H = p(X) \leftarrow q(v_1, \ldots, v_n) \in \mathcal{H}(S)$, where $v_i = X$ for any $i \in I$ and $v_j = t_j$ for any $j \in \{1, \ldots, n\} - I$. For the program $P = B \wedge H$, $P \vdash p(x)$ and $P \nvdash p(y)$. So $(p(x), p(y)) \notin R(S)$.
This completes the proof of necessity.

We now prove the sufficient condition $\Longleftarrow$:
We first prove that if B is (x,y)–equivalent, then $P \vdash p(x) \Rightarrow P \vdash p(y)$ for any $P = B \wedge H \in \mathcal{P}(S)$.
Consider the resolution tree that shows that $P \vdash p(x)$. Replace every occurence of x (in the tree) with y. The new tree consists of clauses from B and H, and shows that $P \vdash p(y)$ (as proved in Proposition 5.1).
A similar discussion shows that if B is (x,y)–equivalent, then $P \vdash p(y) \Rightarrow P \vdash p(x)$ for any $P \in \mathcal{P}(S)$.
Thus we have showed that if B is (x,y)–equivalent, then $P \vdash p(x) \Leftrightarrow P \vdash p(y)$ for any $P \in \mathcal{P}(S)$.
This completes the proof of sufficiency. □

It is to be noted that Fact 5.4 gives an *operational* way to check if two examples are in the same elementary set, for $L = L_{pi} \wedge L_{rd} \wedge L_{eu}$.

5.2.2 Elementary sets when the declarative bias restricts certain predicate places in H to have only variables

We now consider the following illustration. Let $S = (E, B, L)$ be an R–RSILP–ESD system with the example set $E = \{active(d1), active(d2)\}$, the background knowledge $B = \{bond(d1, d1\_1, d1\_2, b7), bond(d2, d2\_1, d2\_2, b7)\}$, and the declarative bias L as defined below.

Let the declarative bias L_2 be such that in any hypothesis $H \in \mathcal{H}(E, B, L_2)$ the predicate $bond$ should have only variable terms in places 2 and 3. Let $L = L_2 \wedge L_{pi} \wedge L_{rd} \wedge L_{eu}$. Then the examples $active(d1)$ and $active(d2)$ belong to the same elementary set of $R(S)$. The mode declaration of ILP system Progol [] can specify this bias L_2.

If the declarative bias L_2 were not a part of L (i.e. $L = L_{pi} \wedge L_{rd} \wedge L_{eu}$), then $P = B \wedge H$, with $H = \{active(X) \leftarrow bond(X, d1\_1, d1\_2, b7)\}$, will distinguish $active(d1)$ from $active(d2)$ (i.e. $P \vdash active(d1)$ and $P \nvdash active(d2)$).

We now formally define declarative biases like L_2 illustrated above. Let V be any set of ground atoms. Let $pred(V)$ denote the set of predicate symbols used in V. For each $A \subseteq pred(V)$, let $V_A = \{q(\ldots) \in V \mid q \in A\}$, and $placelist(A) = \{(q,i) \mid q \in A$, and $1 \leq i \leq n_q$ where n_q is the arity of $q\}$.

Let B be any background knowledge of the R–RSILP–ES system. For each $Z \subseteq placelist(A)$, where $A \subseteq pred(B)$, let L_Z be the declarative bias such that, for any universe E of the R–RSILP–ES system:
$\forall H \in \mathcal{H}(E,B,L_Z)$, $\forall C \in H$
$[q(t_1,\ldots,t_n) \in C \Rightarrow [q \in A \wedge \forall i \in \{1,\ldots,n\}\ [(q,i) \in Z \Rightarrow t_i$ is a variable]]].

Let x and y be ground terms. Let $a = q(t_1,\ldots,t_n)$ and $b = q(u_1,\ldots,u_n)$ be ground atoms where q is an n–arity predicate for some $n \geq 1$. Let $Z \subseteq placelist(pred(\{a,b\}))$. The two ground atoms a and b are called *(x,y)–equivalent except Z* iff for any $i \in \{1,\ldots,n\}$, (i) $(t_i = x) \Leftrightarrow (u_i = y)$ and (ii) if $(q,i) \notin Z$ and $t_i \neq x$, $t_i = u_i$.

For any set V of ground atoms, any ground terms x and y, and any $Z \subseteq placelist(pred(V))$, the set V is called *(x,y)–equivalent except Z* iff for every ground atom a $(\in V)$ that has x or y as a term, there exists a ground atom b $(\in V)$ such that a and b are (x,y)–equivalent except $placelist(pred(\{a,b\})) \cap Z$.

Fact 5.5 *Let $S = (E,B,L_Z \wedge L_{pi} \wedge L_{rd} \wedge L_{eu})$ be any R–RSILP–ESD system where $Z \subseteq placelist(A)$ and $A \subseteq pred(B)$. Let $B_A = \{q(\ldots) \in B \mid q \in A\}$. Then, for any $p(x), p(y) \in E$,*
$(p(x),p(y)) \in R(S)$ if
(i) B_A is (x,y)–equivalent except Z and
(ii) for each (x,y)–paired $a, b \in B_A$ and for each i such that $(q,i) \in Z$ (where q is the predicate symbol of a and b), B_A is (u,v)–equivalent, where $(u,v) = place\_pair(a,b,i)$.

Proof. The proof is similar to the proof of sufficiency of Fact 5.4. □

We again consider the illustration used earlier in this section. In this illustration, let $A = pred(B)$. Then we see that $A = \{bond\}$, $B_A = B$,
$placelist(A) = \{(bond,1),(bond,2),(bond,3),(bond,4)\}$.
Let
$Z = \{(bond,2),(bond,3)\}$. We see that $R(S) = \{(active(d1),active(d2))\}$.
B is $(d1,d2)$–equivalent except Z since both $bond(d1,d1\_1,d1\_2,b7)$ and
$bond(d2,d2\_1,d2\_2,b7)$ have $b7$ in place 4. This fulfils condition (i) of Fact 5.5. Condition (ii) does not play a role in this example since A is a singleton.

6 Comparison with other learning paradigms

6.1 Version spaces

The concepts of rough set theory are different from those of version spaces. Version spaces consider a general hypothesis and a specific hypothesis that are both consistent with respect to both positive and negative evidence. Both the general and the specific hypothesis cover all the positive evidence without covering any of the negative evidence. This contrasts with the rough setting where a hypothesis that covers all the positive evidence also covers some of the negative evidence.

6.2 Learning from positive data alone

The rough setting is not applicable to learning from positive data alone. A rough setting exists when all the positive examples are covered by a hypothesis only when some of the negative examples are also covered. The inconsistency in the data environment occurs due to the presence of both positive and negative examples, and so cannot occur in the paradigm of learning from positive examples alone.

