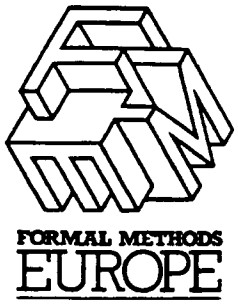

FACS Europe

The Newsletter of the BCS Formal Aspects of Computing Science Special Interest Group and Formal Methods Europe.

Series I Vol. 2, No. 4, Spring 1996 ISSN 1361-3103

Contents

Editorial

Welcome to the Spring '96 issue of FACS Europe. We're a bit thin on material this time - so anyone out there with a tale to tell, or an axe to grind, please get writing, so we can have more beef (or mutton, perhaps...) in the next issue.

In the next issue there will be a review of useful and interesting electronic information resources, and contributions are requested to help get good coverage of the topics of interest to our readers. Please keep contributions pithy and brief, and include all relevant URLs etc.

Notices

We have had a suggestion for a workshop next year on 'B'. Please contact Ib Sorensen *IB@comlab.ox.ac.uk* if you would be interested in getting this to happen. Also see the 'B' website,
http://www.tees.ac.uk/bresource/b.html

The FACS website has changed (virtual) location, and is now to be found at *http://cs-fm.lboro.ac.uk/facs* (You are intended to read cs-fm as Computer Studies - Formal Methods).

ZUM '97 details can be found on *http://www.cs.reading.ac.uk/archive/z/zum97/*

The Formal Aspects of HCI workshop is on 10-12 September, at Sheffield. Contact *C.R.Roast@shu.ac.uk* for details.

BCS-FACS Xmas Workshop 1996 is a joint event with the BCS Requirements Engineering SIG, focussing on challenges and synergies currently felt between Formal Methods and Requirements Engineering. There will be full details in the next newsletter issue; for now please note that 16-17 December, at City University, will be the place to be for Xmas 96.

CSIRO (Australia) are looking for two senior researchers on their software ansd systems engineering programme. Further information on *http://www.cbr.dit.csiro.au/division/*

There is an EPSRC sponsored short course on Safety Critical Systems on 16-20 September 1996, at Imperial College, London. Details from Sally Verkaik, Imperial College Continuing Education Centre Tel: (+44)171 594 6882/1 Fax: (+44)171 594 6883 E-mail: *cpd@ic.ac.uk*

Contributions Welcome...

Contributions to the Newsletter on any relevant topic are welcome. Please send them electronically if possible, in LaTeX or TeX form if you can; next best is plain ASCII. Otherwise please send A4 copy fit to reproduce by fast photocopying (i.e. no paste-ups), with 300dpi laserprint or equivalent a minimum standard.

FACS/FME Newsletter
c/o Ann M Wrightson
School of Computing and Mathematics
University of Huddersfield
Queensgate
Huddersfield
HD1 3DH
UK

Contributions express the opinions of contributors, not of FACS, FME or any other organization with which they are associated (unless they say otherwise!).

Letters are welcome and should be sent to the Editor.

Advertising

We have had a couple of enquiries about advertising in FACS Europe. Advertisements are welcome, as full or half page printed ads, or as inserts (i.e. loose sheets or booklets mailed with the Newsletter). Advertisements and inserts will only be accepted where they are clearly of specific interest to the FACS/FME community. Please contact the editor for current rates and due dates for copy.

9th International Conference of Z Users

ZUM'95

Organised by the Z User Group
Sponsored by BT, Forbairt, Praxis and the University of Limerick
Supported by BCS FACS and ESPRIT ProCoS-WG
7-8th September 1995
Department of Computing, University of Limerick, Ireland,

The conference was well attended and was ably hosted by the University of Limerick. The standard of presentation was very high with speakers from all over the world. Invited speakers were Professor David L. Parnas (McMaster University, Canada), Dr. John Rushby (SRI International, USA), Professor Jeannette M. Wing (Carnegie Mellon University, USA) and Professor David Gries (Cornell University). The main conference opened with a message of welcome which had been sent by Mary Robinson, President of Eire.

David Parnas gave a provocative talk: *Language-Free Mathematical Methods for Software Design* suggesting that new notations played too great a role in specification. He remarked that in traditional engineering disciplines, mathematical methods are *not* introduced by teaching languages. For example Electrical Engineers learn circuit design in three distinct courses: mathematics, physics and engineering. He viewed this "separation of concerns" as extremely important, in distinguishing syntax from method and semantics. He went on to distinguish between *descriptions* and *specifications*, in that "the statement that a product satisfies a given specification constitutes a description". He advocated a simple tabular notation combined with predicate calculus for specifying programs.

[*The talk generated a response from John Nicholls of Oxford University, which we hope to publish in FACS Europe in the near future.*]

John Rushby gave a talk entitled: *Mechanising Formal Methods: Opportunities and Challenges.* He described both the opportunities created by mechanised formal methods, and the technical challenges in effective implementation. The talk was based on the author's experience in the development and use of the PVS theorem prover. The importance of establishing correct requirements was emphasised: 50% of the critical faults discovered during integration testing of JPL spacecraft were due to flawed requirements. The use of mechanisation was advocated in e.g. calculating properties of formally specified designs, and in adapting to changed requirements.

Jeannette Wing presented: *Specifications and Their Use in Defining Subtypes* (joint paper with Barbara Liskov). The paper was concerned with the interaction between specifications and types hierarchies and described a new way of showing how one type is a subtype of another. The method provided a specification technique for object types that allowed creators to be specified separately from types. The specifications were based on Larch, and contained explicit constraints identifying a minimal set of history properties that methods of the type and all its subtypes must preserve.

