Contents

Preface
P. Machanick ............................................................... 1

Research Articles

Active Learning: Issues and Challenges for Information Systems and Technology
RD Quilling, GJ Erwin and O Petkova ............................................................... 5

A Generic Modelling Framework for Interactive Authoring Support Environments
Paula Kotze ............................................................... 15

O Petkova and JD Roode ............................................................... 26

An Information-Theoretic Semantics for Belief Change
T Meyer ............................................................... 33

A Complexity Metrics Model for Software Correction
A Törn, T Andersson and K Enholm ............................................................... 40

A Conceptual Design for High-Volume Data Processing of Warehouse Database into Multidimensional Database
Paisarn Trakulsuk and Vichit Avatchanakorn ............................................................... 49

A Pragmatic Approach to Bitemporal Databases: Conceptualization, Representation and Visualisation
Chiyaba Njovu and WA Gray ............................................................... 58

A Building Recognition System
SP Levitt and B Dwolatzky ............................................................... 68

Computer Programming and Learning to Write
John Barrow ............................................................... 77

Co-operating to Learn using JAD Technologies
TA Thomas ............................................................... 87

Critical Success Factors for the Implementation of DSS at a Selection of Organisations in Kwazulu/Natal
URF Averweg and GJ Erwin ............................................................... 95

Enhancing the Predictability of Two Popular Software Reliability Growth Models
Peter A Keiller and Thomas A Muzzuchi ............................................................... 105
Generalised Unification of Finite Temporal Logic Formulas
  Scott Hazelhurst .......................................................... 110

Harmonizing Global Internet Tax: A Collaborative Extranet Model
  E Lawrence and B Garner .................................................. 119

Improving Object Oriented Analysis by Explicit Change Analysis
  Lui Yu, Siew Chee Kong, Yi Xun and Miao Yuan ....................... 128

Reconciling the Needs of New Information Systems Graduates and Their Employers in Small, Developed Countries
  Rodney Turner and Glenn Lowry ........................................ 136

Shortest Delay Scheduling Algorithm for Lossless Quality Transmission of Stored VBR Video under Limited Bandwidth
  Fei Li, Yan Liu, Jack Yiu-Bun Lee and Ishfaq Ahmad ................ 146

Software Croma Keying in an Immersive Virtual Environment
  Frans van der Berg and Vali Lalioti ................................... 155

Some Automata-Theoretic Properties of $\cap$-NFA
  Lynette van Zijl and Andries PJ van der Walt ....................... 163

Statistical Analysis of an Automated Computer Architecture Learning Environment
  JT Waldron, J Horgan and G Keogh ...................................... 168

The CILT Multi-Agent Learning System
  Hema L Viktor ............................................................. 176

The Development of a Generic Framework for the Implementation of Cheap, Component-Based Virtual Video-Conferencing System
  Soteri Panagou and Shaun Bangay ....................................... 185

The Role of Experience in User Perceptions of Information Technology: An Empirical Examination
  Meliha Handzic and Graham Low ......................................... 194

What are Web Sites Used for: Cost Savings, Revenue Generating or Value Creating?
  Man-Ying Lee .................................................................. 201

New Ideas Papers

Approaches to Video Transmission over GSM Networks
  Bing Du and Anthony Maeder .............................................. 210

From Information Security Baselines to Information Security Profiles
  Rossouw von Solms and Helen van der Haar .......................... 215

Grounded Theory Methodology in IS Research: Glaser versus Strauss
  J Smit .......................................................................... 219

Introducing a Continuum of Abstraction-Led Hierarchical Search Techniques
  Robert Zimmer and Robert Holte ......................................... 223
Multimedia as a Positive Force to Leverage Web Marketing, with Particular Reference to the Commercial Sector

Stan Shear ................................................................. 229

Understanding HCI Methodologies

Peter Warren ............................................................... 234

Electronically Published Papers


Experience Papers

A Java Client/Server System for Accessing Arbitrary CANopen Fieldbus Devices via the Internet

Dieter Bühler, Gerd Nusser, Gerhard Gruhler and Wolfgang Küchlin .............................................. 239

An Object-Oriented Framework for Rapid Client-side Integration of Information Management Systems

Ralf-Dieter Schimkat, Wolfgang Küchlin and Rainer Krautter ........................................................ 244

Distributed Operating Systems: A Study in Applicability

Jürgen Prange and Judith Bishop .............................................................. 249

Formal Verification with Natural Language Specifications: Guidelines, Experiments and Lessons so far

Alexander Holt .......................................................... 253

Introducing Research Methods to Computer Science Honours Students

Vashti Galpin, Scott Hazelhurst, Conrad Mueller and Ian Sanders ........................................... 258

Visualising Eventuality Structure

ST Rock ................................................................. 264

Electronically Published Papers

SAICSIT'99

South African Institute of Computer Scientists and Information Technologists

Annual Research Conference 17-19 November 1999

Prepare for the New Millennium

Is there life after y2k?

Mount Amanzi Lodge, Hartebeespoort, South Africa

Sponsors

http://www.cs.wits.ac.za/

PHDS

research/PHDS.html

Think different.
Preface

Philip Machanick, Overall Chair: SAICSIT'99

Running SAICSIT’99, the annual research conference of the South African Institute for Computer Scientists and Information Technologists, has been quite an experience.

SAICSIT represents Computer Science and Information Systems academics and professionals, mainly those with an interest in research. When I took over as SAICSIT president at the end of 1998, the conference had not previously been run as an international event. I decided that South African academics had enough international contacts to put together an international programme committee, and a South African conference would be of interest to the rest of the world.

I felt that we could make this transition at relatively low cost, given that we could advertise via mailing lists, and encourage electronic submission of papers (to reduce costs of redistributing papers for review).

The first prediction turned out to be correct, and we were able to put together a strong programme committee.

As a result, we had an unprecedented flood of papers: 100 submitted from 21 countries. As papers started to come in, it became apparent that we needed more reviewers. It was then that the value of the combination of old-fashioned networking (people who know people) and new-fashioned networking (the Internet) became apparent. While the Internet made it possible to convert SAICSIT into an international event at relatively low cost, the unexpected number of papers made it essential to find many additional reviewers on short notice. Without the speed of e-mail to track people down and to distribute papers for review, the review process would have taken weeks longer, and it would have been much more difficult to track down as many new reviewers in so little time.

Even so, the number of referees who were willing to help on short notice was a pleasant surprise.

The accepted papers cover an interesting range of subjects, from management-interest Information Systems, to theoretical Computer Science, with subjects including database, Java, temporal logic and implications of e-commerce for tax.

In addition, we were very fortunate in being able to invite the president of the ACM, Barbara Simons as a keynote speaker. Consequently, the programme for SAICSIT’99 should be very interesting to a wide range of participants.

We were only able to find place in the proceedings for 36 papers out of the 100 submitted, of which only 24 are full research papers. While this number of papers is in line with our expectation of how many papers would be accepted in each category, we did not have a hard cut-off on the number of papers, but accepted all papers which were good enough, based on the reviews. Final selection was made by myself as Programme Chair, and Derrick Kourie, as editor of the South African Computer Journal. Additional papers are published via the conference web site.

We believe that we have put together a quality programme, and hope you will agree.

Acknowledgments

I would like to thank the South African Computer Journal production team, Andries Engelbrecht and Herna Viktor, respectively from the Department of Computer Science and Informatics, University of Pretoria, for their work on producing the proceedings.

The reviewers listed overleaf did an excellent job: many wrote very detailed reports, sometimes after being called in on very short notice. Inevitably, there were some glitches resulting from the unexpected workload, but the buck stops with the programme chair: I promise to do better next time.

I would also like to thank my own department for putting up with the extra work and expense that running a conference entails. I tried not to burden them with too much extra work, but our secretaries, Zahn Gowar and Leanne Reddy, inevitably had to take on some extra work. John Ostrowick provided valuable assistance with design of our web pages and call for papers poster. Carol Kernick, who handles our finances and membership records, did a fine job of keeping up with the demands of the conference.

Finally, I would like to thank our sponsors, whose contribution made this conference been possible:

- PricewaterhouseCoopers – sponsored generous prizes and the conference banquet
- National Research Foundation (NRF) – provided financial support
- University of the Witwatersrand – provided financial support
- Programme for Highly Dependable Systems, University of the Witwatersrand – provided financial support
- Standard Bank – provided financial support

SART/SACJ, No 24, 1999
Editorial

- Apple Computer – provided equipment for the conference
- Qualica – provided technical support including helping with the conference web site

Web Site

For more information about SAICSIT, including a pointer to the conference site, see <http://www.saicsit.org.za>.

Referees

- Department of Computer Science, University of Pretoria
  - Derrick Kourie
  - Bruce Watson
  - Vali Lalioti
  - Andries Engelbrecht
  - Ivan Mphahlele
- German National Research Center for Information Technology - GMD
  - Gernot Goebbels
- School of Math Stat and IT, Natal University Pietermaritzburg
  - Peter Warren
- Graduate School of Business, University of Cape Town
  - Kurt April
- CSIR
  - James Jardine
- Department of Computer Science, Rhodes University
  - Shaun Bangay
  - Peter Clayton
  - John Ebden
  - Richard Foss
  - George Wells
  - Peter Wentworth
- Information Technology Division, Rhodes University
  - Caro Watkins
- Department of Information Systems, Rhodes University
  - Brenda Mallinson
- Department of Informatics, University of Pretoria
  - Carina de Villiers
  - Herna Viktor
  - Niek du Plooy
  - Elsje van Rooyen
  - Machdel Matthee
  - Alan Abrahams
  - Jackie Phahlamohlaka
- School of Information Technology & Engineering, University of Ottawa
  - Stan Szpakowicz
  - Dwight Makaroff
- Department of Information Systems, Victoria University
  - Glenn Lowry
  - Peter Shackleton
  - Tas Adam
  - Alastair Wallace
  - Stephen Burgess
  - Dave Burgess
  - Julie Fisher
  - Jerzy Lepa
  - Geoff Sandy
  - Rod Turner
  - Alastair Wallace
  - AndrewWenn
- Peninsula School of Computing and Information Technology, Monash University
  - Ainslie Ellis
- School of Information Management and Systems, Monash University
  - Angela Carbone
- School of Computer Science & Software Engineering, Monash University
  - Dianne Hagan
  - Judy Sheard
- School of Management Information Systems, Deakin University
  - John Lamp
  - David Mackay
  - Philip Joyce
- Marie van der Klooster
- Bill Hewett
- Jennie Carroll
- Rodney Carr

- Computer and Automation Research Institute, Hungarian Academy of Sciences
- Ferenc Vajda

- Department of Computer Science, University of the Witwatersrand
- Ian Sanders
- David Lubinsky
- Conrad Mueller
- Yinong Chen
- Bob Baber
- Vashti Galpin
- András Salamon
- Scott Hazelhurst
- Philip Machanick
- Zoltán Fazekas

- Department of Electrical Engineering, University of the Witwatersrand
- Barry Dwolatzky
- Farzin Aghdasi

- Department of Mathematics, University of the Witwatersrand
- Jonathan Burgess

- School of Law, University of the Witwatersrand
- Victoria Bronstein

- Edward Nathan & Friedland Attorneys, Johannesburg
- Justine White

- Department of Information Systems & Technology, University of Durban-Westville
- Geoff Erwin

- Department of Computer Science, Clemson University, Clemson, SC
- Karen Hay

- Logical SA
- Philip Green

- Department of Computer Science, University of Bristol
- Alan Chalmers

- Department of Systems and Computer Science, Howard University, Washington, D.C
- Todd Shurn
- John Trimble

- Department of Computer Science, University of Cape Town
- Sonia Berman

- King’s College, London
- Peter Wood

- Department of Information Systems, University of Cape Town
- Paul Licker
- Steve Erlank
- Alemayehu Molla

- Department of Department of Mathematics and Applied Mathematics, University of Cape Town
- Ingrid Rewitzky
- Peter Jipsen
- Renkuan Guo

- Department of Information Systems, University of the Western Cape
- Andy Bytheway

- Department of Electrical Engineering and Computer Science, The George Washington University, Washington, DC
- Rachelle Heller

- School of Biophysical Sciences and Electrical Engineering, Swinburne University of Technology
- Ian Macdonald

- Electrical and Electronic Systems Engineering, Queensland University of Technology
- Anthony Maeder

- Computer Science Department, Åbo Akademi
- Aimo Törn
- Kaisa Sere
- Mats Aspnas

- Division of Informatics, University of Edinburgh
- Graeme Ritchie
- Arturo Espinosa-Romero
- Jon Oberlander
Editorial

- Chris Brew
- Alexander Holt

- Department of Computer Science and Engineering, Helsinki University of Technology
  - Tommi Juntila
  - Nisse Husberg

- Technical University of Budapest, Department of Measurement and Information Systems
  - Andras Pataricza

- NASA-Goddard Space Flight Center
  - Nigel Ziyad

- Department of Computer Science & Information Systems, University of Port Elizabeth
  - Janet Wesson
  - Leon Nicholls

- Computer Technology Department, Indiana University Purdue University Indianapolis
  - Tim Price

- Department of Electrical Engineering and Computer Science, The University of Michigan
  - Trevor Mudge

- Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
  - David Forsyth

- School of Computer Science, University of Birmingham
  - Mark Ryan

- Faculty of Mathematics, Computer Science, Physics & Astronomy, University of Amsterdam
  - Carlos Areces

- Department of Computer Science, Universidade do Vale do Rio dos Sinos - UNISINOS Rio Grande do Sul
  - Marcelo Walter

- School of Information Technology & Mathematical Sciences, University of Ballarat
  - Binh Pham
A Pragmatic Approach to Bitemporal Databases: Conceptualisation, Representation and Visualisation

Chiyaba Njovu and W.A.Gray

Department of Computer Science, Cardiff University, Queen's Buildings, Newport Road, PO Box 916, Cardiff CF2 3XF, U.K.
{C.Njovu,W.A.Gray}@cs.cf.ac.uk

Abstract

In this paper, we describe a pragmatic approach to bitemporal data modelling, logical representation and visualisation. Our proposed approach, herein referred to as the Bitemporal Object, State and Event Modelling Approach (BOSEMA), enhances an event-oriented to temporal databases and uses enhanced RUBIS-Schema event concepts for representing bitemporal event types. In this approach, a typical bitemporal object type can be perceived as a composite of its bitemporal event types and bitemporal state types. We contend that time concepts and semantics of time-varying information in bitemporal data models should be captured using valid time and transaction time attributes defined on bitemporal event types. At the logical representation level, our approach relies on the semantic knowledge embedded in temporal relationships between object types for deriving bitemporal historic states at runtime. We show how bitemporal database systems utilising this approach can capture bitemporal event data and derive bitemporal historic state details using built-in knowledge rules. The knowledge rules represent the semantic assumptions associated with the stored bitemporal event data. In order to help users visualize bitemporal data, we describe operators on bitemporal databases that present different bitemporal database views to the user. The different user views can be graphically presented to the user in the form of screen objects. This work is based on our experience in the development of a prototype bitemporal database system called the Bitemporal Object Viewing System (BOVS)). In the BOVS, bitemporal database objects are graphically depicted as collections of active bitemporal screen objects representing bitemporal event and bitemporal historic state objects.

Keywords: bitemporal objects, bitemporal states, bitemporal event, time-varying attributes, bitemporal databases, valid time, transaction time, event-oriented approach, visualisation

Computing Review Categories: H.2.8

1 Introduction

We are all quite familiar with the notion of time, and use time concepts when cooperating and communicating with others. Typically, we associate time with some activities involving objects of interest in the real world or some important events in our lives. We tend to associate time with measurements of processes, places of interest in the world and at times descriptions of some known icons that once existed in the world. However, the precise definition of the time notion largely remains unclear. The difficulties arising from our inability to precisely define the time notion become apparent when we attempt to model real world situations where time concepts and semantics of time-varying information feature prominently.

Traditionally, conventional database systems have been known to provide users with only the most recent and consistent versions of data. These versions of data usually represent a snapshot view of the modelled reality. However, in many application areas (e.g. banking systems, medical records, planning and scheduling systems), there is a need for not only the current versions of data but also past and possibly future versions. The lack of temporal support in such applications raises serious problems. Many database systems that support such application areas have been forced to manage temporal information in an ad hoc manner[29]. However, given the complexity of temporal semantics, such an approach is inadequate. In order to better model such applications, we require temporal data models. Temporal data models capture the semantics of time-varying information and therefore play a key role in temporal data management. In other words, temporal models were designed to capture past, present and possibly future versions of data stored in databases. However, the full semantics of time-varying data have proved to be very difficult to understand and their explicit representation in a model inherently creates further complications[24, 27].

In representing time concepts in database models, a distinction is often made between valid time and transaction time. Valid time denotes the time when the facts are true in the modelled reality[19]. Transaction time denotes the time when the facts are stored in the database and can be retrieved[19]. Temporal databases support one or both time dimensions. A temporal database that represents exactly one system defined valid time and one system defined transaction time is called a bitemporal database[10, 20, 30]. The other two temporal database
models are valid time or historical and transaction time or rollback. Valid time database models support valid time alone. Transaction time database models support transaction time alone [19]. Bitemporal database models can capture the full semantics of both valid time and transaction time.

Temporal database research has been an active area of research since the early 80s. Researchers proposed different time concepts, models of time, temporal database models and modelling notations, algebras and query languages, etc. The underlying premise of this expanding body of research was the recognition that time is not merely another dimension or another data item that can be tagged along with each tuple, but rather a more fundamental organising aspect that users treat in very special ways [9]. Most of the proposed temporal data models either extended the relational data model [30, 25, 17, 28, 23, 3, 30, 8, 13, 11] or the entity-relationship model [15, 14, 16, 21, 31]. These models mainly supported valid or transaction time while a few were bitemporal data models. A survey of the proposed temporal relational models can be found in Tansel et al. [30] and/or Snodgrass et al. [13] while a detailed review of the temporally extended entity-relationship models is given in Gregersen et al. [18]. In fact temporal database research has progressed to a point where a consensus glossary of time concepts [19], a consensus Bitemporal Conceptual Data Model (BCDM) [13] and Temporal Query Language (TSQL2) have been proposed.

In spite of the consensus data model and query language, current research shows that the semantics of the two time dimensions in bitemporal database models are still a subject for further research [7, 10] in that

- researchers argue for different modelling approaches and semantic interpretations without much consensus [7, 10]
- researchers still argue for different representation and presentation models for BCDM relations
- there are still a lot of problems in user discernment of bitemporal database semantics even though bitemporal database semantics required for implementation purposes are well defined [10]

In this paper, we propose a pragmatic approach to bitemporal database modelling, logical representation, processing and data visualisation. Our approach provides for database models in which built-in knowledge rules can be used for deriving bitemporal state details from bitemporal event data. We propose the Bitemporal Object, State and Event Modelling Approach (BOSEMA) as a basis for developing bitemporal databases. Database models developed using this approach shall be called Bitemporal Object, State and Event (BOSE) models. A typical BOSE model can be perceived as a collection of primary bitemporal event data and derived bitemporal state information. Thus, the BOSEMA extends Soukera et al. [25]'s event-oriented approach to temporal databases by defining bitemporal conceptual patterns that characterise bitemporal object types. Like other event-oriented database models, BOSE models consist of two closely related components: the bitemporal event component and the bitemporal state component. This paper outlines how BOSE models can be logically represented and describe how knowledge rules in the form of database triggers can be built into the database for use in the automatic derivation of bitemporal state information. Further, we propose a set of operators that can be used for visualising bitemporal objects. Such an intuitive representation of bitemporal data makes it easy for users to discern the full semantics of bitemporal databases.

The rest of the paper is structured as follows: In the next section, we describe the BOSEMA concepts and show by way of an example application how this approach can be used in the development of BOSE models. In section 3, we describe issues related to bitemporal database representation and derivation assumptions associated with bitemporal historic states. This is followed by a description of our proposed user discernible bitemporal view operators which define the requirements for realising each view in section 4. Finally our concluding remarks are discussed in section 5.

2 Modelling Bitemporality: The BOSEMA

The BOSEMA was born out of our realisation that bitemporal database systems are complex and that complex systems have common patterns. The architecture of a complex system is a function of its components as well as the hierarchical relationships among these components [5]. The BOSEMA defines three bitemporal concepts that describe generic BOSE models. The three concepts are bitemporal object, bitemporal state and bitemporal event. The bitemporal object concept is analogous to the concept of a relation in the RUBIS-Schema [25]. Both concepts represent physical or conceptual entity types in the real world and have an associated bitemporal behavioural history. Their characteristic properties are either Time-Varying Attributes (TVAs), Constant Attributes (CAs) or time-varying relationships. An object or entity type is said to be bitemporal if it possesses at least one time-varying attribute or participates in at least one time-varying relationship.

The bitemporal event concept captures the semantics of the event notion in bitemporal database models. The notion of events has not reached an exhaustive and homogeneous formulation [4, 2]. Oberweis and Lausen [26] consider events as having zero duration at some level of granularity. Mittal [22] views events as an assertion about the occurrence of some data concept. Brunet et al. [6] see events as an object state change triggering one or several
operations. In addition to the Operation concept, the Rubis Time Model\cite{25} defines three event types namely factual events, message events and temporal events.

Factual events describe noticeable state changes in one and only one relation. (e.g. an employee's salary as recorded in the database becomes greater than that of his/her manager). Message events describe some actions in the real world. (e.g. a cheque arrival). But more importantly, the message event predicate describes the acceptance condition of the event. (e.g. the date of the cheque is valid) after which the message event is said to recognise the message object. Temporal events describe a situation with reference to time (e.g. payslips are sent out on the 30th of each month). Operations represent the elementary actions triggered by the event occurrences in the database system. In other words, an operation stands for an action type (e.g. modify account balance) and can modify at most one relation\cite{25}. The relationships between the RUBIS-schema event concepts and the bitemporal event concept can be represented as shown in figure 1. From figure 1, it is clear that only transaction timestamps can be associated with the RUBIS-Schema factual events and operations. Valid timestamps can be associated with the RUBIS-Schema message events. The bitemporal event concept clearly defines the event notion by associating database facts with the semantics of time in the two miniworlds - the miniworld representing the modelled reality and that representing a history of facts in the database.

The bitemporal state concept describes the information structures used for storing a series of timestamped state information. The intention of the bitemporal state concept is to allow the bitemporal object to alter its behaviour when its internal state changes. Associations between bitemporal state types and bitemporal event types can be used for capturing the full semantics of bitemporal objects. Typical relationships between these two types are the before and after associations. Bitemporal event types represent collections of bitemporal event instances while bitemporal state types represent collections of historic state instances of the modelled bitemporal object. In BOSE models, a typical bitemporal object type can be represented by a composite object type consisting of a bitemporal event type and a bitemporal state type. This representation of a generic bitemporal object type is shown in figure 2.

The before and after associations explicitly capture the fact that a given bitemporal event can be associated with two historic bitemporal states namely the object exists in one bitemporal state before the bitemporal event occurrence and exists in another after the event has occurred. Since bitemporal events have both valid and transaction time attributes, the two associations (before and after) capture the semantics of the partial ordering between time types. Full semantics of the partial order functions defined on the time domain are described in\cite{1}.

Motivating Example

Most modern application domains view temporal data as an invaluable asset from which decision support information can be derived in order to understand past, present and possibly future data trends. In medical applications, for example, medical authorities may be interested in knowing full histories of patients' visits to the hospital, times and causes of admissions, frequently used drugs, drug requirements by patient categories and the times of the year when such drugs are often required, etc.

To understand why complete bitemporal histories are

Figure 1: The Bitemporal Event Concept as an Extension of the Temporal Event Concept

Figure 2: A Generic Object-Oriented Type View of A Bitemporal Object Type
important in medical applications, consider scenario 1 given below.

**Scenario 1**

- On day 1 (21/05/94), patient \( X'Y'Z \) visits hospital \( A \) and is attended to by Physician \( AX \) who recommends drug \( V \) for administration commencing on the same date.
- On day 5 (25/05/94), patient \( X'Y'Z \) returns to the hospital complaining about lack of improvement in his condition in spite of his continued administration of drug \( V \). Physician \( AX \) attends to the patient and recommends that the patient stops taking drug \( V \) and instead switches to drug \( Z \) starting from day 6.
- On day 7 (27/05/94), patient \( X'Y'Z \) returns to the hospital with further complaints that his condition was getting worse since the change of drugs on day 6. Physician \( AX \) attends to the patient and wonders why the change of drug was effected. He recommends that the patient reverts to drug \( V \) with immediate effect for a further two days.
- On day 11 (31/05/94), patient \( X'Y'Z \) goes back to the hospital citing some improvements in his condition. Physician \( AX \) attends to him and recommends the administration of drug \( V \) for a further three (3) days.
- Five years later (1/06/99), a letter from notary agents of patient \( X'Y'Z \) addressed to the hospital authorities is received. The letter states the complaint from the patient about drug \( Z \) which he received on one of his visits to the hospital in 1994. The patient complains that the drug had left him with some permanent medical condition which had made his life unbearable and that he was in the process of taking legal action against the hospital authorities for compensation.

From Scenario 1, it is clear that medical authorities may not be in a position to immediately respond to such a complaint without having to go through the past records to find out what really transpired. Depending on the organisation of their filing system, they may never have access to the patients' records dating as far back as the past five years. This may cause a lot of problems which may lead to serious financial liabilities.

**Medical Domain Application Semantics: Our Modelled View of Scenario 1**

Three main distinct time types have been defined for timestamping data. These are *time points*, *durations* and *intervals*. Time points, also known as absolute timestamps, are indivisible instants in time (e.g. Monday the 1st of January, 1990); durations are lengths of time which are independent of any time points (e.g. 3 weeks). Intervals are blocks of time each lying between two specific time points (e.g. from 10.00 A.M. to 10.00 P.M). Bitemporal events are timestamped with absolute time points (e.g. '07/03/65 00:00'). However, different granularities may still be employed and this needs to be specified in the design as part of the semantic assumptions for state derivation. Since events occur instantaneously in the real world and do not persist, bitemporal event types in BOSE models effectively represent memories of event occurrences in the real world. By extending the generic BOSE model shown in figure 2, application specific models can be built by identifying application domain object types that conform to the definition of bitemporal object types. The BOSE model for scenario 1 is shown in figure 3. Application domain object types define properties peculiar to the modelled reality. For example, figure 3 represents some of the object types required in the development of a bitemporal data model for Scenario 1. The figure shows that the entity types *Patient* and *Physician* are bitemporal object types. In the case of the bitemporal *Patient* object type, its associated bitemporal event and bitemporal state types are *Patient.Visit* and *Patient. History* respectively. The semantics of the *Patient. History* type are contained in its associations with the *Treatment. History* and *Reported. Symptoms*. Furthermore, the model captures the fact that *Bitemporal. State* types such as *Patient. History* have associated with them four derived attributes which represent the two interval timestamps for the represented bitemporal state instances.

The model presented in figure 3 though incomplete captures enough bitemporal database semantics for supporting application specific decision making tasks. The question highlighted in Scenario 1 is one such example.

**3 Bitemporal Data Representation, Semantics and Derivation Rules**

**Choice of Domain Types For Bitemporal Event Timestamps**

Timestamps require an appropriate domain definition so that a database system that deals with bitemporal events can present consistent semantics to the user[10, 32]. The fact that bitemporal database semantics have been formalized in [10, 19] means that bitemporal DBMSs, where available, can determine, schedule, optimize and execute transactions involving bitemporal information. In order to achieve the required consistency, there is need to avoid a mismatch of domains.

Our approach reaffirms our belief that factual data should be collected at the source based on the valid time and transaction state timestamps associated with bitemporal events. However, bitemporal queries and snapshots need time intervals to retrieve information since their parameters will rarely match event timestamps. The bitemporal state database component stores bitemporal data timestamped with derived intervals which can be used for querying purposes. Since bitemporal event timestamps may be associated with arbitrarily fine granularity, we
need a way of converting this event data to a well-formed representation which we associate with the bitemporal historic state database data.

Handling Granularity Differences in Bitemporal Event Representations

In any specific application, the granularity of time has some practical significance[32]. For example, the time point associated with a patient’s visit to a hospital is associated with a date and time so that minute would be an appropriate granule for most hospital appointments. Different application domains may employ different granularities in their transactions. Eventually, the limit depends on the precision offered by the computing devices. For example, most computer hardware in use today can only represent time up to fractions of microseconds. The finiteness of measurement granules leads to a confusion of event times and intervals. There is need to remember that in the discrete model of time, a point event is associated with a chronon which is the smallest indivisible interval representation of the granule[19]. If we denote the chronon by (G = one day, for example), and we say that an event occurred on a particular day, then implicit for most of us is the fact that the event spanned some interval within that day. This simplified assumption may lead to inconsistencies in interval computations. We shall show in the next subsection that the assumptions associated with the semantics of the derived bitemporal historic state data may be used to resolve this problem.
Derivation Rules and Associated Assumptions For Bitemporal Historic State Data

<table>
<thead>
<tr>
<th>Type</th>
<th>Transformation Data</th>
<th>Interval State</th>
<th>Example Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable</td>
<td>Use start value</td>
<td>Salary</td>
<td>Salary-Charge</td>
</tr>
<tr>
<td>Stable</td>
<td>Use difference point</td>
<td>Growth</td>
<td>Interest-Rate-Charge</td>
</tr>
<tr>
<td>Average</td>
<td>Use average point</td>
<td>Productivity</td>
<td>Output-Measures</td>
</tr>
<tr>
<td>Minimum</td>
<td>Use minimum point</td>
<td>Illumination</td>
<td>Light</td>
</tr>
<tr>
<td>Maximum</td>
<td>Use maximum point</td>
<td>Power-Rating</td>
<td>Power-Uage</td>
</tr>
<tr>
<td>None</td>
<td>Not Continuing</td>
<td>Not Applicable</td>
<td>Bonus</td>
</tr>
</tbody>
</table>

Figure 4: Bitemporal Historical State Attributes' Semantics

Two main issues arise when dealing with the semantics associated with the bitemporal state component. The first issue concerns the assumptions attached to the bitemporal historic state attribute semantics. Unless we formalize all relevant bitemporal data semantics, we cannot leave the task of executing queries to the system. The responsibility of defining the semantics for the bitemporal data falls onto the users or the database designers. Users may need to specify these semantics at query definition time. Database designers can provide these semantics as part of the derivation knowledge built into the database. What then are these assumptions? A critical step in processing bitemporal data is the conversion of bitemporal event data to bitemporal state histories. The derivation of values of time-variant attributes at intermediate points within the bitemporal historic state intervals, defined by event times, require an understanding of what actually occurs within the interval[32]. Figure 4 shows some of the possibilities.

Though desirable in many respects, bitemporal event representations have the problem that they cannot directly provide answers to queries dealing with event points that are not explicitly represented. For example, citing events in Scenario 1, a query to determine the period when patient X’s Y’s Z was taking drug Z may not yield an answer from the events data alone. For this, we need to use the derivation rules before an answer can be obtained. The derivation rules implicitly endorse the assumption that once a drug is prescribed, starting from the valid timestamp, a patient takes that drug until there is a change after another Patient Visit or the course comes to an end. In other words, the stability assumption is made implicitly. Our design approach assumes the stability rule in time-varying attribute values as a basis for state derivation. However, the stability assumption may not hold for all attributes. For example, if the database also records patient’s body temperatures, or bonuses paid to Physicians, then the snapshot query should not assume that the patient’s body temperature or the bonus payment holds throughout every implicit granularity unit used. Consequently, assumptions other than stability are considered. Some of these are average rule, minimum or maximum rules, rate and in some cases, no assumptions at all. Figure 4 summarises these assumptions.

The second issue concerns the semantics of the intervals for computation purposes. Inconsistencies may occur when an inclusive interval is defined by two bitemporal event time measurements with an implicit granule. The way forward lies in our ability to round off actual measurements to the next integral grain size used before we add the granule size to the result. This can formally be expressed as:

\[ T_G = t_f - t_i + G \]

where \( t_i \) denotes the value corresponding to the start of the interval and \( t_f \) the value when the interval is finished. For example, if a patient’s treatment starts on 23rd to the 29th of a month, he/she would need to be supplied with drugs for a period given by \( T_G = 29 - 23 + 1 = 7 \) treatment days.

While we are all quite familiar with such adjustments when computing time intervals, a database system that deals with bitemporal event data must present consistent semantics to users in order to avoid confusion.

The foregoing assumptions are essential for the definition of operators that allow for the derivation of bitemporal historic state data from bitemporal events.

4 Required Steps For Processing Bitemporal Data

Given our proposed design approach, the steps required for processing bitemporal database data have to be explicitly outlined. There are three main processing steps:

New Bitemporal Event Data Entry Scheme

For all object types represented in our model, the bitemporal event database component is the means for object creation and new data collection. Each bitemporal object instance is created by means of a special bitemporal event whose role is to instantiate new bitemporal object instances. For example, a FirstPatientVisit event type can be used for creating new patient instances in the medical records database or a LoanApplication event type can be used for creating new loan records in the loans information system. For each TVA or relationship, we define a bitemporal event type whose instances are used for collecting new event attribute values. Bitemporal event instances are all absolutely timestamped. However, application designers are free to choose the appropriate level of granularity. For example, a (year/month/day/hour:min) timestamp would serve medical applications well as most hospital appointments are scheduled to the minute level of granularity. Each bitemporal event transaction time timestamp is a read-only date and time value which is automatically gen-
Bitemporal Historic State Derivation

This is probably the most important processing step. In this step, we use our built-in knowledge in the form of derivation rules and assumptions to compute bitemporal state information from the stored bitemporal event data. The knowledge required for bitemporal state derivation can be built into the database system either in the form of triggers or knowledge rules which can be interpreted by the DBMS to calculate the consequences of a bitemporal event on the current state of a bitemporal object. There are two possible cases that may be considered in deriving bitemporal historic states: The first concerns those bitemporal object properties whose historic state values depend on previous state values in addition to the new bitemporal event data. The second concerns those properties whose historic state values can be derived directly from bitemporal event data only. For the first case, there is need to cache the initial bitemporal state values required for state derivation. Using this information, the system can compute all subsequent states dynamically as bitemporal event data is accessed. To effect this, our derivation operators should identify each case dynamically in order to ascertain the algorithms to be used for state derivation.

Basically two phases are required in the derivation process and these are:

- Creation of bitemporal state interval timestamps for each bitemporal instance to be derived
- Evaluate the built-in knowledge rules according to the assumptions determined for the required state information

Using the Patient.Visit bitemporal events described in scenario 1, the following algorithm can be applied for deriving the treatment history of patient XYZ. These steps are accomplished by the following algorithm.

**Derivation Algorithm**

1. Repeat steps 2 - 6 until all bitemporal objects in the collection have been considered
2. Get the first Patient.Visit event and extract the two timestamps which form the lower bounds of the bitemporal state interval timestamps
3. Determine the upper bounds for the bitemporal state interval timestamps. If validity period information is stored, then upper bounds for the interval can be computed by adding the duration to the valid timestamp otherwise, get next Patient.Visit bitemporal event instance
4. Get next Patient.Visit bitemporal event and extract the timestamps that form the upper bounds of the bitemporal state interval timestamp
5. If the next Patient.Visit event is not available, then the upper bounds for the interval are designated as Until Changed (uc), which means the current date becomes the upper bound for the interval until it is changed
6. Evaluate bitemporal state information using the stability rule. For example, the rule for patient treatments stipulate that a patient continues taking that drug until it is changed or the course comes to a logical end
7. Repeat steps 2 - 5 for the next state

Applying the foregoing algorithm to the bitemporal events described in Scenario 1 yields the information shown in figure 5. In fact, the main difference between our approach and that employed by conventional (state-oriented) databases lies in the ability to build in database knowledge that can be used by the system to convert bitemporal event data into historic state information. Our approach, by way of this step, overcomes the restrictions cited in conventional databases in that it ensures that the interpretation and calculation of the consequences of bitemporal event occurrences on the current bitemporal state is done within the database. In conventional (state-oriented) databases, the interpretation of event consequences on database states is usually done outside the database with the user having to input new state values each time a database state change occurs.
Visualising Bitemporal Database Query Results

Given the difficulties associated with user discernment of bitemporal database semantics, there is need for operators to allow users to declaratively specify their data requirements. We propose a set of operators that extract specified temporal views from the bitemporal database. These views mainly describe the different histories we can define on bitemporal objects in the database. Jensen et al[19] defines a history as the temporal representation of an object of the real world or of a database. All bitemporal objects have histories in that they have both their valid time history and their transaction time history. We can discern a total of nine temporal views from each bitemporal object. These are summarised in figures 6. The operators required for realising each view are defined below.

Bitemporal View Operators

In figure 6, three temporal operators can be defined on a bitemporal object. The bitemporal.history.operator deals with the bitemporal history of the object. It takes as its arguments, four timestamps, two representing the valid time interval and two representing the transaction time interval. The result of applying this operator is a collection of bitemporal historic states defined over the specified interval. The valid.time.history.operator and transaction.time.history.operator are equivalent to the valid and transaction timeslice operators[12]. Each of these operators takes two arguments representing their valid time and transaction time intervals respectively and return a collection of respective valid time and transaction histories. Using similar arguments, we can define operators for viewing the attribute histories, relationship histories and event histories. Figure 6 shows a complete set of views we can define on bitemporal objects stored in a bitemporal database using similar operators and semantics.

Bitemporal Object Operators

These operators are used for managing components of bitemporal objects. They are defined on the bitemporal object type in figures 2 and 3. The first operator, the create takes as an argument a bitemporal event instance and creates a bitemporal object whose first event instance is the given bitemporal event. The other three operators are used for managing collections on the bitemporal object timeline. The next and previous operators take as an argument a bitemporal event instance and return a bitemporal event instance which is either the next or the previous on the timeline respectively. The fourth operator insert inserts a bitemporal event instance at a specified position on the timeline. It takes a bitemporal event which is inserted at a position given by its transaction timestamp on the timeline. These operators can be extended for navigating through bitemporal historic state collections by simply changing the arguments passed to the operators. Using these operators, users can discern different views of the bitemporal database.

5 Conclusion

The BOSEMA can be used for developing bitemporal data models which allow for the storage of bitemporal event data and some rules (knowledge) on how bitemporal historic state data can be derived from the stored event data. The significance of this approach lies in its ability to capture data about real world events in a natural way. Data in the real world originates from event occurrences which generate the details required for changing states. However, conventional (state-oriented) databases and other non-event-oriented temporal databases only store state details and do not explicitly store any events data. Usually, the interpretation of event consequences on database states is done outside the database. In conventional database, no previous state details or events data are explicitly stored making it impossible to reconstruct previous states when errors are detected. The lack of such historic data and knowledge rules in state-oriented databases implies that the database itself cannot include knowledge of how a bitemporal event affects an associated bitemporal state.

Our approach overcomes the restrictions cited in conventional databases and other non-event-oriented databases by ensuring that the interpretation and calculation of the consequences of bitemporal event occurrences on the current bitemporal state is done within the database using built-in knowledge rules. The built-in knowledge stipulates how bitemporal event occurrences should affect their associated bitemporal states. It is our view that deriving historic states from events data raises the semantic level of the database making it more flexible to manipulate. By maintaining bitemporal data in database management systems, data can be shared by different user applications, each accessing a view of the data required for its functionality.
Furthermore, using our approach, it is easy to expand the view that the bitemporal database provides to the user. This can easily be done by defining additional bitemporal event types to capture more time-varying attributes and relationships of interest to the application. Our approach also allows for implementations that use either tuple or attribute timestamping in their presentation schemes.

References


Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal’s readers will be considered for publication under sections marked as Communications of Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

Form of Manuscript
Manuscripts for review should be prepared according to the following guidelines:

- Use wide margins and 1½ or double spacing.
- The first page should include:
  - the title (as brief as possible)
  - the author's initials and surname
  - the author's affiliation and address
  - an abstract of less than 200 words
  - an appropriate keyword list
  - a list of relevant Computing Review Categories
  - Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text according to the Harvard method.

Manuscripts accepted for publication should comply with guidelines as set out on the SACJ web page, http://www.cs.up.ac.za/sacj which gives a number of examples.

SACJ is produced using the \LaTeX\ document preparation system, in particular \LaTeX\ 2e. Previous versions were produced using a style file for a much older version of \LaTeX, which is no longer supported.

Please see the web site for further information on how to produce manuscripts which have been accepted for publication.

Authors of accepted publications will be required to sign a copyright transfer form.

Charges
Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30.00 per final page for contributions which require no further attention. The maximum is R120.00, prices inclusive of VAT.

These charges may be waived upon request of the author and the discretion of the editor.

Proofs
Proofs of accepted papers may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Letters and Communications
Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words. Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book Reviews
Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement
Placement of advertisements at R1000.00 per full page per issue and R500.00 per half page per issue will be considered. These charges exclude specialised production costs, which will be borne by the advertiser. Enquiries should be directed to the editor.