6.3 Associated probabilities or thresholds

The rough set model handles inconsistencies in the examples without the use of any additional data such as associated probabilities or thresholds. The model uses only the examples and the knowledge about the examples, and does not need the aid of any additional information such as probabilities.

6.4 The fuzzy–set rough–set ILP system EAGLE

The EAGLE system [, ,] is a fuzzy–set rough–set ILP system which uses a pre–processing phase that precedes the learning, where fuzzy sets are used to discretize numerical features and model uncertainty within data. Then an inductive learning process is performed to generate fuzzy relational definitions of a target concept in terms of others. Three main steps are used to achieve this learning goal: (1) partitioning of the learning data, (2) approximation of the target concept to learn and finally (3) generation of definitions.

The use of fuzzy set theory to handle quantitative numeric values makes the Eagle a powerful tool. Rough set theory is used in the Eagle to make learning easier by using only a portion of the knowledge available. This increases performance tremendously. However the Eagle does not model roughness itself. It is only a crisp model (even though fuzzy), and not a rough model. Grouping into granules takes the label of the example (positive or negative) into consideration, and so a granule consists of only

positive examples or only negative examples (and thus a granule can never have both a positive example and a negative example). Roughness, in the classical rough set theory sense, does not occur. Hence a rough setting, as defined in the gRS–ILP model, does not occur. (Upper and lower approximations as used in the Eagle represent the maximal and the minimal amounts of data available about the target concept and is used to restrict the size of the knowledge used for learning. This is in contrast to the classical rough set theory representation of upper and lower approximations (with respect to the examples) that is used in the gRS–ILP model.)

6.5 PAC–learning and ILP

A subset of the universe is said to be *shattered* by a concept class [,] when every subset of this subset is describable by a concept in the concept class. The *Vapnik–Chervonenkis (VC) dimension of a concept class* is the cardinality of the largest finite subset of the universe that is shattered by the concept class. If arbitrarily large finite sets are shattered, the VC–dimension of the concept class is infinite. Using the terminology followed in our paper, the example space is said to be shattered by the hypothesis space when every subset of the example space is describable by a hypothesis in the hypothesis space. In other words, every elementary set has to be a singleton for this to occur. Hence the VC–dimension can be considered to be the number of elementary sets. The application of PAC–learning [] (Probably Approximately Correct learning) concepts to ILP has been studied in several papers [,]. In a rough setting, both positive and negative examples are in the same elementary set, and hence the accuracy of learning is restricted by the numbers of positive and negative examples together in the same elementary set (in relationship to the total size of the universe).

6.6 Rough logic

Several extensions of rough set theory to first order logic are reported in literature [,]. Another approach to rough set inductive logic programming would be the extension of ILP principles and methods to a rough set first order logic.

7 Experimental illustration

The gRS–ILP model has useful applications in the definitive description of large data. Knowledge discovery in databases is the non–trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data([]). This usually involves one of two different goals: prediction and description. Prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest. Description focuses on finding human–interpretable patterns describing the data.

Progol is an Inductive Logic Programming system written in C by Dr. Muggleton []. The syntax for examples, background knowledge and hypothesis is Dec-10 Prolog. Headless Horn clauses are used to represent negative examples and constraints. Progol source code and example files are freely available (for academic research) from `ftp.cs.york.ac.uk` under the appropriate directory in `pub/ML_GROUP`.

The following experimental illustration of definitive description using the gRS–ILP model uses the *mutagenesis* data set available by ftp access from `ftp.comlab.ox.ac.uk` in `pub/Packages/ILP/Datasets/mutagenesis`. These experiments concern the discovery of rules for mutagenicity in nitroaromatic compounds described in [] using the data reported in []. Progol version 4.2 dated 14.02.98 is used.

Some changes are made in the mode settings with respect to those described in []. A discrete charge (*dcharge*, which is p or m depending on whether the charge is $\geq$ or < 0) is used instead of the real–valued charge. The predicate *datm* is used in any clause of the hypothesis only for atoms that are part of a *bond* that is already in that clause.

Positive (E^+) and negative (E^-) examples ($E = E^+ \cup E^-$) use the predicate `active` to indicate that a compound is mutagenic. Background knowledge B uses the predicates `bond` and `datom` to describe the type of the bond existing between various pairs of atoms in the compound, and details of the various atoms in the compound, respectively. Appropriate mode declarations are used in Progol to incorporate the required declarative bias $L = L_Z \wedge L_{pi} \wedge L_{rd} \wedge L_{eu}$. (However, L_{eu} is done by hand, since Progol lists uncovered examples in the induced hypothesis.) Let $S = (E, B, L)$. It is thus seen here that an R–RSILP–ESD system with bias $L = L_Z \wedge L_{pi} \wedge L_{rd} \wedge L_{eu}$ is powerful enough to model the classic ILP experiment for mutagenesis.

The *first step* is any conventional Progol experiment using the data set. Conventionally, the aim is to maximise the correct cover of both positive and negative examples (in other words, try to increase the number of positive examples covered and decrease the number of negative examples covered). Let this induced program be known as P for the purpose of this outline.

The *second step* uses Progol with the *default* `noise` setting of zero, where any induced hypothesis is consistent and *cannot* cover *any* negative example. Let this induced consistent program be P_{cons}. The induced hypothesis of P_{cons} follows.

```
active(A) :- bond(A,B,C,b3).
active(A) :- bond(A,B,C,b7), datm(A,C,c,at195,m).
active(A) :- bond(A,B,C,b7), datm(A,C,o,at52,m).
active(A) :- bond(A,B,C,b7), bond(A,D,B,b1),
 datm(A,C,c,at29,p).
active(A) :- bond(A,B,C,b7), bond(A,D,C,b1),
```

```
 datm(A,D,c,at10,m).
active(A) :- bond(A,B,C,b7), datm(A,B,c,at27,m),
 datm(A,C,c,at27,m).
active(A) :- bond(A,B,C,b7), datm(A,B,c,at27,p),
 datm(A,C,c,at27,m).
active(A) :- bond(A,B,C,b7), datm(A,B,c,at29,p),
 datm(A,C,c,at22,p).
```