David Gries (Cornell University) gave a presentation at the Educational Session, *Equational Logic: A Great Pedagogical Tool for Teaching a Skill in Logic.* The tutorial was for people who teach logic or discrete mathematics and described a 6-7 week course on equational logic. (Substitution of equals for equals is the dominant inference rule.) Students soon gain a skill and hence lose some of their fear of the subject; students also see that formal logic is *useful.* The text "A Logical Approach to Discrete Math" by D Gries and F B Schneider (Springer-Verlag, 1993) accompanied the talk. (See: http://www.cs.cornell.edu/Info/People/gries/gries.html.)

The main conference concluded with a vote for best paper(s) and two papers tied for this honour. These were *An Algebraic Proof in VDM*♠ by Arthur Hughes and Alexis Donnelly (Trinity College, Dublin), and *Testing as Abstraction* by Susan Stepney (Logica, UK). The first paper presented an algebraic, constructive style of specification and proof used in the Irish School of VDM; the example used to illustrate was a novel one however, viz. the Irish parliament or Dáil, and its T.D.s (*teachta dála* is member of parliament). Susan Stepney's paper described work on the PROST-Object project, a method of formally specifying tests based on systematic abstraction from a "state-plus-operation" style specification. (Titles of other papers are listed below,)

Social aspects of the conference included a magnificent dinner at Dromoland Castle, where we were accompanied by traditional Irish music. Another enjoyable feature of the conference was a Boat Trip on Lough Derg on the last evening. Evenings at the conference were also enlivened for some delegates by visits to Irish Music Sessions in the pubs of Limerick. Since the location was Limerick, "limericks" were an important part of the conference, which included a novel feature: a limerick competition. David Gries gave a memorable after dinner speech at Dromoland Castle, composed entirely of limericks! A sample follows:

> This conf'rence is all about Z.
> Well thats far too narrowly said.
> It's formality
> and its uses you see,
> ¿From which we all earn our good bread.

David Parnas concluded his talk with:

> A method that's simply called math,
> Will seldom incur people's wrath.
> So, abstract from the state,
> Show how traces relate,
> For specs people read in the bath.

Margaret West, University of Huddersfield;
David Till, City University

The conference proceedings have been published as:

ZUM'95: The Z Formal Specification Notation - 9th International Conference of Z User's, Limerick, Ireland, September 1995, Proceedings, Lecture Notes in Computer Science 967, Springer-Verlag, Heidelberg.

This contains papers presented in the Main meeting and the Educational issues session; contents of the Main meeting are given below.

ZUM'95 Programme: 7-8th September 1995

Opening Remarks and welcome message sent by Mary Robinson, President of Eire. — Jonathan Bowen, Oxford Univ., UK (Conference Chair); Mike Hinchey, NJIT, USA & Univ. of Limerick, Ireland (Programme Chair)

Methods

Chairs: Jonathan Bowen & Mike Hinchey

Language-Free Mathematical Methods for Software Design. — David Lorge Parnas, McMaster University, Canada (*invited speaker*)

A Formal Approach to Software Design: The Clepsydra Methodology. — P. Ciaccia, P. Ciancarini & W. Penzo, Italy

Refining Database Systems. — David Edmond, Australia

Applications I

Chair: Chris Sennett

Structuring a Z Spec. to Provide a Formal Framework for Autonomous Agent Systems. — Michael Luck & Mark d'Inverno, UK

On the use of Formal Specs. in the Design and Simulation of Artificial Neural Nets. — P. Duarte de Lima Machado & S. L. Meira, Brazil

Structuring Specification in Z to Build a Unifying Framework for Hypertext Systems. — Mark d'Inverno & Mark Priestley, UK

Proof
Mechanizing Formal Methods: Challenges and Opportunities. — **Chair**: David Till; John Rushby, SRI International, USA (*invited speaker*)
An Algebraic Proof in VDM♣. — Arthur Hughes & Alexis Donnelly, Ireland

Testing
Testing as Abstraction. — **Chair**: Elspeth Cusack; Susan Stepney, UK
Improving Software Tests using Z Specifications. — Hans-Martin Hörcher, Germany
Compilation of Z Specifications into C for Automatic Test Result Evaluation. — Erich Mikk, Germany

Language
Equal Rights for Schemas in Z. — **Chair**: Neville Dean; Sam Valentine, UK
Structuring Z Specifications: Some Choices. — Anthony MacDonald & David Carrington, Australia
Experiments with the Z Interchange Format and SGML. — Daniel M. Germán & D. D. Cowan, Canada

Panel Session
The Future of Industrial Formal Methods. — Participants: Anthony Hall, Nico Plat, David Parnas, John Rushby, Chris Sennett

Object Orientation
Specifications and Their Use in Defining Subtypes. — **Chair**: Elspeth Cusack; Jeannette M. Wing, Carnegie Mellon University, USA (*invited speaker*) (joint paper with Barbara Liskov, MIT, USA)
How Firing Conditions Help Inheritance. — Ben Strulo, UK
Extending W for Object-Z. — Graeme Smith, Australia

Applications II
A Formal Semantics for a Language with Type Extension. — **Chair**: Jim Woodcock; Peter Bancroft & Ian Hayes, Australia
From Z to Code: A GUI for a Radiation Therapy Machine. — Jonathan Jacky & Jonathan Unger, USA
The French Population Census for 1990. — Pascal Bernard & Guy Laffitte, France

Animation
Implementing Z in Isabelle. — **Chair**: Sam Valentine; Ina Kraan & Peter Baumann, Switzerland
The Z-into-Haskell Tool-kit: An Illustrative Case Study. — Howard S. Goodman, UK
Types and Sets in Gödel and Z. — Margaret West, UK
Exploring Specifications with Mathematica. — Colman Reilly, Ireland

