South African Computer Journal

Suid-Afrikaanse Rekenaar-Tydskrif

Number 24
November 1999

Nommer 24
November 1999

Special Issue: SAICSIT '99

The South African Computer Journal
An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarvereniging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

World-Wide Web: http://www.cs.up.ac.za/sacj/

Subscriptions

	Annual	Single copy
Southern Africa	R80.00	R40.00
Elsewhere	US$40.00	US$20.00

An additional US$15 per year is charged for airmail outside Southern Afr

+27 (11) 315-1319 Fax: +27 (11) 315-2276

South African Computer Journal

Number 24, November 1999
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 24, November 1999
ISSN 1015-7999

Contents

New Ideas Papers

Experience 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

PRICEWATERHOUSECOOPERS

Standard Bank

NRF

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 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

- 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
  - Dave Sewry
- 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 Deal
  - Julie Fisher
  - Jerzy Lepa
  - Geoff Sandy
  - Rod Turner
  - Alastair Wallace
  - Andrew Wenn
- 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

  - Chris Brew
  - Alexander Holt

- Department of Computer Science and Engineering, Helsinki University of Technology
  - Tommi Junttila
  - 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

Visualising Eventuality Structure

S. T. Rock*

Department of Computing and Electrical Engineering, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, Scotland,
sheila@cee.hw.ac.uk

Abstract

This paper describes the use of pictorial representations in showing the understanding of natural language that is about multiple or extended events. The focus is on the structure of such eventualities, and depiction of the patterns of the events within a structured eventuality. The visualisation exploits the complementarity relationship between objects and events, often noted in the literature, to map from an essentially temporal entity domain—eventualities—to a spatial one—pictorial representation of their execution structure.

Keywords: *visualisation, natural language, eventuality structure*
Computing Review Categories: *D.1, I.2.0, I.2.7*

1 Introduction

The focus of this paper is the pictorial representation of the meaning of natural language sentences that describe multiple, repeated or extended execution of events, such as those in (1).

(1) a. At noon a bell rang in every room of the hotel.
b. Vusi knocked on the door three times.
c. An exhausted Thandi slept for ten hours.

For these examples, of interest are the patterns of multiple, repeated or extended execution of events, rather than the detail of the event itself; it is the visualisation of these that I am concerned with. In (1a) it is the multiple execution—in every room of the hotel[1]—of some event; in (1b) it is the repeated execution—three times—of some event; and in (1c) it is the extended execution—for ten hours—that will be represented.

1.1 The scope of exposition

The broad area of interest here is visualising EVENTUALITY[2][3] structure. Previous work has looked at the ontological analysis of this kind of eventuality phenomenon[12, 13] and at the semantic analysis of language that is about these kinds of eventualities[13, 14]; neither of these is discussed here, other than where necessary to set the context for the visualisation that is the focus of this paper.

I adopt the view that an eventuality may be composed of sub-eventualities and for simplicity I take sentence boundaries to indicate eventuality boundaries. In (1b) then, the whole eventuality is the one of ≪*Vusi knocking on the door three times*≫[4], and its sub-eventualities—of which there are three—are simple eventualities of ≪*Vusi knocking on the door*≫. We are thus looking at the internal eventuality structure and how sub-eventualities are related to each other, and issues of mereology and of topology are both of concern.

Although temporal issues are of interest, this is focused on the temporal relation between sub-eventualities; no treatment of linguistic tense is included. I have also indicated that the detail of the activity involved in an eventuality is not of interest. In (1b), no analysis is done of what ≪*Vusi knocking on the door*≫ means; it is viewed just as something that is an eventuality. I have chosen, therefore, to simplify matters by using instructions as my primary source of examples: they are usually tenseless, and the subject is usually implicitly the hearer of the instruction. Finally, the treatment is limited to the visualisation of the temporal relations between eventualities that have some aspect of repeated, multiple or extended execution associated with them. Issues such as eventualities overlapping, meeting, etc., as in the manner described by Allen [2] for example, if they are not multiple instances of activity, are not addressed here.