The *third step* is to determine a reverse–consistent program denoted by $P_{rev-cons}$, by exchanging the roles of E^+, E^-, and then repeating step 2. The induced hypothesis of $P_{rev-cons}$ follows.

```
active(A) :- bond(A,B,C,b1), datm(A,B,c,at16,p).
active(A) :- bond(A,B,C,b1), datm(A,B,f,at92,m).
active(A) :- bond(A,B,C,b1), bond(A,C,D,b1),
 datm(A,B,n,at34,m).
active(A) :- bond(A,B,C,b1), bond(A,C,D,b1),
 datm(A,B,o,at50,m).
active(A) :- bond(A,B,C,b1), bond(A,C,D,b1),
 datm(A,D,n,at36,m).
active(A) :- bond(A,B,C,b1), bond(A,D,E,b1),
 datm(A,B,n,at34,m), datm(A,D,c,at21,p).
active(A) :- bond(A,B,C,b1), bond(A,D,E,b1),
 datm(A,E,cl,at93,m), datm(A,C,f,at92,m).
active(A) :- bond(A,B,C,b1), bond(A,C,D,b1),
 datm(A,B,c,at22,p), datm(A,C,n,at32,m),
 datm(A,D,c,at10,m).
active(A) :- bond(A,B,C,b1), bond(A,D,E,b1),
 datm(A,B,c,at22,p), datm(A,C,h,at3,p),
 datm(A,D,n,at38,p).
```

The results are tabulated below.

$\|E^+\|$	$\|E^-\|$	$\|E\|$	$\|cover(S, P_{cons})\|$	$\|cover(S, P_{rev-cons})\|$
125	63	188	77	22

Using Facts 4.8 and 4.10 we have the following.
If $P_{cons} \vdash e$, then $e \in E^+$.
If $P_{rev-cons} \vdash e$, then $e \in E^-$.
Otherwise P is used:
If $P \vdash e$, then it is very likely that $e \in E^+$;
else if $P \nvdash e$, then it is very likely that $e \in E^-$.

77 out of 125 positive examples are definitively described by P_{cons} and 22 out of 63 negative examples are definitively described by $P_{rev-cons}$.

Earlier systems conventionally do not use P_{cons} and $P_{rev-cons}$. They handle the rough setting by inducing P to maximize correct cover by maximizing the number of positive examples covered and negative examples not covered. However, this does not definitively describe the data, since P cannot say with certainty whether an example definitely belongs to the evidence or not. When the gRS–ILP model is used, P_{cons} and $P_{rev-cons}$ are induced to definitively describe part of the data. The rest of the data can be described by P, but not definitively.

We believe that the gRS–ILP model lays a sound theoretical foundation for an experimental method that can be easily performed on many existing ILP systems. The ILP system should allow the consistency level to be fully consistent. P_{cons} and $P_{rev-cons}$ can be easily determined by such ILP systems with the consistency level being fully consistent.

The gRS–ILP model as described in this paper needs to be extended to use real valued terms. The original Rough Set Theory concepts used in this paper are based on the use of discrete valued attributes and not real valued attributes. It can be seen in this experimental illustration that a discrete charge value of p or n is used to indicate positive or negative charge instead of the real valued charge originally available.

8 Conclusions

In this paper, the formal definitions of the gRS–ILP model are presented, and definitive description in a rough setting discussed. An illustrative experiment of the definitive description of mutagenesis data using the ILP system Progol is presented.

The gRS–ILP model is applied in this paper to the definitive description of data that inherently cannot be described consistently and completely. Traditional experiments using ILP systems usually try to describe such data with as much consistency and completeness as possible. However such a description is not definitive. The description will either describe some positive examples as negative or describe some negative examples as positive. The gRS–ILP model is used to definitively (accurately) describe some of the data. The rest of the data is described (but not accurately) by using the ILP system in the traditional manner.

Further work is to be done in areas such as the following: algorithms for making the consistent program cover the entire lower approximation, the use of real values, and areas other than definitive description, such as prediction.

Acknowledgements

The authors thank anonymous referees of this paper for their useful comments. They also thank Professors S. Miyano, K. Morita, V. Ganapathy, K. M. Mehata, and R. Siromoney for their valuable comments and support; Professor S. Muggleton and Drs. A. Srinivasan and D. Page for the warm welcome and the sharing of their research results during the first author's brief visit to the Oxford University Computing Laboratory; Dr. N. Zhong for help in providing rough set material; Professor H. Motoda for his encouragement and help; and the Japan Society for Promotion of Science for the Ronpaku Fellowship for the first author.

References

[] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Classifying learnable geometric concepts with the Vapnik–Chervonenkis dimension. In *Proc. 18th ACM Symposium on Theory of Computing*, pages 273–282, 1989.

[] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik–Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, 1989.

[] W.W. Cohen. Pac–learning recursive logic programs: efficient algorithms. *Journal of Artificial Intelligence Research*, 2:501–539, 1995.

[] A.K. Debnath, R.L. Lopez de Compadre, G. Debnath, A.J. Schusterman, and C. Hansch. Structure–activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medical Chemistry*, 34:786–797, 1991.

[] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–36. AAAI Press / The MIT Press, 1997.

[] J.U. Kietz and S. Dzeroski. Inductive logic programming and learnability. *SIGART Bulletin*, 5(1):22–32, 1994.

[] T.Y. Lin, Q. Liu, and X. Zuo. Models for first order rough logic applications to data mining. In *Proc. 1996 Asian Fuzzy Systems Symposium, Special Session Rough Sets and Data Mining*, pages 152–157, Taiwan, 1996.

[] Emmanuelle Martienne and Mohamed Quafafou. Learning fuzzy relational descriptions using the logical framework and rough set theory. In *Proc. of the 7th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'98)*, Anchorage, Alaska, May 1998.

[] Emmanuelle Martienne and Mohamed Quafafou. Learning logical descriptions for document understanding: a rough sets-based approach. In *Proc. first International Conference on Rough Sets and Current Trends in Computing (RSCTC'98)*, Warsaw, Poland, June 1998.

[] Emmanuelle Martienne and Mohamed Quafafou. Vagueness and data reduction in learning of logical descriptions. In *Proc. 13th biennal European Conference on Artificial Intelligence (ECAI'98)*, Brighton, UK, August 1998.

[] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and Methods. *Journal of Logic Programming*, 19/20:629–679, 1994.

[] S. Muggleton and C. Feng. Efficient induction in logic programs. In S. Muggleton, editor, *Inductive Logic Programming*, pages 281–298. Academic Press, 1992.

[] S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[] S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[] S. Parsons and M. Kubat. A first–order logic for reasoning under uncertainty using rough sets. *Journal of Intelligent Manufacturing*, 5:211–223, 1994.

[] Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341–356, 1982.

[] Z. Pawlak. *Rough Sets — Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.

[] A. Siromoney and K. Inoue. A framework for Rough Set Inductive Logic Programming — the gRS-ILP model. In *Pacific Rim Knowledge Acquisition Workshop*, pages 201–217, Singapore, November 1998.