Method Integration
Using Z to Rigorously Review a Specification of a Network Management System. — **Chair**: Mike Hinchey; Tony Bryant, A. Evans, L. Semmens, R. Milovanovic, S. Stockman, M. Norris & C. Selley, UK
A 2-dimensional View of Integrated Formal and Informal Specification Techniques. — Robert B. France & Maria M. Larrondo-Petrie, USA
Viewpoints and Objects. — Howard Bowman, John Derrick & Maarten Steen, UK

Existence and Intuitionistic ∀-Elimination (*summary*)

Alan Hutchinson
King's College London

Keywords

Predicate logic, Lambda calculus, Constructive logic

Abstract

There is a straightforward constructive form of higher order logic generated by ⇒, the existence predicate **E** of Fourman and Scott, and predicate variables. The introduction and elimination rules of conventional intuitionistic logic can be interpreted in it if one adds a condition to the ∀*E* rule, that the term substituted for the bound variable must exist. This seems a very natural assumption. There appear to be two kinds of construction involved: object level constructions with combinators, and meta level ones which require syntactic analysis. The intuitionistic negation rule can also be interpreted if one allows some substitutions of wffs for wffs, but substitution is not obviously constructive. Several untidy aspects of intuitionism, particularly the Curry-Howard correspondence, appear neater in the new form.

Introduction

Consider the following reasoning.

The Jabberwocky is dangerous;
all dangerous creatures should be avoided;
hence, one should make some effort to avoid the Jabberwocky.

When did you last make a serious attempt to avoid the Jabberwocky? Which is at fault: your behaviour, or the logic?

One natural setting for intuitionistic logic is with higher order types. It is usually written in a natural deduction style, using introduction and elimination rules for ⇒ and ∀:

$$\forall I^{*}\ \frac{\Gamma:\ \varphi}{\Gamma:\ \forall x\varphi} \qquad \forall E\ \frac{\Gamma:\ \forall x\varphi}{\Gamma:\ \varphi[t/x]}$$

These do not fit neatly into other aspects of the subject.

- According to the Curry-Howard correspondence, there are two distinct elimination rules which correspond with β-reduction. The first is $\Rightarrow E$, *modus ponens*, which matches β-reduction nicely. The other is $\forall E$. In λ calculus, there seems to be no essential difference between a β-redex corresponding to $\Rightarrow E$ and one corresponding to $\forall E$. This is odd.
- The Curry-Howard correspondence for $\forall E$ is not neat. It involves a term t which does not occur in the premiss of $\forall E$.
- The Brouwer-Heyting-Kreisel explanation of $\forall$ seems unduly elaborate.

Below is presented a slight modification of the usual form of this logic.

Existence

In the accounts by Fourman and Scott, there is an *existence predicate* **E**. If we use **E**, then the $\forall E$ rule should read

$$\forall E \quad \frac{\Gamma : \forall x \varphi \qquad \Delta : \mathbf{E}t}{\Gamma \cup \Delta : \varphi[t/x]}$$

For practical purposes, $\forall x \varphi$ is a shorthand for $\mathbf{E}x \Rightarrow \varphi$.

The language of $\Rightarrow$ and E

As in (Fourman 1977) and (Scott 1977), $L_{\mathbf{E}}$ involves a set called *Sort*, with a (meta-)function called the *power-type map* from $\bigcup_{n \geq 0} Sort^n$ to *Sort* written as

$$(A_1, \ldots A_n) \longrightarrow [A_1, \ldots A_n].$$

The sort [] is thought of as the sort of *truth values*. $[A_1, \ldots A_n]$ is the sort of predicates of arity n whose arguments are of sorts $A_1, \ldots A_n$.

The *symbols* of $L_{\mathbf{E}}$ are parentheses, (and); a conective called $\Rightarrow$; for each sort A in *Sort*, a predicate called $\mathbf{E}_A$; for each sort A, a countable set of variables x_i^A ($i \in \mathbb{N}$). The *well-formed formulas* (wffs) of $L_{\mathbf{E}}$ are $x^{[A,\ldots B]}(y^A, \ldots z^B)$ for any variables $x, y, \ldots z$ of these sorts; $\mathbf{E}_A x^A$ for any variable x^A whose sort is A; $(\varphi \Rightarrow \psi)$ whenever φ and ψ are wffs. Note that $x_i^{[]}()$ is a wff. These particular wffs may sometimes be treated like propositional

variables. Any free occurrence of x in ψ is *bound* in $(\mathbf{E}x \Rightarrow \psi)$ by this occurrence of $\mathbf{E}x$.

There are just five inference rules for $L_{\mathbf{E}}$: *Axiom*, $\Rightarrow I$, $\Rightarrow E$, *Thin*, and *CBV* (Change Bound Variable). It is also useful to use Sub_0: substitution of a wff for $x^{[]}()$.

> Sub_0 is not derivable from the other inference rules.

Curry-Howard correspondence

> There is a bijection between *CBV*-equivalence classes of proofs and classes (under α-conversion) of typed λ-terms, typed by *CBV*-equivalence classes of wffs.

A wff is *normal* iff it is of one of the three forms $x(y,..z)$, $\mathbf{E}x \Rightarrow \varphi$ where φ is normal, $\varphi \Rightarrow \psi$ where φ and ψ are both normal. Abnormal wffs are those of the form $\mathbf{E}x$ and those which contain a sub-wff of the form $\varphi \Rightarrow \mathbf{E}x$.