A simple approach is taken for eventualities that consist of two or more sub-eventualities that are not repeated, extended or multiple instances of some event, such as in (2). In this exposition the two sub-eventualities ≪*Thandi*

[1] The spatial location of these events is not important here; what is important is that there are many of them—as many as there are rooms in the hotel.

[2] I adopt the term eventuality (as used by Bach [3] and others) to refer to any occurrence in the world; a possible 'happening' or 'state'. The term SITUATION is also used in the literature to describe these kinds of entities.

[3] A note about terminology: There has been much conflation and overriding of terms in the literature about the ontology of world entities. My approach here is to use the terminology used by the authors when discussing previous work, and to clarify my own use of terminology when I describe my own ideas. This is either in the form of relating it to established usage, or defining my own when there is no clear term already in use.

[4] This notation is used to indicate a semantic gloss.

going to the party≫ and ≪*Thandi having a good time*≫ are seen as distinct and for convenience just follow each other. While it is acknowledged that this analysis is a simplistic one, and there could in fact be a host of relations between the two sub-eventualities, these are not within the focus of the study and the simple approach is adopted for now.

(2) Thandi went to the party and had a good time.

2 Objects and eventualities

2.1 Ontology and analogies

There have been various taxonomies suggested regarding what can be described by language. A common view says that we talk about entities, and divides the entity space into objects and eventualities—objects are things that exist and eventualities are things that occur. There have been proposals that say that objects are related to space and eventualities are related to time in analogous ways[16, 9, 4, 6, 3].

The first analogy that is of interest is the one proposed and formalised to some degree by Mayo[9]—that objects and events[5] are the ontological complements of each other with respect to time and space. For example, objects have limited extension and unlimited duration, while events have limited duration and unlimited extension.

Zemach[16] takes this analogy further and proposes that we conceive of things from four distinct ontologies; the ontologies are distinguished by their boundedness or continuity in time and in space). Objects (which he calls 'things') and events (which he calls 'processes') are entities from two of his ontologies; the other two are of entities that are bounded in both time and space, and entities that are continuants in both time and space. Objects are thus bounded in space and continuous in time, while events are bounded in time and continuous in space. So, with regard to the spatial component, an object has spatial parts. It can be divided in parts and these parts are different parts of the object. With respect to time, however, an object is a continuant. At different times, the object is *not* a different object. A table has parts that are different parts of the table (legs, drawers): spatially the table is bounded. But the table I see today is the same entity as the table I saw yesterday: with respect to time, the table is continuous. And two table-entities are one and the same thing only if there is temporal distance between their respective temporal locations.

A similar argument applies to event entities—'this noise', 'the Roosevelt era'—they are bounded (defined) in time but continuous in space. As an event 'World War II' can be at the same time in many places—continuous in space—but cannot be in the same place at many times—bounded in time. This is not to say that 'World War II' occurs only for an instant; it can occur for a large period of time, but any occurrence of 'World War II' that is outside of this period (say the period 1939 to 1945) is a different instance of 'World War II'.

The second analogy [6, 15] relies on the object-event complementarity thesis summarised above. Put very simply, it analyses continuous events as mass and repeated events as plural. Jackendoff [6, pg 16] says that

> 'the semantic value of repetition is identical to that of the plural, that is, it encodes the multiplicity of a number of tokens belonging to the same category.'

Although Jackendoff's view is a little narrow (it does not include the possibility of the eventuality described in (1a) being the event analogue of plural), it essentially encompasses the correspondence required—that multiple events can be viewed as analogous to multiple objects. There are also similarities between events that occur continuously and mass objects—many of the tests that we might apply to some object to determine whether it is mass also apply to determining whether an event is plural.

2.2 Structure and sub-structure

Central to this study is the view that ACTIVITY[6] is always itself made of activity, and it is possible to *view* the activity from a number of different levels. This is a position analogous to the one of seeing objects as themselves consisting of objects; it is only the fact that we view an object from a particular perspective that allows us to distinguish it from other objects and allows us to delimit its boundaries. For example, a dining suite may be seen as a dining suite, or as a table and six chairs, or as wood and nails and glue, or as hydrocarbon molecules. However it is seen though, it is still the same object. It is less usual, though certainly possible, to see the dining suite object as two chairs and some carbon and other molecules. An object is itself composed of things that can themselves be conceptualised as objects.

Applying the same analysis to an activity—for example that of dancing—performing a dance can be seen as executing a series of dance steps, or as doing a stream of body movements. It can also be seen as the response to a series of nerve impulses. Performing a series of 16 pirouettes can be seen as a composite activity that consists of 16 sub-activities, each of which is the activity of performing one pirouette.

With regard to language visualisation, then, I am concerned only with that degree of structure that can be determined from the language. I argue that it is the language that makes us conceive of the structure in a particular way. Contrast these language fragments:

(3) a. Albertina bought a dining suite.

b. Albertina bought eight chairs and a table.

[5]The term event here is being used to describe something that happens, an eventuality that is not 'a state of being'.

[6]For now I use the term activity to refer to the substance of an eventuality, or to what exists when an instruction gets executed. This is a simplification: I give a fuller description of the distinction between activities and eventualities elsewhere[13].

(4) a. Do the pirouette sequence.

b. Do that pirouette 16 times.

In (3a) and (4a), there is no suggestion that there is more than one object or eventuality, and each sentence could be represented by an unstructured object or eventuality entity. However, in (3b), we clearly have access to nine objects, and in (4b) there is an eventuality whose execution would involve the execution of 16 sub-eventualities.

2.3 Visualising activity structure

Previous work on visualisation has focused on visualising objects that have been described in text, or on simple activities[10, 8]. While it is popular to visualise activity through the use of animation, it is clear that animation does not really illuminate activity *structure*. There is also the issue of time in animation; it is not straightforward to visualise the passage of time in any kind of comparative way. Finally, it is not easy to express quantitative aspects without, say, having the animation take as much time to run as the activity which it is intended to visualise; this is not always appropriate.

There has been some work done around the cognitive aspect of how we understand images[11]. Okada sees events[7] as being those things represented by verbs. An event, anything represented by a verb, is a change from one state to another. For example, the event '*sing*' is seen as having the feature of a change in voice and sound; '*turn*' has a change in direction. While I note that Okada's view of events is an over-simplistic one, to turn the model from understanding to presenting a visualisation would require a means of visualising the change. In the model there is no distinction made between, say, '*sing*', and '*sing for half an hour*' or '*sing one song*'; there is no obvious way to model these differences by simple state changes.

My approach exploits the object-event analogy; activity structure is represented in the form of object structure, where each activity is represented as an object and the sub-activities are represented as sub-objects. Representing time is done by choosing a spatial direction to show the passage of time. The final result is effectively a time graph; because I am not interested in representing exact timing, but rather concepts that are relevant to temporal sub-structure, the graphs are also not exact spatially but they represent—in spatial terms—the important concepts.

So, it is possible to represent the structure of (4b) by drawing an unmarked time graph that contains 16 entities, as in Figure 1. That is, the structure of the entire eventuality is represented by the outer block; it contains—within it and arranged consecutively along the time-axis—16 smaller blocks. Each of these blocks represents one instance of ≪*do that pirouette*≫ and all of the blocks have boundaries to indicate that their extent is defined. So, using objects—the blocks—and arranging them spatially

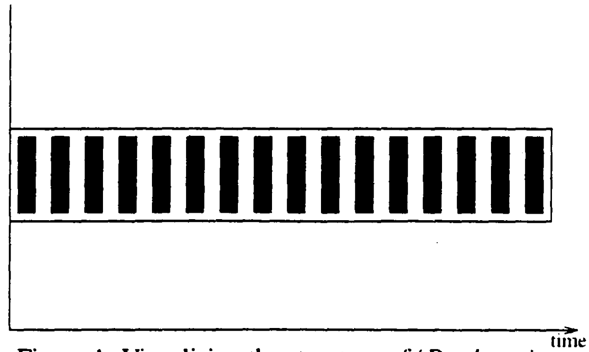


Figure 1: Visualising the structure of '*Do that pirouette 16 times.*'

with respect to each other, it is possible to represent events and their temporal relationships.

3 Components

In this section, I show how the distinctions in the previous sections can be used in the formation of an approach to visualising eventuality structure. This approach is a subset of a more comprehensive approach[8] to visualising repetition, multiple occurrence and extended execution of eventualities.

3.1 Basic building blocks in the ontology of eventualities

For the purposes of this paper, two basic kinds of component have been identified; representing eventuality structure then involves combining instances or multiple instances of these basic components in different ways. The basic components are called **process** and **happening**. Process is used as analogous to mass in objects and a happening is used as analogous to a count object.

Because of the recursive nature of my analysis, it is possible that a happening may at a lower level itself consist of a structured arrangement of processes and happenings, much in the same way that a dining suite consists of 8 chairs and a table. A bottom level happening or process is one that is indicated via the language.

3.2 Basic kinds of eventuality structure

In keeping with the analogies described earlier, I use two basic structures, which I call **mass** and **set**. These are analogous to mass objects and plural count objects; each uses a different basic representation component. Mass eventuality structure uses the process, and set eventuality structure uses the happening.

1. Mass is an eventuality that takes place continuously over a time period. As is the case with objects, some

[7]According to Okada[11] there are four types of real world objects, namely substance, event, attribute and others.

[8]For a complete treatment of eventualities, all four of Bach's[3] eventuality categories are needed, as well as a more sophisticated analysis of the object-event analogy and of the concept of mass [16] and plural. However, what is presented here is sufficient to illustrate the important concepts.

delimiter is useful if we are to conceptualise the entity more completely. Compare '*water*' and '*a glass of water*' with '*heat the butter*' and ' *heat the butter until it is foaming*'.

2. Sets may be ordinary sets or ordered sets. These are essentially collections of happenings, or multiple happenings.

   - An ordered set has the essential property of being a set of happenings that must occur consecutively.

     (5) Baste the roast twice.

   - An ordinary set is a set of happenings where no temporal order is specified; any timing may be chosen.

     (6) Cut two circles of pastry

A set may have happenings which are executed concurrently (such as in (1a)), consecutively, or randomly. This also allows the possibility of temporal relations between sub-events being dependent on context[5]. So, in (6), the two cuttings could be simultaneous if the cook has available a double pastry cutter, while they would be consecutive if not. Here, unlike in (5), the language is not stipulating consecutiveness.

Two other kinds of structure are included:

1. Concurrent activities consist of two or more (each of these may be itself structured) that occur at the same time.

   (7) Boil the soup, stirring continuously, for 10 minutes.

   There are two activities (≪*boil the soup for 10 minutes*≫ and ≪*stir the soup continuously*≫) that are concurrent. The former activity is a mass, and the latter is an ordinary set.

2. Conjunctive[9] activities are those that contain one activity (again this may be structured) whose extent is expressed in more than one way.

   (8) Simmer the soup for five minutes or until it is thick.

3.3 Arranging the components

The pictorial representation requires arranging the basic components on a time line, to reflect the understanding of the repetition. Descriptions of a repetition activity have information that contains more detail about the nature of the repetition. For example, we may know that a happening occurs every 10 minutes, or that a process occurs for an hour. Individual components and time-lines are arranged in a particular way with annotation, to include the depiction of this additional information.

So, *sleep for three hours* would be a mass event *sleep*, with a size of three hours. *Roll the dough out three times* is an ordered set of happenings *roll the dough out* with a cardinality of three. Finally, *cut three circles of dough* is a set of happenings *cut a circle of dough* with a cardinality of three.

[9]The term conjunctive has been chosen for these sorts of entities; while the syntactic expression of their extent may seem like disjunction, the semantics of the extent is conjunctive in that both options will be true when the overall activity is complete.

4 A computational system

To show the feasibility of the approach described here, I have developed a program that produces graphs that depict the eventuality structure.

Happenings are drawn as shaded rectangular blocks, which are arranged on a graph to indicate how they occur with regard to time. The passage of time is depicted as from left to right across the page. Different happenings have different shading; a set of happenings, forming an activity, has a number of blocks all with the same shading but arranged in a way on the time graph to show their repetition. Processes are drawn as shading between horizontal lines; the temporal delimitation of these events is omitted. However, once a delimiter is present (such as might come from '*for three hours*'), the process is represented as a block, similar to a happening. This is a reasonable approach; once '*water*' becomes '*a glass of water*', it has become a count object.

The drawing is done by a PostScript [1] program—input to the PostScript is a single data structure, the result of the semantic analysis for each discourse[10].

A semantic analysis of (9) determines that the activity consists of two concurrent activities—simmering the soup and stirring the soup. Further, stirring is a happening; that it occurs frequently gives the information that it is part of an activity that is an ordered set. The simmering the soup activity is a mass that has a duration of one hour.

(9) Simmer the soup for one hour, stirring frequently[11].

My semantic analysis produces this data structure:

```
[[60  [concurrent [1 2]]]
 [60  [mass [(simmer the soup) ]
          [(one hour) ]]
 [60  [orderedset [12 3]]]
 [ 2  [happening [(stir the soup) ]]]  ]]
```

This is a PostScript array, containing four sub-arrays, one for each activity. Thus, numbering the arrays 0–3, the activities each represents are:

0 ≪*simmering the soup for one hour, stirring frequently*≫

[10]A discourse is one or more sentences; each sentence is one instruction.

[11]There are different semantic readings that will be obtained from the seemingly similar sentence '*Simmer the soup, stirring frequently for one hour*'; the detail and semantic analysis of these is discussed elsewhere[14]. These would then be visualised differently—for example, the mass of simmering would not have a delimiter

1 ≪*simmering the soup for one hour*≫

2 ≪*stirring the soup frequently*≫

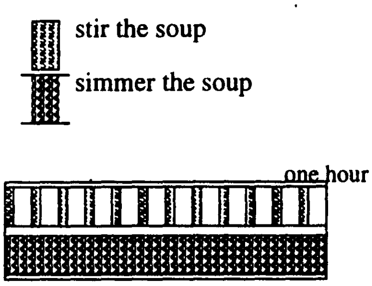
3 ≪*stirring the soup*≫

To enable the actual drawing of the blocks, each activity is given a size; conveniently this could be related to the time that such an activity might take. In the array then, the number at the left is the size of the activity; we could imagine it represents in minutes the time taken for the activity to occur.

For happenings or processes—the basic components identified earlier—text describing the activity follows. This text is used in the creation of the key, as in elements 1 and 3 in the above array.

For the activity that is structured the places where the activity descriptions will be found (the element numbers) follow. For element 0, the concurrent elements of this activity are found in elements 1 and 2. For sets, the cardinality and the place where the description of the set elements will be found follows; thus in element 2, the ordered set is made of twelve instances of element 3.

This is input into the PostScript program, which produces this pictorial representation[12]. Each activity type (such as `mass`, `orderedset`) corresponds to a call to a PostScript routine. Thus, additional activity types can be added in a modular fashion, by adding new routines. More importantly, changes to the way an activity type is drawn can also be made in a modular fashion, if we wish for instance to depict a more comprehensive analysis of intra-event relations as mentioned in section 1.1.

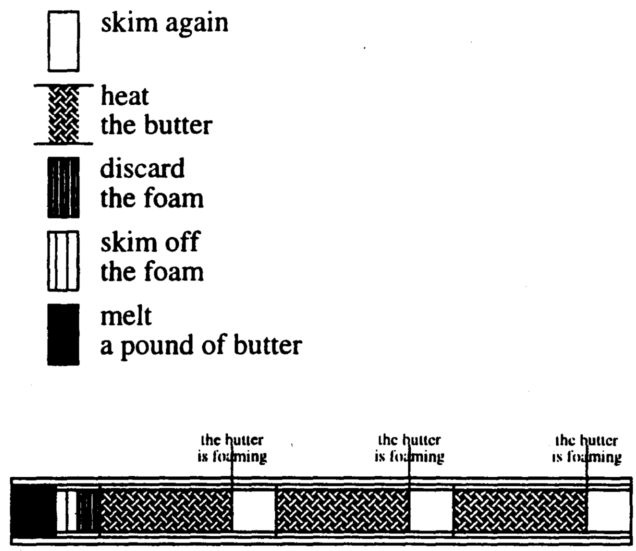


This section ends with a more complex example, of a discourse of more than one sentence.

(10) Melt a pound of butter. Skim off the foam and discard it. Heat the butter until it is foaming and skim again. Do this twice more.

A semantic analysis[13] that might be produced for (10) is[14]:

```
[[750 [complex [1 2 5 8]]]
 [60  [happening [(melt) (a pound of butter)]]]
 [60  [complex [3 4]]]
 [30  [happening [(skim off) (the foam)]]]
 [30  [happening [(discard) (the foam)]]]
 [210 [complex [6 7]]]
 [180 [mass [(heat) (the butter)]
           [(the butter) (is foaming)]]]
 [30  [happening [(skim again)]]]
 [420 [orderedset [2 5]]]   ]
```



[12]This and the following representation were both produced by the code developed.

[13]There are some interesting linguistic issues regarding how to resolve the anaphoric reference '*this*' in the discourse, which are not discussed here.

[14]`Complex` is used to process discourse—it covers events that are constructed of a number of unlike events, either from a number of sentences or from, say, the use of '*and*' to conjoin events.

5 Concluding remarks

5.1 Limitations

There are of course limitations to visualisation, and these come in two forms—those pertaining to visualisation itself and those pertaining to any computational implementation of visualisation. I only mention some of each of these; space restrictions here prohibit a fuller exposition.

One principal difficulty with visualisation as described in this paper is the implicit information that comes along with any picture. If a repeated event is represented by evenly spaced boxes, there is an implication that the repetition occurs at equal time intervals; boxes of similar sizes will be assumed to represent events that occur for similar lengths of time. In both of these cases, it is possible to imagine examples where the evenness or the temporal similarity should not be implied. The solution of say spacing the boxes unevenly, rather than removing the implication, may imply the opposite—that the events explicitly occur at random intervals.

A second problem is how to visualise concepts like '*a few times*' or '*a hundred times*'. For the first, representing three blocks may imply that '*three times*' is intended. For the second, do we draw one hundred blocks? How do we fit them all in? Or do we use dots to indicate some missing items?

With regard to the implementation described here, there are a number areas where things could be done differently. A time axis might be useful; however including a time axis where there is no actual reference to time, such as in (10) can also clutter the visualisation. There have been many examples that I have mentioned where my analysis is not comprehensive. For example, what of mechanisms for visualising more than two concurrent events, and also concurrent events that do not have the same start and end times as each other. Extending the analysis would entail defining other ways in which the basic building blocks could be combined into structured eventualities; it would then be a matter of coding a PostScript routine for each additional structure. I argue that the modular approach taken would facilitate this kind of extension.

5.2 Related work

The work presented here is related to work on understanding instructions (e.g. [5],[4]); work on the object-eventuality analogy ([6],[9], [16]; work on visualisation of events and situations (e.g. [8], [10]). However, none of these deal with the issue of extracting and visualising information that is about repeated, extended or multiple eventualities.

There is an obvious connection with work in artificial creativity. We can see music and dance as artforms that exist in the eventuality domain and fine art existing in the object domain. It is clear that one aspect of aesthetic appeal in art comes from the recurrence of themes[7]; the repetition of musical phrases is a good example, while visually, say in a painting, patterns of repetition may be appealing. It would be useful to extend some of the models that exist of creativity[7] to include aspects of repetition and massness, in both the object and eventuality domains.

5.3 Summary

This paper describes an illustrative segment[15] of an approach to representing the understanding of language about event sub-structure. Given a semantic analysis of the repetition described by instructions, it is possible to produce a visual image of this. PostScript's functionality is exploited to provide a simple back-end to semantic analysis that is clean and modular. Additional event sub-structures can be included in a modular fashion.

Although the visualisation has some limitations, it is very useful in providing a means of representing semantic information in a medium that is distinct from the conventional ways of doing this, such as logical forms. Moving to a different medium has advantages that include being able to represent information that could not otherwise be easily represented, such as the passage of time. Also, applying the results of semantic processing to another application can be a good test of the validity of semantic output[8].

References

[1] Adobe Systems Incorporated. *The PostScript Language Reference Manual.* Addison-Wesley, 1985.

[2] Allen, J. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 11(26):832–843, 1983.

[3] Bach, E.A. The algebra of events. *Linguistics and Philosophy*, 9:5–16, 1986.

[4] Dale, R. *Generating Referring Expressions in a Domain of Objects and Processes.* PhD thesis, University of Edinburgh, 1989.

[5] How, K. Y. *A Processing Framework for Temporal Analysis and its Application to Instructional Texts.* PhD thesis, University of Edinburgh, 1993.

[6] Jackendoff, R. Parts and boundaries. *Cognition*, 41:9–45, 1991.

[7] Lansdown, J. Artificial creativity: An algorithmic approach to art. In C. Beardon, editor, *Digital Creativity: Proceedings of CADE 95*, pages 31–35, 1995.

[8] Ludlow, N. D. *Pictorial Representation of Text: Converting Text to Pictures.* PhD thesis, University of Edinburgh, 1992.

[9] Mayo, B. Objects, events and complementarity. *Philosophical Review*, LXX:340–361, 1961.

[10] Novak, Gordon S. Jr and Bulko, William C. Understanding natural language with diagrams. In *AAAI 8th national conference on Artificial Intelligence*, pages 465–470, 1990.

[11] Okada, N. Integrating vision, motion and language through mind. *Artificial Intelligence Review*, 10:209–234, 1996.

[12] Rock, S.T. Understanding repetition in natural language instructions - the semantics of extent. In *Proc. 30th Annual Meeting of the Association for Computational Linguistics*, Delaware, USA, 1992.

[13] Rock, S.T. *Understanding natural language about multiple eventualities and continuous eventualities.* PhD thesis, University of Edinburgh, 1996.

[14] Rock, S.T. More readings than I thought — quantifier interaction in analysing the temporal structure of repeated eventualities. In *SAICSIT annual conference*, South Africa, November 1997.

[15] Talmy, L. The relation of grammar to cognition - a synopsis. In D. Waltz, editor, *Theoretical Issues in Natural Language Processing 2*. Association for Computing Machinery, New York, 1978.

[16] Zemach, E. Four ontologies. In F. J. Pelletier, editor, *Mass Terms : some philosophical problems*, pages 63–80. D.Reidel, Dordrecht, 1979.

[15] Some detail has been omitted in order to simplify the exposition.

South African Computer Journal

Number 24, November 1999
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 24, November 1999
ISSN 1015-7999

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 $1\frac{1}{2}$ 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. References should also be 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 2_ε. 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.