[] A. Siromoney. A rough set perspective of Inductive Logic Programming. In Luc De Raedt and Stephen Muggleton, editors, *Proceedings of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming*, pages 111–113, Nagoya, Japan, 1997.

[] A. Srinivasan, S.H. Muggleton, R.D. King, and M.J.E. Sternberg. Theories for mutagenicity: a study of first-order and feature based induction. *Artificial Intelligence*, 85:277–299, 1996.

[] L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11), 1984.

A Meeting Report: Charleston, IL, Consciousness Conference, November 6–7, 1998

Conference initials

The intention of this report is to keep the most significant contents and accompanying events of the conference alive not only for the participants, organizers and supporters, but also for the emerging community of consciousness studies, especially in the field of artificial or informational consciousness, and such implementation trials and possibilities worldwide. If one would imagine that the conference size was rather small than large it was carefully woven together with certain provoking accents in its background. One of such trial was to get philosophically, scientifically, and technologically more transparent under which conditions an implementation of artificial consciousness would become possible.

Figure 1: The initiator and the executive organizer of the conference, Professor Suhrit Kumar Dey, Department of Mathematics, Eastern Illinois University. He was the host chairman and local organizer of the conference

The idea of the conference goes back to June 1997, when Professor S.K. Dey visited Ljubljana and gave a lecture at the Department of Philosophy, University in Ljubljana. The direct question concerning artificial consciousness was put into a vivid discourse at the dinner in the student restaurant "Pod lipco" ("Under a Small Lime Tree") after his lecture. This question was directed to A.P. Železnikar who, at that time was editing technically the special issue of *Informatica* on "Informational Phenomenalism of Consciousness" [1] and was, at that time primarily studying the informational background of phenomenalism.

The question came as a surprise, since Železnikar studied, in fact, the informational problems as such, their general philosophy, new kinds of formalization possibilities, and aplicabality of the theory to the fundamentals of informational problems expression, formalism, conceptualization, "phenomenal fields" (like psychology, psychoanalysis, understanding), and the like [4]. Later, in 1997 [5], the problem of artificial consciousness was tackled by a twodimensional shell using the principles of the general and standardized metaphysicalism.

Pre-conference events

On November 5, Professor Železnikar gave a lecture entitled "Artificial Consciousness" (AC) at the Eastern Illinois University, Department of Mathematics, for students and the research staff of the University. As he said, this was the first kind of such a lecture for him bringing into the foreground the most relevant problems of the AC project implementation. The lecture treated the problem of informational emergence being one of the key problems of the philosophy and theory of the informational in the context of consciousness.

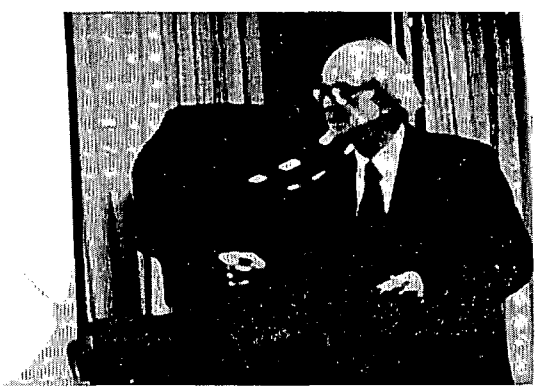


Figure 2: Professor Anton P. Železnikar giving the lecture on November 5, 1998, in the Department of Mathematics, for students and the staff of EIU.

Systematically, the following subjects have been presented: supervenience problems; informational operands, operators, formulas, formula systems and primitive formula systems; formula gestalt and system gestalt; schemes of formulas and formula systems; informational measures; informational frames; graphs of formulas and formula systems; informational experiments; informational axioms concerning emergentism and intentionalism, pretransition, basic transition, constitution, formula system, decomposition, operand rotation, formula cutting, formula system solution upon operands (by formula cutting and operand rotation), schematism, gestaltism, framism, graphism, overlapping of schemes and graphs, shell, experiment, metaphysicalism and topologism; circularism and rotationalism;

decomposition systems; structure, organization and identification of shells; informational phenomenalism of consciousness; premetaphysicalistic concepts and decomposition of consciousness; metaphysicalistic shells of consciousness; characteristic components of consciousness shell; consciousness experiments; informational communication; and informational machine. The lecture was supported by formally and graphically precisely elaborated slides (using enlarged LaTeX format).

The consciousness the lecturer was advocating is an informational consciousness based on a supercomputer-computer net in the worldwide environment of existing database organized libraries, archives, expert systems, and other unforeseeable sources of significant information. Informational consciousness needs an informational operating system giving support to emergentism of informational operands, operators, formulas and formula systems. A strict formalism of concepts was presented, especially the possibilities of AC shells which can be representatives of huge formula systems defined outside of the shell as such. In this case, the shell performs as a pure structural and organizational concept of informational consciousness, being a kind of organizational invariance, as proposed by Chalmers [2].

The conference has been carefully organized by S.K. Dey, his wife Roma and members of the family, J.H. Fetzer, J. Chandler, D. Bhardwaj, J.P. Ziebarth, M. Peruš, A.P. Železnikar, and the personel of Mathematical Department, EIU, and the authorities of Charleston. The great deal of paper evaluation, abstract printing and conversation was organized by M. Peruš from Ljubljana, Slovenia. A separate set of the conference home pages of was designed by Dr. Dheeraj Bhardwaj, India. The abstracts have been printed before the conference in Informatica [7].

Conference course

The conference began the work on November 6, 1998 at 8.45, in Arcola/Tuscola Room, MLK University Union. The welcome address was given by Dr. Lida G. Wall, Dean, College of Sciences, EIU.

Conference subjects (by authors and titles of papers

On the first day of the conference, November 6, 1998, in four sessions 16 papers have been presented in the following order.

1. Structure and organization of consciousness

- *Anton P. Železnikar,* Artificial consciousness
- *Bruse MacLenan,* The protophenomenal structure of consciousness, with special application to the experience of color

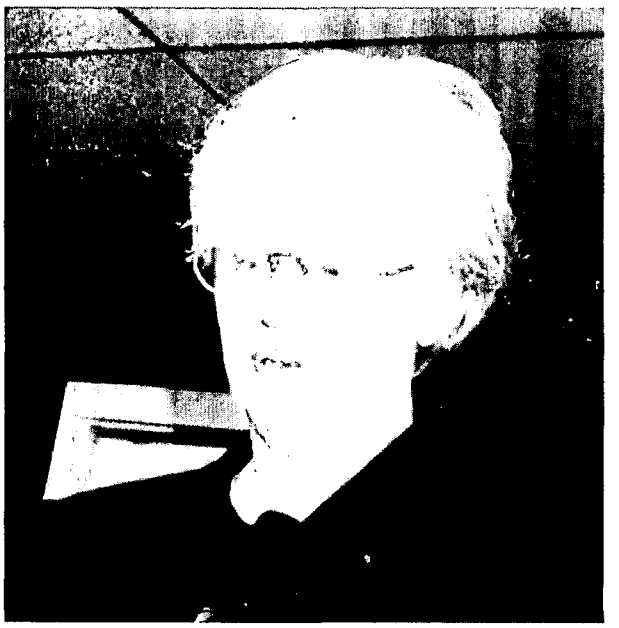


Figure 3: Dr. Lida G. Wall, Dean, College of Sciences, EIU, giving the Welcome Address to the conference participants.