Terms and the language $L_{\forall}$

The alphabet of $L_{\forall}$ consists of the alphabet of $L_{\mathbf{E}}$ and the symbols

$$\forall \quad \wedge \quad \Leftrightarrow \quad \exists \quad = \quad \mathbf{I}.$$

Its *wffs* are the normal wffs of $L_{\mathbf{E}}$. Among them,

$$\forall x\varphi \qquad \varphi \wedge \psi \qquad \varphi \Leftrightarrow \psi \qquad \exists x\varphi \qquad t^{A} = u^{A}$$

are abbreviations. Its *terms* are variables x^{A} and defined terms $\mathbf{I}x\varphi$, meaning "the unique thing x satsifying property φ". Wffs involving $\mathbf{I}$ are also abbreviations.

Theorem

> The rules of normal logic can be derived from those of $L_{\mathbf{E}}$, except perhaps $=E$. (Unless one uses substitution, the proof only works for a modified form of the $\exists E$ rule. Negation also requires substitution.)

Conjecture

> The rule $=E$ can also be derived if one uses substitution.

Comments

- *Rules* $\Rightarrow I$, $\Leftrightarrow I$, $\forall I$, $\forall E$, $\exists I$ can each be derived by a *combinator* from proofs of their premisses.

  Rules $\wedge E1$, $\wedge E2$, $\Leftrightarrow E1$, $\Leftrightarrow E2$, $\exists E$ can (as far as I know) only be derived by *syntactic analysis* of proofs of premisses.
- The language L_E is considerably simpler than Fourman's and Scott's. Fourman's language, for instance, has five basic symbols besides variables and constants, and fourteen axiom schemes and rules of inference.
- The logic of L_E-wffs which are not necessarily normal remains to be explored.
- The notion of substituting wffs for wffs in a predicate calculus also seems not fully explored.
- In some forms of logic based on Martin-Löf type theory, implication ($\Rightarrow$) is treated as a special case of $\forall$ rather than the other way round.
- There are much more ambitious theories of the structure of proofs and constructive logics.

References

M P Fourman 1977

The logic of topoi

pages 1053-1090 in Barwise 1977

D S Scott 1977

Identity and existence in intuitionistic logic

pages 660-696 in

Applications of sheaves (Proceedings, Durham 1977)

M P Fourman, C J Mulvey & D S Scott (editors)

Springer-Verlag 1979

Situation Semantics
Tutorial Notes

Ann M Wrightson

December 14, 1995

1 Introducing Situations

The key idea caught by the notion of a situation, and the theory of meaning which goes with it, is the way we all manage to talk about the complexity of the world using relatively simple signals. If I say 'The bus was late again this morning.', then your recognition of a familiar situation allows that simple sentence to explain my late arrival and short temper. Notice that this only works if you and I have sufficient common experience or knowledge for you to understand what it is I am talking about. You may even be able to tell me more about the situation, for example by replying 'Yes, there was an accident on the ring road.' This brings out another characteristic of situations we recognize in everyday life - they are not completely known, and two people can talk about the same situation, each contributing observations new to the other. Situations can be very fuzzy like the late bus example above, or they can be clearly delimited, either by the participants (eg a football game, a circus performance) or by others (eg an accident under investigation). Here are a some more examples:

> s1: After Monday's maths class, Pat and Kay get together with several other students who were there, including one who missed the previous class, to compare notes and work together on some problems.

A series of classes, missing a class, a group of students working together, comparing notes, working problems: the new situation s1 is described in terms of these familiar (kinds of) situations. Note here that situations can be grouped together, recognized and characterized, as being of various kinds or types. This is central to situation semantics.

> s2: I listen to two of my colleagues discussing a rugby match.

I don't play rugby, and have only a very basic understanding of the rules and patterns of play. I learn about the match, but very imperfectly; much of what they say leaves me with pretty vague and probably pretty faulty impressions of what happened. Yet my colleagues are responding to these same utterances in a way which tells me that to them they are vivid and precise. The meaning conveyed by what they say to each other is clearly depending very much on connexions between what is said and the actual described situation, the match they both watched on Saturday. Also involved is the 'rugby game' situation type, with all its intricacies of rules and patterns of play. In the situation I have in mind, my poor understanding was due to not knowing much about either the the specific game or the 'rugby game' situation type.

> s3: An experienced requirements analyst, unfamiliar with the application area, listens to two domain experts discussing the intended role and scope of a proposed new system.

Although s3 is superficially similar to s2, it is nevertheless a very different situation. The role of the analyst in s3 is quite different from my role in s2, and this would come out in a fuller description of the analyst's behaviour. For example, the analyst would pay close attention to the speakers, take notes, and ask for clarification of unfamiliar terms - whereas in s2 I am happy to relax and drink my coffee.

2 Situation Semantics

Situations don't have to include any language, but those that do are of particular interest; in fact the abstract notion of a situation was originally developed to help in investigating how natural languages support meaningful conversation. The account of the meaning of natural language utterances coming out of this line of investigation is known as situation semantics. I have two reasons for using it. The first could be considered aesthetic, in that I like the account it provides of the meaning of, considered as information conveyed by, natural language. The second is that situation semantics provides some interesting reflections on the basis for meaningful communication, both between humans and machines, and between humans through the medium of stylized, restricted forms of communication such as computer system specifications.

In order to see how it works, first I need to introduce some of the underlying concepts, then put these together to provide an account of meaningful utterance.

2.1 Agents and Individuation

The world according to situation theory contains many *agents* which make sense of each other and of the rest of the world in terms of *schemes of individuation*. Each agent has its own scheme of individuation: what this means is that the agents involved (eg people) not only perceive objects such as doors, pens, dogs, people, but also recognize them *as* doors, pens, dogs, and people. They also recognize *situations*, such as a coffee break, presenting a paper at a conference, taking a dog for a walk, rushing to catch a train. Different agents can, and generally will, have different schemes of individuation, which reflect their different capacities for distinguishing objects and situations.