- *Jerry Chandler,* Third-order cybernetics and the evolutionary root of consciousness

2. Consciousness measurements, patterns, states, and learning

- *Edmond Chouinard,* Floating consciousness—a coupling of mind and matter—with related experimental results
- *Frederick Travis,* Physiological patterns during transcendental meditation practice as a basis for a neural model of transcending
- *Kapil Deo Pandey,* Transcendental states of consciousness
- *Sajalendu Dey,* Modeling consciousness as a super class to generate productions: an investigation using the machine learning approach

3. Paradoxes, gateway, logic and theory formation concerning consciousness

- *Ana Pasztor,* Pragmatic paradoxes and the hard problem of consciousness
- *Jazz Rasool,* The nature of the mind-body gateway
- *Jeremy Horne,* Information processing efficiency of logical operators
- *Richard Amoroso,* The utility of the transcendental state as a tool in theory formation

4. Physical and mathematical aspects of consciousness

- *John Hagelin,* Is consciousness the unified field? A field-theorist's perspective
- *Evan Harris Walker,* Consciousness as a factor in the modified Schrödinger equation
- *Richard Amoroso,* The role of gravitation in the dynamics of consciousness
- *James Glazebrook,* Creative evolution in mathematics
- *Mitja Peruš,* Neural and Quantum complex system dynamics as a background of consciousness and cognition

In the evening, a banquet was given honoring Anton P. Železnikar (look at the next section).

On the second day of the conference, November 7, 1998, in Phipps Hall, Science Building, within five sessions 16 papers have been presented.

5. Consciousness-based technologies

- *Kenneth G. Walton,* Consciousness-based technologies in offender rehabilitation: Psychological/physiological benefits and reduced recidivism
- *Jeremy Z. Fields,* Consciousness-based medicine: Breaking the vicious cycle of stress, chronic disease and aging

Figure 4: Professor James H. Fetzer, University of Minnesota, MN, the Editor of *Minds and Machines* and the keynote speaker, at the banquet. His keynote lecture entitled "Consciousness and cognition: Semiotic concepts" was presented in a philosophically rigorous manner.

- *John S. Hagelin,* Natural law and practical applications of consciousness-based technology in government
- *Rachel Spigel Goodman,* International peace initiative through Maharishi Mahesh Yogi's consciousness-based technology

6. Research of higher states of consciousness

- *Daniel Meyer-Dinkgrafe,* Theatre as an aid to development of higher states of consciousness
- *David Orme-Johnson & Kam-Tim So,* Three randomized studies of the effects of the transcendental meditation technique on intelligence: Support for the theory of pure intelligence as a field underlying thought and action
- *Dejan Raković,* EEG-correlates of some states of consciousness: Transcendental meditation, musicogenic states, microwave resonance relaxation, healer/healee interaction and alertness/drowsiness
- *Imants Baruss,* Overview of consciousness research

7. Philosophical and formal concepts of consciousness

- *James H. Fetzer,* Consciousness and cognition: Semiotic concepts
- *Anton P Železnikar,* Informational consciousness in philosophy, science and formalism

8. Miscellaneous concerning consciousness

- *Tony Wright,* The expansion and degeneration of human consciousness
- *Suhrit K. Dey,* Convergence of information in Indian philosophy
- *Dejan Raković,* Transitional states of consciousness as a biophysical basis of transpersonal transcendental phenomena
- *Richard Amoroso,* The feasibility of constructing a conscious quantum computer

9. Some aspects of consciousness

- *Madhu Jain,* Female aspects of consciousness
- *Cyrus F. Nourani,* Intelligent thought trees and conscience science
- *Steven Thaler,* Fragmentation of universe and the devolution of consciousness

Figure 5: Mr. Tony Wright presented *The Expansion and Degeneration of Human Consciousness* from the aspect of tradition and nutrition.

Figure 6: Mr. Mitja Peruš was the speaker at the banquet in honor of Professor Anton P. Železnikar.

Conference banquet

The banquet honoring Dr. Anton P. Železnikar was held at the elite E.L. Krackers Restaurant for a hundred of invited guests. The speaker to the ceremony was Mr. Mitja Peruš, University of Ljubljana, Slovenia, with the speech entitled *All in One, One in All.* He pointed out the following.

We experience everything through our consciousness. Our consciousness or, in a narrower sense, our brain, *co-creates* our world, i.e. the world we experience. We never know anything completely reliable about any *objective* world, except about our subjective world (the first person perspective). This world of our experience is partially determined by our own perception and partially by something *external.* Experiments show that we cannot distinguish well which part belongs to *internal* and which part to *external* objects and processes. They all are connected into a dynamic whole. Consciousness pervades all the objects, processes, or ideas about them, respectively, therefore consciousness *is* that dynamic wholeness.

Consciousness is very heterogeneous. We may be in many different states of consciousness. Our daily consciousness is our individual experience of concrete objects or ideas from a personal point of view. On the other hand, through meditation, we can enter the *transcendental* state of consciousness which is a deep universal experience that everything is connected to or dependent on everything else; everything is implied in or informationally connected to everything else. The observation of this wholeness is valid for the most fundamental level of physical existence—the sub-quantum level (*quantum vacuum*). This was also experimentally verified (EPR-effect). The sub-quantum *sea*, a *mixture* of all possible states, coexists with the classical world of concrete reality.

So we have two worlds:

1. The world of concrete objects located in space and time.

2. The world of a distributed, non-local *fog,* in physical terms, Fourier-transformed, hologram-like worlds—*all in one, one in all.*

The first world consists of localized objects like a human body. The second world consists of entities like human mind. The connection between these worlds is in complex systems like brains. Mind is more like a state/country rather than like an individual man or neuron. A man is localized, but a state/country is all around. The men in a society as well as the neurons (nerve cells) constituting a network in the brain work according to the principle *all for one, one for all.* Neuronal networks are rooted in the most microscopic and fundamental quantum networks. Such interdependent network processes constitute a subtle web of life.