Another key feature of schemes of individuation is that the objects are recognized as having (or not having) *properties* eg a dog may be black, a coffee break may be short; and also as having *relationships* eg that the black dog and me are related by it being my dog; that the black dog likes the situation of my taking it for a walk. A weaker form of individuation is *discrimination*. The distinction is that discrimination is evidenced by changed behaviour, while individuation is something like being able to conceptualize or talk about it as well.

When an overall scheme of individuation is used for a scope including several agents, then it is traditional in situation semantics to call the agent involved (often the researcher) the *theorist* i.e. objects and situations discussed are implicitly individuated by the theorist, or described as individuated by another agent (as seen by the theorist!).

How to contain the possible pathological fuzziness? Well, meaningful communication has a role in sustaining a common repertoire of concepts, but how this happens is outside the scope of situation semantics. However, the situation semantics account does sit quite well with eg [Eco94]; also [Bar89] contains discussions of various general issues of this kind. The rationale behind the foundational place of agents and schemes of individuation in situation semantics is discussed thoroughly in [Dev91].

2.2 Situations, individuals, properties and relations

2.2.1 Situations

A *situation* is an abstract model of our everyday experience of being in a situation, for example the situation I am in at present of sitting writing these words. Also taken from everyday experience, into the idea of a *situation-type*, is our ability to recognize various kinds of situations, for example all the occasions I have sat preparing a document. Notice that a situation naturally carries with it a time and place, and that talking about kinds of situations (and hence a situation-type) naturally carries with it the relevant common features of situations at different times and places. These times and places can be short or long, large or small.

2.2.2 Individuals

An individual is something individuated by a scheme of individuation. An individual can be an object, an agent, a situation. Individuals considered as given. Not atomic - eg table plus components. This very general notion needs to be made more precise when situations are modelled formally, eg as part of a requirement model for a computer system. [Wri95]

2.2.3 Properties, Relations

A relation is something which holds or fails to hold (or some intermediate/other if you want your logic to do that) of a number of individuals. Relations hold or fail to hold between particular identified individuals, or between some or all individuals. A property is similar, but only involves one individual.

Each property or relation has argument roles which need to be filled by appropriate individuals. This appropriateness is considered to be a given feature of the world as it is, not constructed in the theory. For example, it just is inappropriate to have a tree and a mountain in a relation of being complementary colours; for black and purple it is appropriate, but the relation does not hold; whereas for red and green it is appropriate, and also holds.

Properties and relations are not themselves individuals (unlike a situation where a particular relation holds, which is).

Situation semantics is usually developed on the basis of properties and relations simply holding or failing to hold. There is no reason in principle not to allow variation of logics between situations, though clearly this needs to be handled carefully in any formal model. (Formalization of situations using *labelled deduction* [Gab94, Gab93] provides a suitable framework, which is used for formalizing viewpoints in requirements in [Wri].)

2.3 Infons

An infon is a formalization of a single piece of information. (The concept of 'infon' is also a formalization of the concept of a single piece of information, though that's not quite the same thing; for a detailed discussion of this, see [Dev91].)

For example, an infon may represent the information that my dog Ben is running at time t:

$$\ll running, Ben, t, 1 \gg$$

And to represent the information that my dog Ben has no collar on at time t:

$$\ll wearing, Ben, collar, t, 0 \gg$$

The '1' or '0' at the end is the *polarity* of the infon; 1 is positive, 0 is negative. Since there is a subtle but significant different between an infon and an assertion, this is a distinct concept from 'true' and 'false' (though it is of course related, via the situation semantics account of the meaning of an assertion, and is also used to represent facts about the world which are represented by true/false statements in other abstract representations).

More precisely: information is arranged into items called infons, of the form: objects $a_1, \ldots a_n$ do/do not stand in relation P. P is some (property or) relation, $a_1, \ldots a_n$ are objects appropriate to their roles in P. The identification of the objects is not part of the infon's content—the infon is assumed to pertain to certain given objects. Notation:

$$\ll P, a_1, \ldots a_n, 1 \gg$$

$$\ll P, a_1, \ldots a_n, 0 \gg$$

1/0 is the polarity - where an infon corresponds to the way the world is, it's a fact. Situations may be constituents of infons.

A situation may *support* an infon, $s \models \sigma$. Informally, *sigma* is 'true in' s—and this notion is fundamental, i.e. for a particular situation and infon it is a fact of the world whether the

relation $s \models \sigma$ holds or not. (It is worth noting that there is a significant and long-running debate in the situation theory community about how, if at all, the usual mathematical notion of *set* is appropriate for modelling situations and infons.)

If w is the (unique) real world, then $w \models \sigma$ is a representation of 'σ is a fact'.

Infons are semantic objects, not syntactic representations. This means that (rather like the names of things and properties in mediaeval logic) the relation of the components to what they represent is taken as basic and unproblematic—or rather that any problems you care to raise belong to some other discipline, eg philosophy of language!

2.3.1 Locations in space and time

A location is an individuated point or region of space or time. Locations may overlap, be part of another location, and so on. In infons, each relation which includes locations comes with its own set of argument-places (and appropriate roles), which determine the kinds of locations being considered in that context.

Even if not individuated by the agent, eg represented from the theorist's point of view, locations will depend on the agent, eg in being of appropriate size and precision/fuzziness, and this dependency they share with uniformities individuated by the agent itself.

2.4 More about Situations

Situations are primitive in the ontology of situation theory, i.e. not derived or constructed from other notions. The theory reflects what agents are observed to do, i.e. discriminate (and individuate) situations; there is clear empirical evidence for this being quite precise in humans, eg [Boy88]. More generally, the behaviour of people and mechanical devices varies dramatically and systematically according to the situations they are in. Situation theory is not itself a formalized abstraction, but a qualitative theory which has been used to underpin several formal theories, using various mathematical and logical assumptions. [CMP90, BGPT91, AIKP93] The one I favour for my work on semantics in human computer interaction and requirements is based on [Gab93].