It seems that (sub)quantum wholeness and consciousness are essentially connected. Individual brains are rooted into an overall (sub)quantum *sea* like individual consciousnesses are rooted into a universal Consciousness (God). So, our consciousness and the physical world both have an origin in the fundamental oneness.

Figure 7: John P. Ziebarth, PhD, Director of NASA (National Aeronautics & Space Administration) Consolidated Supercomputing Management Office (CoSMO), Ames Research Center (Moffett Field, CA).

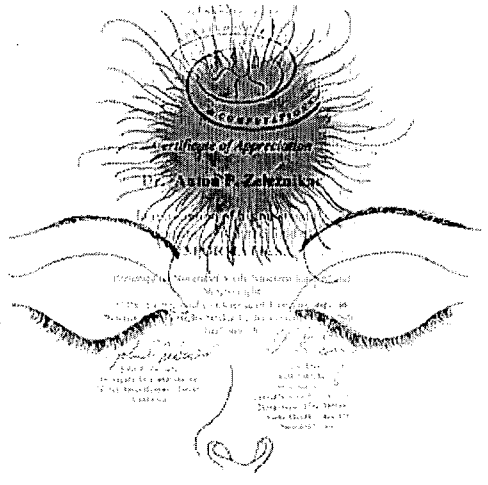


Figure 8: Certificate of Appreciation awarded at the conference banquet to the conference keynote speaker Professor Anton P. Železnikar. The art work was designed by the graphic artist, Mrs. Sujata Dey-Koontz, receiver of artistic awards in Poland, India, Belgium and the United states. The certificate was awarded by Dr. John P. Ziebarth on behalf of the Conference Organizing Committee.

This connection with the fundamental physical wholeness and unity gives great potentials to consciousness. The qualitative capacity of conscious experience may be em enormous—the *transcendental* experience is much more powerful and blissful than the ordinary state of consciousness. The quality of consciousness brings happiness rather than quantity of objects of consciousness. That's why consciousness is so important.

Technological development was enormous and very successful. The people of USA have had especially important role in it. The dream of technological and scientific progress has become a reality. However, material means are needed for *existence* (and these are now satisfied for large parts of the world); for higher *quality* of life, on the other hand, spiritual means are necessary. It is necessary that we direct huge technological and economical growth and growth of consummation with unpredictable environmental consequences over into a growth of spiritual values like mutual understandings and love.

Nowadays, locally things work relatively well (in the West at least), but globally hardly-controlable problems emerge because we are creating complex systems.

Emerging modern techniques like quantum computing or genetic engineering etc. give us enormous and powerful opportunities for development and completely new dimensions, but we also have to face complexity, its' side-effects and its' quite unpredictable impact on the wholeness. A change of *values* using a global and qualitative perspective is needed to ensure an equilibrium of individual and collective interests and freedom.

Higher states of consciousness are realizable. Just if we restore our subtle awareness, feelings and understanding of our independent realities we can be optimistic about our future.

To conclude: Consciousness, *all in one, one in all,* and the creative Wholeness (God) are deeply connected. If experienced, they are the best guide ...

Provoking questions of the future

A serious study of the informational and the conscious within it, the corresponding philosophy, formalism, logo-typographical symbolism and terminology, methodological design, and implementation requirements, can be embraced sufficiently precise merely in the book-form study [6], serving simultaneously as a guide, learning means, reminder, and design motivation. Železnikar hopes that in this respect the work lying before the reader will be of use especially in the understanding of the new approach and in internalization of concepts. These are gathered in the most concentrated form by informational and consciousness axioms and examples in [6].

Artificial consciousness as explored informationally in [6] comes next in the theoretical computer science and theoretical artificial intelligence. Beside the fundamentally new axiomatism and methodology, it puts into the foreground the informational emergentism not being explored systematically in computer science neither in artificial intelligence. Artificial consciousness belongs to the new frontiers of theoretical informatics and as a machine im-

plementation to the new frontiers of computer science—as an organizational structure of the informational machine. Within this context, it has to tackle with informational algorithms, complexity, and models of conscious computation, exploring the new informational logic, meaning, specification, and verification[1].

The most provoking question remains how to implement informational emergentism in the framework of artificial consciousness by machine. The second question is in the traditional doubt that such implementation is probably impossible. Together with these questions the answer to possible implementation must include, besides the formal, methodological and technological means, a transparent concept of the engineering approach—in architecture and programming of the informational machine.

Figure 9: After the trip, returning to Ljubljana, Professor Anton P. Železnikar was confronted with the next demanding challenge, how to write the paper and the book on artificial consciousness promised to the auditorium at the conference [6].

Certainly, many other questions of the conference remained unanswered, for instance those concerning consciousness effects on government, meditation, environment and other existential questions.

[1]A similar, however, still traditional determination of theoretical computer science can be found in *TC1 to hold first conference on theoretical computer science.* 1999. IFIP Newsletter 16:4:2. Železnikar believes that in the next decades the architecture, methodology, and implementation of artificial consciousness will become customary and evident in theoretical and engineering computer science. In that concern, informational principles will become widely used in the old and new sciences, for instance, in the more exact and applicative communication theory, cognitive sciences and, for instance, in systems representing emotional and situated software agents [3], p. 820.

Participant impressions

The conference was organized by Professor Dey and his wife Roma, with the help of the family and the EIU authorities. The question of artificial consciousness was put as a hard problem of concept, method and implementation. The view was that formal means must be elaborated to the level of sufficiently clear technicalities and that for this purpose a comprehensive paper and a book as a guide for artificial consciousness implementation has to be written by Professor Železnikar.

During the paper and the book writing, several new questions emerged requiring to study and elaborate several entirely new concepts on the axiomatic, methodological and formal level. The project in progress will hopefully end during the year 2000. The book is written as a guide for potential implementers, with the most comprehensive informational background and, certainly, with implementation shells for different artificial consciousness concepts. The methodology and formalism presented might essentially change the style of observation and understanding the reality in science, society and cosmological research. Within this perspective, all kinds of information become relevant: linguistic, formal, physical, phenomenal, image, voice, and other possible kinds.

References

[1] Consciousness as informational phenomenalism: An informational, phenomenological, neural and quantum-mechanical view. 1977. *Ed.* A.P. Železnikar. Informatica 21:333–562.

[2] CHALMERS, D.J. 1996. The Conscious Mind. Oxford University Press. New York.

[3] TRAPPL, R. 1999. The Austrian Reserch Institute for Artificial Intelligence: A short presentation. Cybernetics and Systems 30:799–827.

[4] ŽELEZNIKAR, A.P. 1996. Organization of informational metaphysicalism. Cybernetica 39:135–162.