2.4.1 Abstract Situations

Abstract situations are extensionally characterized analogues of actual situations, where what characterizes an abstract situation is a set of infons which it supports. Abstract situations are an obvious starting point for formalizations of situation theory.

Given a real situation s, the set $\{\sigma | s \models \sigma\}$ represents a corresponding abstract situation. This doesn't go backwards—there is generally no real situation s corresponding to an arbitrary set σ of infons. This is a consequence of the abstraction chosen (set of infons) does *not* correspond to the concept being modelled. For example, it is possible to construct incoherent abstract situations, which are sets of infons which cannot correspond to a real situation because they contain contradictions, either formal logical ones, or ones which involve real-world consequences of the information represented (eg because one object is actually lighter than another). We can sometimes exclude incoherent abstract situations in practice (though the subtler kinds of incoherence are clearly not effectively determinable), and will often wish to do so in theory. (Beware forgetting about them altogether!)

3 Natural Language Semantics

Having covered some background concepts, now to building out of them a theory of meaning for natural language. Let's take another look at one of the situations described earlier on.

s2: I listen to two of my colleagues discussing a rugby match.

The interesting thing here is that much of what they say leaves me with pretty vague impressions of what happened, yet my colleagues clearly find it vivid and precise. The same bits of language are conveying very different meanings to different people. How can that be?

Part of the answer is that the conversation is drawing on two situations individuated by both the speakers: the match they both watched on Saturday; and a generalized 'rugby game' situation. The match is the *described situation*; the 'rugby game' situation is a *resource situation.*

There are three other situations which come into a full account of the meaning using situation semantics. The *utterance situation* is the situation of speaking an utterance, eg "Now the second one - that was neatly done if you like." which is part of the *discourse situation* - the whole conversation. There is also an *embedding situation*, eg the coffee break when it happened. The embedding situation can have a strong influence on the meaning of utterances, witness the popularity with comic writers of putting a character into a situation, with the wrong idea about what's going on, i.e. about what kind of situation it is.

Going back to s2, I hear the same words they do, yet I don't individuate the same situations. How can I get any meaning at all? Well, I guess that's because I substitute some general "team ball game" situation for the more precise "rugby game" situation, and then find some connexion between what they say about the particular game and other games I've watched. So I get some information, but not much, and not precise. This shows clearly the *relational* nature of situation semantics: the meaning is in the whole relationship between the utterance situation, the discourse situation, the embedding situation, the resource situation and the described situation. With the *meaning* relatively hard to put a finger on, especially where many situations are involved, it is better to focus on the *information* conveyed by an utterance.

3.1 Information Flow

Information flows to an agent as a result of an utterance in two ways. First, the utterance may itself be rich in information, so that the utterance situation itself can be said to contain certain infons. A richer source of information is the connexions made by a hearer, for example from the described situation to a resource situation. These connexions are usually called *constraints.*

Different agents will get different information from the same utterance because they will apply (be attuned to, in the usual situation semantics terminology) different constraints linking the utterance situation to other situations. Sometimes this is a matter of minor differences, different constraints between substantially the same situations. Sometimes different people link the same utterance situation to radically different situations they believe the utterance to be about, and so gain very different information from it. (Again, a classic device in comedy.)

Formalization of this aspect of situation semantics is quite a challenge, and various work is in progress in this area. A recent contribution is [BGH95].

4 Just Natural Language?

Situation semantics is intimately concerned with semantics as *import*, i.e. as what an utterance means-*to* an agent, rather with identifying some abstraction to stand alone as 'the meaning' of an utterance. Although originally devised to account for the meaning of human language, this makes it well suited to other agents, and communication by signs other than language. For example, it seems right to account for a dog understanding the meaning of a signal (eg if I put my coat on, or stand up and say 'come on, Ben, we're going out') through a simple relationship between two kinds of situation individuated or discriminated by the dog, rather than via grammatical structure etc. Conversely, if my dog communicates to me that he needs a drink of water by playing noisily with his empty water dish, that succeeds through a connexion between that kind of situation, and the situation where his dish is empty when he wants a drink.

Since machines are often built to discriminate various conditions, situation semantics carries over nicely to utterances from machines, with much less fuzziness about the nature and identity of the situations involved.

4.1 Alarm Clock

Consider an alarm clock, which has a face displaying the time, and can be set to ring a bell when the hands show some particular time, say 6-30am. When I wake and look at the clock, and see that it is 6am, then the clock has successfully conveyed to me the information that it is 6am. When the bell rings, then the clock has successfully conveyed to me the information that it is time to get up, or more precisely that the time which I decided last night was the time I should wake up this morning, has arrived.

This meaning for the time on the clock face is there because of a link (a constraint) between two types of situations: S0, where the clock shows a time of 6am, and S1, it being actually 6am. Because I am aware of that constraint (I have learned to tell the time) the clock can tell me it is 6am. (The clock face cannot communicate successfully to my young son, since he has not yet learned to tell the time.) Similarly, the meaning for the alarm is there because of a link between a situation of type S2 where the alarm goes off, and a situation of type S3, of its being time to get up.

What I want to examine here is the constraint which links S1 and S0. It has some connexion with how the clock works, since if the clock is not working correctly, eg is running fast, then the information that it is 6am will be misinformation - perhaps it is actually 5-45am. Yet the constraint is not dependent on the detailed working of that actual clock, for I could replace it with another clock with a different kind of mechanism (eg electric instead of clockwork), and the communication I am talking about would not be significantly different. Indeed, any of a number of quite different kinds of clock would do, with the form of the communication varying: a dial; a numeric display; a synthesized voice saying 'it is 6am'. All these could be part of situations of type S0, and all can have the required link with S1.

Perhaps more can be seen about this constraint by seeing how the communication can fail. One example has already been mentioned—the clock may be wrong. It is still telling the time, but not doing it right. For example, the clock may be running fast, it may have been set ten minutes fast, or it may show Tokyo time. Or I may be interpreting as a clock something which is not a clock at all, eg reading a dial as a clock when it is a barometer, or reading a numeric display as time of day when it is showing day of month. In the case of a clock being set wrong, or showing Tokyo time, I can still use it to provide me with the same information if I change the constraint so that eg a clock showing 6am, which I know to be 15mins fast, conveys to me that it is 5-45am. Similarly, if I pass a clock showing 6-30am Tokyo time, and it has a sign by it saying 'Tokyo', then it can at least convey that it is half-past some hour, and if I can remember how many hours difference there is, I can work out the full local time. The other cases are hopeless. (A stopped clock belongs to this last group - a clock stopped at 6am is not actually telling me the time at all, even if it happens to be 6am when I look at it.)

4.2 Words uttered by a machine

A machine often communicates information to its operators using text. Sometimes the text gives straightforward information about the machine. Often the text is intended to be understood as if it were uttered or written by a human performing a task analogous to that being performed by the machine. Also, the text is often put together from several parts, mimicking more or less successfully the construction of complex utterances in natural language. Although this kind of communication is widely used, it is not free of problems; misunderstanding and misinformation can occur, which can of course have dire consequences in critical systems.

According to situation semantics, the meaning of this communication is determined by constraints between situation-types in the real world and situation-types characterized by particular machine utterances (for example, displayed text). These constraints are partly due to the nature of the machine, the way it works, and partly to do with the way it is used, the way it is expected to work by the people receiving the communication.

A simple example is a switch on a machine, which displays the word 'ON' when the machine is on, and 'OFF' when it is off. (The switch I have in mind is mechanical, but the technology doesn't

matter here.) Even in this simplest case, the veracity and effectiveness of the communication depends on a number of factors. Without attempting an exhaustive analysis, these must include the design and construction of the machine, in particular the fact that the switch is actually connected to the rest of the machine so as to operate as an on/off switch; the existence, and applicability to the device in question, of the concept of a machine having a two-state switch as its primary control; the linguistic conventions surrounding the use of the words *on* and *off* to talk and write about those two states; and the use of the graphics ON and OFF to represent those words.

The meanings of ON and OFF on the switch provide links between four kinds of situations: where the machine is on, where the machine is off, where the switch says 'ON', and where the switch says 'OFF'. Using S_1, S_2, S_3, S_4 for these situation-types, then what is required is:

$$S_1 \Rightarrow S_2$$

and

$$S_3 \Rightarrow S_4$$

where $\Rightarrow$ is *involvement*, which is a uniform connexion between two situation-types expressing a *constraint* linking the two. These constraints appear to be parts of what should be an indivisible whole, 'a meaning for the on/off switch'. However, the meanings, and respective constraints, for ON and OFF can be distinguished by considering a fault in the switch which leads to the machine always being off when the switch says 'OFF', but sometimes not being on when the switch says 'ON'. This affects only the link between S_3 and S_4. So merging them is not a good idea.

This kind of constraint could be called a *system constraint*, since the existence of the constraint is dependent on the correct functioning of the system. The constraint is also an effective limitation on what can correctly implement the system - in this case, what can fulfil the role of an on/off switch. This suggests an interesting relationship between the specification of a system (as discussed in the literature on formal specification) and the meaning of communications from the system. In some sense the meaning of the communications from a system *depend on the system specification.* Or rather, bearing in mind the example of a child who cannot tell the time, that *that part of the specification which links the machine utterance to associated situation(s)* is what makes it possible for a machine utterance to convey meaning determined by those constraints.

However, it is also necessary (and backed up empirically by research [RBW88]), just as in human-to-human communication, for the recipient to have some awareness of the constraint, i.e. this awareness affects the meaning conveyed. This accounts neatly for how the same display content can convey different things to different people. For example, someone making routine use of a system encounters an obscure error message, which conveys to them no more than that a severe fault has occurred. The same message then conveys quite a lot more meaning to the engineer called in to rectify the fault, who knows more about the system.

References

[AIKP93] P Aczel, D Israel, Y Katagiri, and S Peters, editors. *Situation Theory and its Applications Vol 3.* Number 37 in CSLI Lecture Notes. CSLI, Stanford, CA., 1993.

[Bar89] J Barwise. *The Situation in Logic.* CSLI Lecture Notes no. 17, 1989.

[BGH95] Jon Barwise, Dov Gabbay, and Chrysafis Hartonas. On the logic of information flow. *Bulletin of the IGPL*, 3(1), 1995.

[BGPT91] J Barwise, J M Gawron, G Plotkin, and S Tutiya, editors. *Situation Theory and its Applications Vol 2.* Number 26 in CSLI Lecture Notes. CSLI, Stanford, CA., 1991.

[Boy88] G A Boy. Operator assistant systems. In *Cognitive Engineering in Complex Dynamic Worlds*, pages 85–98. Academic Press, 1988.

[CMP90] R Cooper, K Mukai, and J Perry, editors. *Situation Theory and its Applications Vol 1.* Number 22 in CSLI Lecture Notes. CSLI, Stanford, CA., 1990.

[Dev91] K Devlin. *Logic and Information.* Cambridge UP, 1991.

[Eco94] U Eco. *Semiotics and the Philosophy of Language.* Macmillan, 1994.

[Gab93] Dov Gabbay. Labelled deductive systems and situation theory. In P Aczel, D Israel, Y Katagiri, and S Peters, editors, *Situation Theory and its Applications Vol 3*, number 37 in CSLI Lecture Notes. CSLI, Stanford, CA., 1993.

[Gab94] Dov M Gabbay. Labelled deductive systems, volume 1 — foundations. Technical Report MPI-I-94-223, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1994. Draft, work in preparation for publication via OUP.

[RBW88] E M Roth, K B Bennett, and D D Woods. Human interaction with an 'intelligent' machine. In *Cognitive Engineering in Complex Dynamic Worlds*, pages 23–69. Academic Press, 1988.

[Wri] Ann M Wrightson. Semantics of computer system requirements from multiple perspectives. *paper submitted to ITALLC 96.*

[Wri95] Ann M Wrightson. How computers mean what they say. Technical Report RR9511, School of Computing and Mathematics, University of Huddersfield, 1995.

Notice of Meeting and First Call for Papers

ZUM'97

10th International Conference of Z Users

Organized by the Z User Group in association with BCS FACS
Supported by the ESPRIT **ProCoS-WG** Working Group
Sponsored by BT and Praxis
3–4th April 1997
University of Reading, England

The 10th International Conference of Z Users (ZUM'97) will be held at University of Reading, England, in April 1997.

Z is a formal notation (formal method) widely used in both academia and industry, for the specification and verification of both hardware and software systems.

The programme committee invites authors to submit papers on or related to the formal specification notation Z, in particular, and formal methods in general, for presentation and inclusion in the published Proceedings to be distributed at the conference.

The conference will also include tool demonstrations and displays by publishers.

An educational issues session is planned to follow the conference, and tutorials may be possible on the days directly preceding and following the main sessions.

The following invited speakers will give presentations as part of the main sessions of the conference:

Egon Börger, University of Pisa, Italy
Anthony Hall, Praxis, UK
Nancy Leveson, University of Washington, USA

NOTE: THE SUBMISSION DEADLINE IS 16 AUGUST 1996

The schedule for submissions is as follows:

Submission of draft paper:	16th August 1996
Notification of acceptance:	15th November 1996
Final copy for Proceedings:	10th January 1997
Z User Meeting in Reading:	3–4th April 1997

Further Information

General enquiries about the meeting and the Z User Group may be directed to:

Jonathan Bowen (Conference Chair)
University of Reading, Department of Computer Science
Whiteknights, BO Box 225, Reading RG6 6AY, UK.
Email: J.P.Bowen@reading.ac.uk
Tel: +44-1734-316544 Fax: +44-1734-751994
URL: http://www.cs.reading.ac.uk/people/jpb/

On-line and up-to-date conference information may be found under the following World Wide Web URL:

http://www.cs.reading.ac.uk/archive/z/zum97/

Book Review

Designing Object Systems: Object-oriented Modelling with Syntropy (S. Cook and J. Daniels, Prentice Hall, 1994)

K. Lano

May 14, 1996

Currently there are (at least) 50 "object-oriented" methods in existence[1] with more being added all the time, so scepticism may greet yet another method with an obscure name being produced and claiming to be different to/better than all the others. However this is a significant book, both from the viewpoint of the formal methods field, and for object-oriented software engineering. It represents a continuation in the evolution of OO methods and notations beginning with OMT, Booch and Fusion, away from simplistic adaptations of data-flow diagram and ERA "structured" method approaches towards techniques that are directly relevant to OO systems. These more recent methods use interaction graphs (or *mechanisms*) to detail how objects collaborate to produce an overall effect, and use various techniques to abstract from the location of methods in classes. They also represent a trend towards greater acceptance of mathematical notation within industrial methods.

Particular innovations in Syntropy are the use of mathematical notation (Z-like set theory) to supplement class diagrams and explain their semantics, and the use of events to abstract from object responsibilities at the initial specification stage. Unlike Fusion, an extension of the statechart notation is used to describe both individual object behaviour and collective behaviour of all objects of a class. Concurrency is also treated in some depth. The semantics for the method however runs out soon after the data model; the more complex statechart notation is only given an informal semantics, although there is current work at Brighton University and Imperial College to rectify this.

The method covers both analysis and design, although it avoids the former term. Its abstract initial model (termed the "essential" model) uses OMT-like class diagrams and statecharts to describe events and objects in the domain,

[1] Speaker at ICECCS, Florida, 1995.

without allocation of methods to classes, and without defining message-passing strategies. Instead, events may be responded to by a number of objects, of different classes. A similar model is used at the software specification level, in "specification" models. The difference is that a statechart for a class may generate further events as a consequence of responding to events. The "implementation model" also uses statecharts and class diagrams, but adds interaction graphs to describe detailed behaviour, now seen as the sending of messages between objects.

There are defined transformations for moving from one level of abstraction to another, although these are not formalised. Recommendations for partitioning a system into "domains" and for interconnecting these domains are provided.

BCS FACS
Department of Computer Studies
Loughborough University of Technology
Loughborough, Leicestershire
LE11 3TU
UK
Tel: +44 1509 222676
Fax: +44 1509 211586
E-mail: FACS@lboro.ac.uk

FACS Officers

Chair	John Cooke	D.J.Cooke@lboro.ac.uk
Treasurer	Roger Stone	R.G.Stone@lboro.ac.uk
Committee Secretary	Roger Carsley	roger@westminster.ac.uk
Membership Secretary	John Cooke	D.J.Cooke@lboro.ac.uk
Newsletter Editor	Ann Wrightson	scomaw@zeus.hud.ac.uk
Liaison with BCS	Margaret West	m.m.west@hud.ac.uk
Liaison with FME	Tim Denvir	timdenvir@cix.compulink.co.uk