[5] ŽELEZNIKAR, A.P. 1997. Informational theory of consciousness. Informatica 21:345–368.

[6] ŽELEZNIKAR, A.P. 2000. Introduction to Artificial Consciousness. An Informational Approach, Formalization, and Implementation. Project and book in progress.

[7] Consciousness in Science and Philosophy 1998—"Charleston I"—Abstracts. 1998. *Eds.* A.P. Železnikar & M. Peruš. Informatica 22:373–394.

S.K. Dey, Chairman

Call for Paper
International Multi-Conference
Information Society - IS'2000

17 – 19 October, 2000
Slovenian Science Festival
Cankarjev dom, Ljubljana, Slovenia

Invitation

You are kindly invited to participate in the "New Information Society - (IS'2000)" multi-conference to be held in Ljubljana, Slovenia, Europe, from 17–19 October, 2000. The multi-conference will consist of eight carefully selected conferences.

Basic information

The concepts of information society, information era, infosphere and infostress have by now been widely accepted. But, what does it really mean for societies, sciences, technology, education, governments, our lives? What are current and future trends? How should we adopt and change to succeed in the new world?

IS'2000 will serve as a forum for the world-wide and national community to explore further directions, business opportunities, governmental European and American policies. The main objective is the exchange of ideas and developing visions for the future of information society. IS'2000 is a standard scientific conference covering major recent achievements. Besides, it will provide maximum exchange of ideas in discussions, and concrete proposals in final reports of each conference.

The multi-conference will he held in Slovenia, a small European country bordering Italy and Austria. It is a land of thousand natural beauties from the Adriatic sea to high mountains. In addition, its Central European position enables visits to most European countries in a radius of just a few hours drive by car. The social programme will include trips by desire and organised trips to Skocjan or Postojna caves. Coffee breaks, the conference cocktail and dinner will contribute to a nice working atmosphere.

Call for Papers

Deadline for paper submission: 15 July, 2000

Registration fee is 100 US $ for regular participants (10.000 SIT for participants from Slovenia) and 50 US $ for students (3.500 SIT for Slovenian students). The fee covers conference materials and refreshments during coffee-breaks.

More information

For more information visit
`http://ai.ijs.si/is/indexa00.html` or contact `milica.remetic@ijs.si`.

The multi-conference consists of the following conferences:

- Information society and governmental services
- Media in information society
- Education in information society
- Warehouses and data mining
- Development and reingeeniring of information systems
- Production systems and technologies
- Cognitive science
- Language technologies.

International Programme Committee:

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, Korea
Howie Firth, Scotland
Vladimir Fomichov, Russia
Alfred Inselberg, Izrael, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, Japan

22nd International Conference on INFORMATION TECHNOLOGY INTERFACES

ITI 2000

Meeting of Researchers in Computer Science, Information Systems, Operations Research and Statistics

Pula, Croatia, June 13 - 16, 2000

Organised by: SRCE University Computing Centre, University of Zagreb, Croatia
Sponsored by: IMACS International Association for Mathematics and Computers in Simulation
SCS The Society for Computer Simulation International
Under auspices of: Croatian Academy of Sciences and Arts
Ministry of Science and Technology, Republic of Croatia
University of Zagreb.

The aim of the conference is to promote meeting of people involved in the development and application of methods and techniques from the broad framework of information technology, especially those involved in the field of computer science, information systems, operations research and statistics. ITI seeks papers that will advance the state of the art in the field and help foster increased interaction between academic and application communities. ITI welcomes papers, posters, tutorials, panel proposals and workshops in any of the areas of topics suggested bellow.

Keynote Speakers)

Ivan Bratko *(Faculty of Computer and Information Science, Slovenia)*
Atam P. Dhawan *(University of Toledo, USA)*
Matyas Gaspar *(John von Neuman Computer Society, Hungary)*
Ray J. Paul *(Brunel University, UK)*
Marian S. Stachowicz *(University of Minnesota, USA)*
Dalibor F. Vrsalovic *(Internet Platforms, AT & T Labs, USA)*

Deadline for Papers, Posters, Panel and Workshop Proposals: March 5, 2000.

Information: Conference Secretariat
SRCE- University Computing Centre, University of Zagreb
J. Marohnića bb, 10000 Zagreb, CROATIA
Tel.: +385 1 616 55 99 / +385 1 616 55 97
Fax: +385 1 616 55 91
E-mail: iti@srce.hr
URL: http://www.srce.hr/iti

Conference Topics

Computer Systems and Networks
Software Engineering and Programming Languages
Information Systems and Databases
Intelligent Systems
Multimedia and Internet Computing
Data Analysis and Statistics
Biometrics
Modelling, Simulation and Optimisation
Mathematics and Computation
Design, Methodologies and Applications

THE MINISTRY OF SCIENCE AND TECHNOLOGY OF THE REPUBLIC OF SLOVENIA

Address: Trg OF 13, 1000 Ljubljana,
Tel.: +386 61 178 46 00, Fax: +386 61 178 47 19.
http://www.mzt.si, e-mail: info@mzt.si
Minister: Lojze Marinček, Ph.D.

Slovenia realises that that its intellectual potential and all activities connected with its beautiful country are the basis for its future development. Therefore, the country has to give priority to the development of knowledge in all fields. The Slovenian government uses a variety of instruments to encourage scientific research and technological development and to transfer the results of research and development to the economy and other parts of society.

The Ministry of Science and Technology is responsible, in co-operation with other ministries, for most public programmes in the fields of science and technology. Within the Ministry of Science and Technology the following offices also operate:

Slovenian Intellectual Property Office (SIPO) is in charge of industrial property, including the protection of patents, industrial designs, trademarks, copyright and related rights, and the collective administration of authorship. The Office began operating in 1992 - after the Slovenian Law on Industrial Property was passed.

The Standards and Metrology Institute of the Republic of Slovenia (SMIS) By establishing and managing the systems of metrology, standardisation, conformity assessment, and the Slovenian Award for Business Excellence, SMIS ensures the basic quality elements enabling the Slovenian economy to become competitive on the global market, and Slovenian society to achieve international recognition, along with the protection of life, health and the environment.

Office of the Slovenian National Commission for UNESCO is responsible for affairs involving Slovenia's co-operation with UNESCO, the United Nations Educational, Scientific and Cultural Organisation, the implementation of UNESCO's goals in Slovenia, and co-operation with National commissions and bodies in other countries and with non- governmental organisations.

General Approaches – Science Policy

Educating top-quality researchers/experts and increasing their number, increasing the extent of research activity and achieving a balanced coverage of all the basic scientific disciplines necessary for:
- quality undergraduate and postgraduate education,
- the effective transfer and dissemination of knowledge from abroad,
- cultural, social and material development,
- promoting the application of science for national needs,
- promoting the transfer of R&D results into production and to the market,
- achieving stronger integration of research into the networks of international co-operation (resulting in the complete internationalisation of science and partly of higher education),
- broadening and deepening public understanding of science (long-term popularisation of science, particularly among the young).

General Approaches – Technology Policy
- promotion of R&D co-operation among enterprises, as well as between enterprises and the public sector,
- strengthening of the investment capacities of enterprises,
- strengthening of the innovation potential of enterprises,
- creation of an innovation-oriented legal and general societal framework,
- supporting the banking sector in financing innovation-orientated and export-orientated business
- development of bilateral and multilateral strategic alliances,
- establishment of ties between the Slovenian R&D sector and foreign industry,
- accelerated development of professional education and the education of adults,
- protection of industrial and intellectual property.

An increase of total invested assets in R&D to about 2.5% of GDP by the year 2000 is planned (of this, half is to be obtained from public sources, with the remainder to come from the private sector). Regarding the development of technology, Slovenia is one of the most technologically advanced in Central Europe and has a well-developed research infrastructure. This has led to a significant growth in the export of high-tech goods. There is also a continued emphasis on the development of R&D across a wide field which is leading to the foundation and construction of technology parks (high -tech business incubators), technology centres (technology-transfer units within public R&D institutions) and small private enterprise centres for research.

R&D Human Potential

There are about 750 R&D groups in the public and private sector, of which 102 research groups are at 17 government (national) research institutes, 340 research groups are at universities and 58 research groups are at medical institutions. The remaining R&D groups are located in business enterprises (175 R&D groups) or are run by about 55 public and private non-profit research organizatios.

According to the data of the Ministry of Science and Technology there are about 7000 researchers in Slovenia. The majority (43%) are lecturers working at the two universities, 15% of researchers are employed at government (national) research institutes, 22% at other institutions and 20% in research and development departments of business enterprises.

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 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 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 **Slove**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.

In the last year on the site of 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.

At the present time, part of the Institute is being reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host 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, 61000 Ljubljana, Slovenia
Tel.:+386 61 1773 900, Fax.:+386 61 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit three copies of the manuscript with good copies of the figures and photographs to one of the editors from the Editorial Board or to the Contact Person. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible directly on the manuscript, from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks. If the paper is accepted, the editor will also send copies to the Contact Person. The Executive Board will inform the author that the paper has been accepted, in which case it will be published within one year of receipt of e-mails with the text in Informatica LaTeX format and figures in .eps format. The original figures can also be sent on separate sheets. Style and examples of papers can be obtained by e-mail from the Contact Person or from FTP or WWW (see the last page of Informatica).

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. Rudi Murn, Informatica, Institut Jožef Stefan, Jamova 39, 61111 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 five 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://ai.ijs.si/informatica/
http://orca.st.usm.edu/informatica/

Referees:

Witold Abramowicz, David Abramson, Adel Adi, Kenneth Aizawa, Suad Alagić, Mohamad Alam, Dia Ali, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Vladimir Bajič, Grzegorz Bartoszewicz, Catriel Beeri, Daniel Beech, Fevzi Belli, Francesco Bergadano, Istvan Berkeley, Azer Bestavros, Andraž Bežek, Balaji Bharadwaj, Ralph Bisland, Jacek Blazewicz, Laszlo Boeszoermenyi, Damjan Bojadžijev, Jeff Bone, Ivan Bratko, Jerzy Brzezinski, Marian Bubak, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Rajkumar Bvyya, Netiva Caftori, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, David Cliff, Maria Cobb, Travis Craig, Noel Craske, Matthew Crocker, Tadeusz Czachorski, Milan Češka, Honghua Dai, Deborah Dent, Andrej Dobnikar, Sait Dogru, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Jozo Dujmović, Pavol Ďuriš, Johann Eder, Hesham El-Rewini, Warren Fergusson, Pierre Flener, Wojciech Fliegner, Vladimir A. Fomichov, Terrence Forgarty, Hans Fraaije, Hugo de Garis, Eugeniusz Gatnar, James Geller, Michael Georgiopolus, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Inman Harvey, Elke Hochmueller, Jack Hodges, Rod Howell, Tomáš Hruška, Don Huch, Alexey Ippa, Ryszard Jakubowski, Piotr Jedrzejowicz, A. Milton Jenkins, Eric Johnson, Polina Jordanova, Djani Juričič, Sabhash Kak, Li-Shan Kang, Orlando Karam, Roland Kaschek, Jacek Kierzenka, Jan Kniat, Stavros Kokkotos, Kevin Korb, Gilad Koren, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Phil Laplante, Bud Lawson, Ulrike Leopold-Wildburger, Joseph Y-T. Leung, Barry Levine, Xuefeng Li, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Matija Lokar, Jason Lowder, Kim Teng Lua, Andrzej Malachowski, Bernardo Magnini, Peter Marcer, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Roland Mittermeir, Madhav Moganti, Reinhard Moller, Tadeusz Morzy, Daniel Mossé, John Mueller, Hari Narayanan, Rance Necaise, Elzbieta Niedzielska, Marian Niedq'zwiedziński, Jaroslav Nieplocha, Jerzy Nogieć, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczyslaw Owoc, Tadeusz Pankowski, William C. Perkins, Warren Persons, Mitja Peruš, Stephen Pike, Niki Pissinou, Aleksander Pivk, Ullin Place, Gustav Pomberger, James Pomykalski, Gary Preckshot, Dejan Rakovič, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Luc de Raedt, Ewaryst Rafajlowicz, Sita Ramakrishnan, Wolf Rauch, Peter Rechenberg, Felix Redmill, David Robertson, Marko Robnik, Ingrid Russel, A.S.M. Sajeev, Bo Sanden, Vivek Sarin, Iztok Savnik, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Dennis Sewer,Zhongzhi Shi, William Spears, Hartmut Stadtler, Olivero Stock, Janusz Stoklosa, Przemysław Stpiczyński, Andrej Stritar, Maciej Stroinski, Tomasz Szmuc, Zdzislaw Szyjewski, Jure Šilc, Metod Škarja, Jiři Šlechta, Chew Lim Tan, Zahir Tari, Jurij Tasič, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Wieslaw Traczyk, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Elisabeth Valentine, Kanonkluk Vanapipat, Alexander P. Vazhenin, Zygmunt Vetulani, Olivier de Vel, John Weckert, Gerhard Widmer, Stefan Wrobel, Stanislaw Wrycza, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Zonling Zhou, Robert Zorc, Anton P. Železnikar

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.

Volume 24 Number 1 March 2000 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics