FACS FACTS

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

Series III Vol. 1, No. 2 Spring 1993

Contents

Editorial

FME and FACS FACTS

Formal Methods Europe (FME) is an organisation whose objectives are to stimulate the use of formal methods by European industry and to provide a forum for the formal methods community, especially industrial users and providers. It is supported by DG XIII of the European Commission. FME holds industrial seminars throughout the European Community and holds a major conference every eighteen months. The 1993 conference is in Odense, Denmark on 19-23 April.

FME was also endeavouring to produce a newsletter but, like FACS, has had difficulty in mustering the necessary human resources to do so. These newsletters require a lot of voluntary editorial effort to produce. The result has been a more spasmodic publication than your committee would wish for the FACS newsletter and no publication at all by FME, although their first issue was 90% complete (heard that phrase before?).

The objectives of the two newsletters are almost indistinguishable. Given the limited resources which both FACS and FME have, your committee have decided to collaborate with FME and produce a combined FACS-FME newsletter, starting with the next issue. Combining the two will avoid duplication of a scarce effort and, we hope, provide an opportunity for an improvement in quality and scope. FACS news will be broadcast to a wider audience and we expect contributions from a greater variety of sources from other European countries.

This move is also, we feel, in accord with today's climate of European cooperation. So the next issue will, if all goes to plan, be the first 'Newsletter of FACS and FME'. If anyone is good at dreaming up a new name for it, please let us know!

Z User meeting

The Z user group has had a successful annual meeting for some time now and it has been agreed with them to hold the next Z user meeting as a FACS event. This will take place on 29-30 June 1994. A call for papers and other details will be forthcoming.

Tim Denvir

Correspondence Column

As we said last issue, we will start a column for correspondence ··· just as soon as we get some! We hope this will become a general forum for technical questions and queries, answers and replies, and of course just comments of a general nature. We look forward to hearing from you!

If you want to send a letter or email the FACS newsletter, write to Ann Wrightson at:

Department of Computing,
University of Central Lancashire,
Preston, Lancs., PR1 2HE

or preferably by e-mail to `annw@uk.ac.uclan.sc`.

Mailshots and Flyers

As a matter of policy, the FACS Committee has decided that commercial flyers and mailings may now be included with the Newsletter, provided that they are of general benefit to FACS members. It was agreed that flyers advertising FACS-linked events (with discounts for FACS members) would incur no charge. In all other cases, this service is offered for a small charge to be agreed in each case.

Adverts in the Newsletter have been specifically barred for the time being.

If you are interested in using this service, please contact the Editors for further information (see back cover for details).

Acknowledgements

This edition of the newsletter was produced by:

Brian Monahan	University of Manchester
Jawed Siddiqi	Sheffield Hallam University
Chris Roast	Sheffield Hallam University

Duplication and distribution by the Department of Computing and Management Sciences, Sheffield Hallam University.

BCS FACS MEMBERSHIP REMINDER

Many thanks to all our readers who have already renewed their FACS membership for 1993. To those of you who have not yet done so, please fill in the forms, write out the cheque and send to Loughborough ASAP. Details of fees can be found on the back cover of this newsletter.

Now that the newsletter has been revamped, it seems a pity to include 'tear out' forms within its pages so, if you have mislaid the forms, please email or phone for replacements.

Currently we are liaising with EATCS with the aim of being to update their records directly. When this is done we should be able to lessen the paperwork next year. Indeed we seem to be moving closer to performing the role of an EATCS chapter.

We are also collecting a large number of email addresses. When a critical mass has been achieved we shall supplement our usual direct mailings with broadcast email notices.

Please help us to maintain and improve the service we offer to the Formal Methods community and especially to yourself.

Best wishes,

John Cooke
FACS Membership Secretary

Address information

FACS Membership Secretary
Dr John Cooke: Dept. Computer Studies
Loughborough University of Technology
Loughborough, Leics LE11 3TU
United Kingdom

Tel: +44 509 222676
Fax: +44 509 211586
Email: D.J.Cooke@lut.ac.uk

The Society Column

A good windmill grinds fine but slow. So it is with BCS. Only by waiting till the last minute has your correspondent obtained the following late news items:

Publishing

A publishing seminar was held at Mansfield Mews on 11th March 1993. The official purpose was to debate the strategy for the relaunch of *The Computer Bulletin.* Like many such events it became a free-for-all, but a constructive one nonetheless.

Consensus was that the Bulletin should be remodelled along the lines of *Physics Today.* Another idea was that there might be two bulletins, one aimed at BCS members who are essentially software engineers, the other pitched more at DP managers. This idea has been resisted in the past in order not to fragment the membership. Other professional institutions, IEE for example, have always recognised such internal divisions. Following IEE would set no great precedent.

Would two publications be divisive? Possibly, but two better focused titles would avoid the perennial problem of satisfying neither camp. Market segmentation might also increase sales potential outside the BCS. The final choice rests with BCS and OUP via Itext. Whatever the decision, it will be based on hard commercial considerations. That is something that FACS, a financially successful group, can support wholeheartedly.

Franchising FACS?

Imagine: fast formal methods shops in premiere locations, a major market share in the retail technology transfer sector, a franchised brand name. Has your correspondent lost his marbles? (Actually yes, but that was years ago.) What's it all about?

Over the years FACS has acquired considerable expertise in organising meetings. We are surprised, but delighted, to find that other groups are sufficiently impressed that they would like FACS to organise their meetings. To keep the right side of the BCS SG guidelines, we would have to do this through a separate legal entity, and a meeting with solicitors has been arranged to discuss possible forms of incorporation. Rest assured that your FACS SG membership will not be affected, but a FACS Ltd. probably consituted as an independent charity, is under consideration.

Watch out for McDomains in your high street. FACS never tilts at windmills!

David Blyth
Incord Ltd.

The Society Column

c/o Incord Ltd,
15, Sherwood Avenue
FERNDOWN
Dorset
BH22 8JS

Tel: +44 202 896834
FAX: +44 202 894834
E-mail: dblyth@cix.compulink.co.uk

The 1992 BCS-FACS Christmas Workshop

Turning a Formal Eye on Requirements

was held at the Department of Computing of Imperial College, London, on 16-17th December 1992.

The aim of the workshop was to survey, as far as possible in the confines of a short workshop, the current state of R&D in the requirements engineering stage of the development of software-intensive systems; the elicitation of requirements, how they are expressed and how they are analysed. I'm not sure how far it met that ambitious aim — certainly it provided an opportunity for lots of interesting conversations, and chances to appreciate different approaches to this area of work.

Tom Maibaum, Imperial

Formalization of the Engineering Process

Tom considered what came out of the idea of *doing a requirements engineering job on systems engineering itself*, and provided a number of examples of formally expressed characteristics which follow from identifying desirable features of systems development. Particularly interesting was an analysis of different levels of knowledge and abstraction involved in specifying and implementing a system, mapped into several interrelated planes, with the connections between them expressing explicitly some correspondences which in my experience are taken for granted between different representations of *the system* as it is developed.

Sara Jones, City University

The GMARC project

The GMARC project has proposed an approach to the use of structured domain knowledge in generating formal specifications of system requirements. In this approach, domain knowledge is structured around a network of interrelated application goals, some of which are encapsulated with fragments of formal theory describing behaviour needed to realise those goals. Requirements specifications are developed incrementally by navigating the network to select, modify and compose appropriate theory fragments.

This talk describes the way in which the approach to the process of requirements specification proposed by the GMARC project addresses the need for a bridge across the gap between imprecise and informally expressed requirements and formal specifications.

Anthony Hall, Praxis

Use of Formal Methods in defining requirements for air traffic control

Anthony Hall described how formal methods had been used as a tool for improving the rigour and exactness of analysing requirements for an air traffic control system. The overall framework of the development was expressed using *structured methods* — the formal work using VDM complemented the other methods used rather than supplanting or invalidating them.

Albert Camilleri, HP Research

Theorem proving as an Industrial tool for System-level Design

The motivation for this use of formal methods was reducing time to market by providing extra help to designers of systems (at the architectural level of design, i.e. above register transfer level). HOL was used because already known and accessible. The resulting pattern of activity could be called (perhaps tongue-in-cheek) *proof-driven specification* — the development of proofs gave information to designers on what it was they needed to design. There was some clear benefit in finding obscure bugs, and clarifying difficult concepts such as stalling conditions. Interestingly, a proof also showed that a proposed priority scheme was not needed (that fairness achieved the desired behaviour) — so directly saving design effort.

Axel van Lamsweerde, Louvain University

Requirements Engineering: Products and Processes

The first part of the talk briefly introduced some of the work done in the ICARUS ESPRIT project: design of an incremental language for requirements specification, interval-based layer for specifying real-time constraints, and a model for the specification elaboration process.

The talk then concentrated on the acquisition phase of the requirements engineering lifecycle. During this phase a global model for the specification of the system and its environment is elaborated. This model involves concepts that are currently not supported by existing formal specification languages, such as goals to be achieved, agents to be assigned, alternatives to be negotiated with the client, etc. This approach to requirements acquisition is driven by such higher-level concepts. Requirements models are acquired as instances of a conceptual meta-model. The latter can be seen as a graph where each node captures an abstraction such as, e.g., goal, action, agent, relationship or event, and where edges capture semantic links between such abstractions. Well-formedness properties on nodes and links constrain their instances — that is, elements of requirements models. Requirements acquisition processes then correspond to particular ways of traversing the meta-model graph to acquire appropriate instances of the various nodes and links according to such constraints. Acquisition processes are governed by strategies telling which way to follow systematically in that graph; at each node specific tactics can be applied for acquiring the corresponding instances — and in particular, the formal assertions that can be attached to them. These concepts support a goal-directed acquisition strategy with reuse-based tactics.

Anthony Finkelstein, Imperial

Methodology in Formal Requirements Capture

Anthony Finkelstein's presentation centred on the concept of a method for requirements engineering. A method inherently entails multiple representation schemes, and the nature and degree of consistency desired between them needs careful thought in designing a method. A method in use gains more than content when a notional network of idealized representations (the method as in the textbook) becomes a network of actual specification fragments — which is used by an organization, the development staff. The rationale for the various actions whereby this network grows lead to a common rationalized design history, and since this can have a great value in underpinning a successful development, it becomes an interesting question how to record and browse this historical rationale. The structure of documented requirements should reflect the elicitation structure,i.e. where the information came from, not how it will be used, because traceability back to the originating organization is all-important.

Matthew Bickerton, Oxford Centre for Requirements

Requirements Engineering in the Workplace

Matthew Bickerton's presentation centred on the use of ethnography and ethnomethodology to develop an understanding of workplaces. He further elaborated on how this understanding can be used to inform requirements engineering practitioners to develop more insightful requirements documents. Much lively discussion followed.

British Aerospace Dependable Systems Centre, Newcastle & York

Andy Coombes, University of York, & Rogerio de Lemos, University of Newcastle

The joint Newcastle/York project on dependable systems

This presentation covered work in progress in two areas: a causal approach to requirements modelling, and process control. Tom Anderson's absence was regretted. There was some useful exchange of information on past and present work after the presentations.

Bill Quirk, Harwell

Validation : 3 Aspects of the FOREST project

Bill Quirk spoke about classic accounts of failures and accidents which provide a solid reminder that validation is not a formal but a human process, which sometimes costs human life. Validation as well as verification needs to be taken into account in the requirements engineering enterprise, since with increasing complexity come unexpected and different modes of failure. Good validation depends on wide lateral thinking, which is not a characteristic of formal systems.

The workshop closed with a panel session, which provided a good opportunity for bringing together the matters raised in separate presentations over the two days.

Ann Wrightson
University of Central Lancashire

This report represents a personal perspective upon the events at the workshop. If you have any opinions or comments either upon the views expressed here or concerning requirements aspects in general, we would be most interested to hear from you.
General correspondance is welcomed on this and other topics; this should be sent to the correspondance editor:

Ann Wrightson
Department of Computing
University of Central Lancashire
Preston, Lancashire, PR1 2HE
United Kingdom

Bent Dandanell

FACS FACTS notes with sadness the death of Bent Dandanell the author of our informative article on RAISE tools and first contributor to the RAISE column. We take this opportunity to convey our condolences to his friends and colleagues.

> Those of you who knew Bent Dandanell from CRI will be sorry to hear that he has been killed in a plane crash. Bent was a keen parachutist and was in a light plane that apparently developed engine trouble just after take off.
>
> Bent, only 29 years old, was a brilliant software engineer. He contributed considerably to the RAISE and LaCoS projects as a tool builder and as a consultant to two of the LaCoS partners. His liveliness, good humour and ability to enthuse people will be sorely missed.
>
> Chris George, CRI, Project Manager of LaCoS project.

RAISE Column
Proving false

Chris George, CRI

The RAISE justification editor is the tool we have been concentrating on recently at CRI. It is the only theorem prover I know of that is designed to be unsound. Since it is supposed to support rigorous arguments ("justifications") rather than just formal ones, it must be possible at any point to say "immediate" (which we call an explanation argument) or "this reduces to ..." (which we call a replacement argument). Clearly such arguments can be erroneous. The editor therefore cannot ensure correctness. It does, however, keep a count of how many informal arguments there are. Hence a complete justification with no informal arguments should be sound.

Provided, of course, that the rules it is using are sound and the implementation is correct. We did at one point have a bug in that the editor was not checking applicability conditions for inference rules used in forwards mode (natural deduction style). This made possible the following proof of falsity, and demonstrates the possible paradoxes of empty types. Here is the erroneous proof:

```
⌞false⌟
lemma
[l1]
      ⌞∀ x : {⌈ i : Int • false |} • false⌟
   all_subtype1 :
      ⌞∀ x : Int • false ⇒ false⌟
   simplify :
      ⌞true⌟
   qed
in
   follows
     from
        [l1]  ⌞∀ x : {| i : Int • false |} • false⌟
           ⌞∀ x : {| i : Int • false |} • false⌟
        all_elimination_inf ⇒
           ⌞false⌟
        qed
   end
end
```

The "proof" starts with the goal **false**. Goals appear in half brackets ⌞ ... ⌟. Then a lemma called "l1" is asserted and proved. The lemma states that **false** is **true** for all integers in the empty subtype of integers. This subtype is denoted by the expression

$$\{|\ i : \mathbf{Int} \bullet \mathbf{false}\ |\}$$

i.e. those integers for which false is true. This lemma seems intuitively reasonable, if strange, and is proved in two steps. Firstly the equivalence rule

[all_subtype1]
$\forall\ b : \{|\ b' : T \bullet eb'\ |\} \bullet eb \simeq \forall\ b : T \bullet \text{subst\_binding}(b, b', eb') \Rightarrow eb$
when no_new_capture(b, b′, eb′)

is used. subst_binding(b, b′, eb′) substitutes a new binding b for an old one b′ in the expression eb′. no_new_capture(b, b′, eb′) is the applicability condition. It checks that b does not capture (bind) any free names in eb′ that were not bound by b′. This applicability condition is is decidable, and the justification editor can always check it. It is also possible to have non-decidable (conjuncts in) applicability conditions, in which case the editor will generate a side-condition to be justified.

After applying all_subtype1 the resulting goal is reduced to **true** by the simplifier, a simple automatic rewrite engine. So the lemma has been formally justified.

Now we use the lemma in the main proof[1]. We apply the inference rule

[all_elimination_inf]

$$\frac{\forall\ b : T \bullet eb}{eb}$$

when no_capture(b, eb)

to eliminate the quantification and achieve the goal **false**, which is what we set out to prove. QED. So we have a formal proof of a contradiction.

In fact I have omitted a conjunct from the applicability condition of one of the two rules quoted above. You might like to consider what is missing.

A reminder: I am just the editor of this column. If you have ideas, opinions, experiences, questions, ... then e-mail them to cwg@csd.cri.dk, or mail them to Chris George, CRI A/S, P.O.Box 173, Bregnerødvej 144, DK-3460 Birkerød, Denmark.

[1] It is not necessary to justify lemmas before they are used. Arguments for main goals, lemmas, side conditions etc. can be tackled in any order.

The KAOS Project: Knowledge Acquisition in Automated Specification of Software

Axel van Lamsweerde

Unité d'Informatique, University of Louvain, B-1348 Louvain-la-Neuve (Belgium)
avl@info.ucl.ac.be

1. Introduction

Requirements analysis is a highly critical step in the software lifecycle. Errors during this step may have disastrous effects on the subsequent development steps and on the quality of the resulting product. Therefore, it is essential that requirements engineering be done with great care and precision. Recently, researchers have devoted considerable effort to the design of formal specification languages. Examples are state-based languages such as Z or VDM, history-based languages such as ERAE or INFOLOG, transition-based languages such as STATECHARTS, and algebraic languages such as LARCH, ASL or PLUSS. Once it is formalized a specification can be checked against a set of desired properties, can be used to generate a prototype implementation, and so forth. In using such languages, requirements engineers face two difficulties - the *limited scope* of the language (only those functional aspects which can be captured in terms of data types and operations are specified), and the preliminary *acquisition* of relevant requirements (to formalize the requirements, one must first know what these requirements are).

In this context, we view requirements analysis as being made of two co-ordinated tasks, requirements acquisition and formal specification.

• In *requirements acquisition* a preliminary model for the specification of the system and its environment is elaborated and expressed in a "rich" representation language. This language needs a variety of built-in concepts to structure requirements about the composite system in terms of the kind of abstractions usually found in requirements documents, such as objectives and constraints to be met by the composite system, entities, relationships, events and actions taking place in it, agents controlling the actions, responsibilities assigned, possible scenarios of system behaviour, and so forth. The language should also provide facilities for capturing multiple automation alternatives in a form amenable to evaluation and negotiation between the analysts and the clients.

• In *formal specification* a specific automation alternative that emerged during acquisition is considered, and the part to be automated in the corresponding composite system is retained; the preliminary specification obtained for the data and operations of that sub-system is refined and made more precise using a formalism suitable for detailed formal proofs and prototype generation.

Our interest is in modelling and formalizing the processes involved in both tasks through two complementary projects: KAOS (for requirements acquisition) and ICARUS (for formal specification). This overview concentrates on the former project.

The driving forces of the KAOS approach are the reuse of domain knowledge and the application of machine learning technology [Vla91a]. Two learning strategies have been adapted to the context of requirements acquisition: *learning-by-instruction*, where the learner conducts the acquisition process by use of meta-knowledge about the kind of knowledge to be acquired [Dav82] [Ben85], and *learning-by-analogy*, where the learner retrieves knowledge about some "similar" system to map it to the system being learned [Hal89].

2. General Approach

The overall approach taken in KAOS has three components: (i) a conceptual model for acquiring and structuring requirements models, with an associated acquisition language, (ii) a set of strategies for elaborating requirements models in this framework, and (iii) an automated assistant to provide guidance in the acquisition process according to such strategies.

(i) The *conceptual model* provides a number of abstractions in terms of which requirements models have to be acquired; it is thus a meta-model. Using this model one can capture objectives of the system under consideration, constraints that make such objectives operational, agents like human beings or programs that control the system's behaviour according to such constraints, events that cause the application of actions on entities, and so forth. Also, other structuring link types are supported beside subclass specialization, like (alternative) refinement links between objectives or between constraints, (alternative) assignment links between agents and constraints, and so forth. The meta-model for requirements acquisition can be represented as a conceptual graph where nodes represent abstractions and edges represent structuring links. (Figure 1 illustrates a portion of this graph.)

(ii) An *acquisition strategy* in this framework defines a well-justified composition of steps for acquiring components of the requirements model as instances of meta-model components. In other words, a strategy corresponds to a specifc way of traversing the meta-model graph to acquire instances of its various nodes and links. For example, the meta-model can be traversed backwards from the objectives to be fulfilled by the composite system, or backwards from the agents available in the system and their respective views, or backwards from client-supplied scenarios for combining actions. Each step in a strategy is itself composed from finer steps like, for example, question-answering, input validation against known properties of meta-model components, application of tactics to select among alternatives, deductive inferencing based on property inheritance through specialization links, analogical inferencing based on knowledge about similar systems, or conflict resolution between multiple views of human agents involved [Rob89].

(iii) The *acquisition assistant* is aimed at providing automated support in following one acquisition strategy or another. It is built around two repositories: a requirements database and a requirements knowledge base. Both are structured according to the meta-model components. The *requirements database* maintains the requirements model built gradually during acquisition; the latter can be analyzed using query facilities similar to those provided by project database systems [Vla88]. The *requirements knowledge base* contains two kinds of knowledge. *Domain-level* knowledge concerns concepts and requirements typically found in the application domain considered. As in [Reu91], this knowledge is organised into specialization hierarchies; requirements fragments for a particular class of systems known to the assistant (e.g., library management, airline reservation, telephone network) are thereby inherited from more general applications (e.g., resource management, transportation, communication) and from more general tasks (e.g., transaction processing, history tracking, device control). Besides, *meta-level* knowledge concerns properties of the abstractions found in the meta-model (e.g., "a constraint that can be temporarily violated needs to be restored by some appropriate action") and ways of conducting specific acquisition strategies. The latter aspect includes tactics that can be used within strategies (e.g., "prefer those alternative refinements of objectives which split responsibility among fewer agents ").

3. A Conceptual Meta-model for Requirements Capture

A distinction is made between three levels of modelling. The *meta* level refers to domain-indepen-

dent abstractions (e.g., the "Goal" or "Agent" meta-concepts); the *domain* level refers to concepts specific to the application domain (e.g., the "GetBookRequestSatisfied" and "Borrower" concepts); the *instance* level refers to specific instances of domain-level concepts. The KAOS meta-model is a model for the meta level; it is made of *concepts*, *relationships* linking these concepts, *attributes* characterizing these concepts and relationships, and *constraints* over these concepts and relationships (e.g., cardinality constraints). Such constraints must be satisfied in the final state of the acquisition process.

A significant portion of the KAOS meta-model is depicted in Figure 1. For a precise definition of its various components, see [Vla91b], [Dar93].

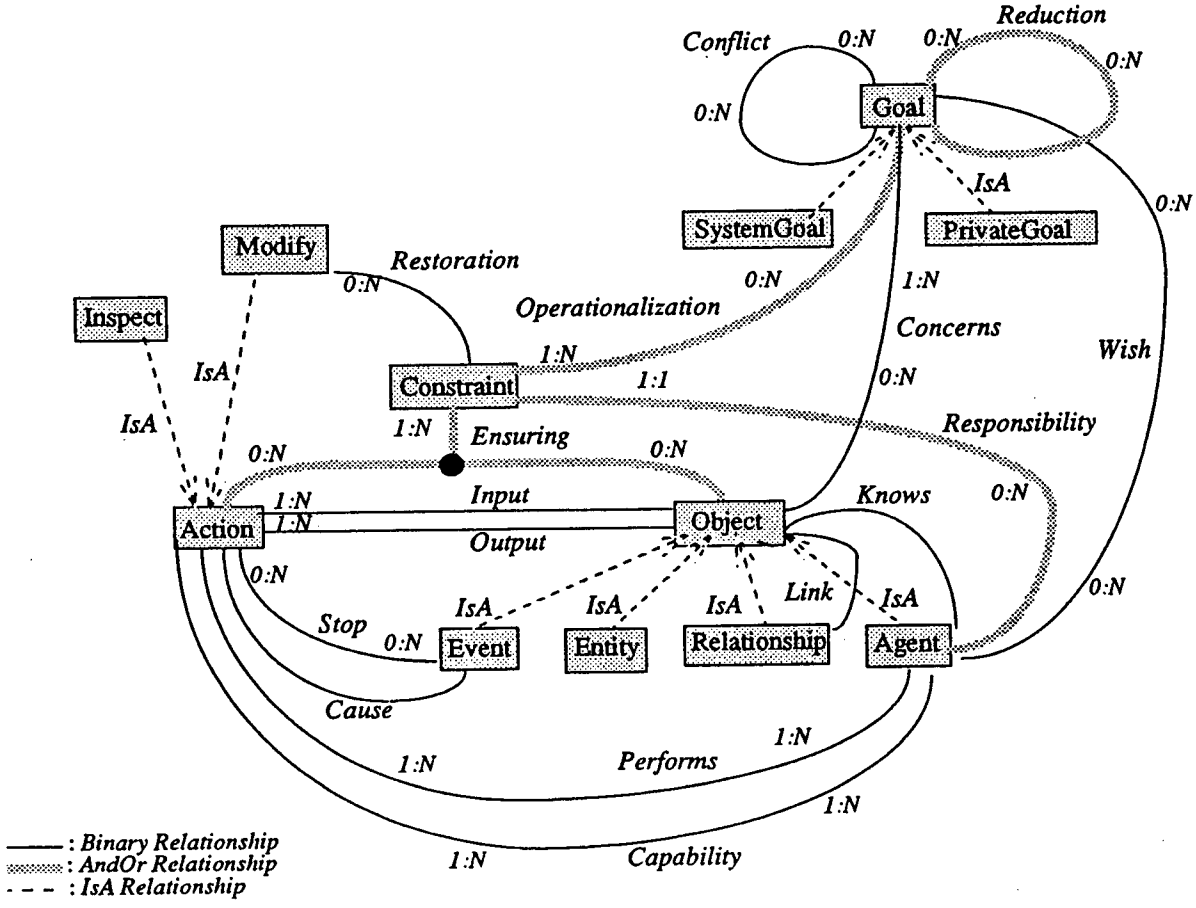

Figure 1: A Portion of the KAOS Conceptual Meta-model

The role of the meta-model can be made more explicit as follows.

• Requirements models are acquired as instances of the meta-concepts, linked by instances of the meta-relationships and characterized by instances of the meta-attributes. In particular, *formal assertions* can be attached as values for attributes inherited from the meta level (like, e.g., the "FormalDef" attribute of the "Goal" meta-concept, the "Invariant" attribute of the "Object" meta-concept, the "TriggerCondition" of the "Action" meta-concept, etc.). The assertion language used for writing such attribute values is a typed temporal first-order logic equipped with real-time temporal constructs [Dub91].

• Reusable domain knowledge fragments are expressed similarly.

• The meta-model drives the acquisition process as in learning-by-instruction systems. For example, with the goal-directed strategy summarized in Section 4, the *Goal* meta-concept is the first to be considered; instances of it are acquired through *Reduction* and *IsA* specialization links (see Figure 1); the objects *Concerned* by the goals acquired are also preliminarily defined. Then the *Agent* and *Capability* meta-level abstractions are considered to identify relevant instances of them. Next the *Constraint* meta-concept is considered; instances of it are acquired from goals through *Operationalization* links, and so forth. It is important to recognize that *the more domain-independent knowledge is attached to meta-level components, the more knowledge-based guidance can be provided in the acquisition process.*

• The meta-model components yield criteria for measuring conceptual similarity when a learning-by-analogy strategy is followed [Dbi92]. Analog requirements fragments are retrieved in the domain knowledge base and mapped to the target requirements from similarities that are evaluated between goals, constraints, actions, events and the like.

4. A Goal-Directed Acquisition Strategy

Strategies differ by the meta-model concept(s) around which they are centered; goal-directed, agent-directed and scenario-directed strategies have been identified. The strategy we have favoured so far is a goal-directed one [Dar91]. It is made of the following steps. (Upper case letters are used to refer to meta-level concepts.)

1) Acquire the Goal Reduction/Conflict structure and identify the Concerned Objects.
2) Identify potential Agents and their Capabilities.
3) Operationalize Goals into Constraints.
4) Refine Objects and Actions.
5) Derive strengthened Actions and Objects to Ensure Constraints.
6) Identify alternative Responsibiliies.
7) Assign Actions to responsible Agents.

(These steps are ordered but some of them may overlap; backtracking is possible at every step.) Each step includes the acquisition of temporal logic assertions that must characterize the corresponding domain-level concepts in an appropriate fashion. Specific *tactics* are provided at each step to help in the acquisition process. Two kinds of tactics have been defined. *Heuristic rules* may be used to select a preferred alternative among several identified ones. *Reuse tactics* may be used to specialize generic descriptions found in the domain knowledge base for the corresponding step. See [Dar93] for more details.

5. Requirements Acquisition by Analogy

Human analysts tend to reuse past experience and expertise about "similar" systems as much as they can. Analogical reasoning is thus a powerful paradigm for requirements acquisition. This has been recognized by others as well - see, e.g., [Mai92]. The general idea is to support the transfer, to the *target* analogous situation, of relevant requirements patterns that were introduced in previously specified *source* analogous situations. This raises several problems. Relevant source analogs must be *retrieved* in the domain knowledge base; appropriate features of the source analog must be selected for *transfer*, transferred features sometimes must be *adapted*; the acquired target description sometimes should be *consolidated* for later reuse [Hal89].

In the context of KAOS, two alternatives have been explored. In the first technique, source analogs are suggested by the analyst; the Assistant checks the analogy for adequacy and proposes associ-

ated meta-model instances for adaptation. In the second technique, source analogs are automatically inferred by the Assistant. They are retrieved in the domain knowledge base using similarity functions we have adapted from [Tve77]. For two candidate analogs, such functions measure those instances of meta-concepts, meta-relationships and meta-attributes which are pairwise similar and those which aren't; it is possible to assign different weights to the various components of the conceptual meta-model so as to give stronger credit to some specific ones. Heuristic rules may then be applied to select preferred features for transfer and to exclude others. Our recent work in this area has been focussed on the analogical acquisition of the formal assertions attached to domain-level concepts. See [Dbi92] for more details.

Acknowledgment

The KAOS project is a team effort that involved Anne Dardenne, Bruno Delcourt and Françoise Dubisy. Special thanks are due to Steve Fickas and his group at the University of Oregon for a lot of inspiring discussions. Our work was supported by the Belgian Ministry of Scientific Affairs in the context of the Belgian Incentive Program for Fundamental Research in Artificial Intelligence under grant No. RFO/AI/13.

References

[Ben85] J. Benett, "A Knowledge-Based System for Acquiring the Conceptual Structure of a Diagnostic Expert System", *Journal of Automated Reasoning*, Vol.1, 1985, 49-74.

[Dar91] A. Dardenne, S. Fickas S. and A. van Lamsweerde, "Goal-directed Concept Acquisition in Requirements Elicitation", *Proceedings 6th International Workshop on Software Specification and Design*, Como, Italy, 1991, 14-21.

[Dar93] A. Dardenne, A. van Lamsweerde and S. Fickas, "Goal-directed Requirements Acquisition", to appear in *Science of Computer Programming*, 1993.

[Dav82] R. Davis, "Teiresias: Applications of Meta-Level Knowledge", in: *Knowledge-Based Systems in Artificial Intelligence*, R. Davis and D. Lenat (eds.), McGraw-Hill, 1982, 227-490.

[Dbi92] F. Dubisy and A. van Lamsweerde, *Requirements Acquisition by Analogy*, Report No. 13, KAOS Project, Institut d'Informatique, Facultés Universitaires de Namur, Belgium, 1992.

[Dub91] E. Dubois, J. Hagelstein and A. Rifaut, "A Formal Language for the Requirements Engineering of Computer Systems", in: *Introducing a Logic Based Approach to Artificial Intelligence*, A. Thayse (Ed.), Vol. 3, Wiley, 1991, 357-433.

[Fic88] S. Fickas and P. Nagarajan, "Critiquing Software Specifications", *IEEE Software*, November 1988, 37-46.

[Hal89] R. Hall, "Computational Approaches to Analogical Reasoning: A Comparative Analysis", *Artificial Intelligence*, vol. 39, 1989, 39-120.

[Mai92] N. Maiden and A. Sutcliffe, "Exploiting Reusable Specifications through Analogy", *Communications of the ACM* Vol. 35 No. 4, April 1992, 55-64.

[Reu91] H.B. Reubenstein and R.C. Waters, "The Requirements Apprentice: Automated Assistance for Requirements Acquisition", *IEEE Transactions on Software Engineering*, vol. 17 no. 3, March 1991, 226-240.

[Rob89] W.N. Robinson, "Integrating Multiple Specifications Using Domain Goals", *Proceedings 5th International Workshop on Software Specification and Design*, May 1989, 219-226.

[Tve77] A. Tversky, "Features of Similarity", *Psychological Review*, Vol. 84 No. 4, July 1977, 327-352.

[Vla88] A. van Lamsweerde, B. Delcourt, E. Delor, M.C. Schayes and R. Champagne, "Generic Lifecycle Support in the ALMA Environment", *IEEE Transactions on Software Engineering*, Vol. 14, No. 6, June 1988, 720-741.

[Vla91a] A. van Lamsweerde, "Learning Machine Learning", in: *Introducing a Logic Based Approach to Artificial Intelligence*, A. Thayse (Ed.), Vol. 3, Wiley, 1991, 263-356.

[Vla91b] A. van Lamsweerde, A. Dardenne and F. Dubisy, *KAOS Knowledge Representations as Initial Support for Formal Specification Processes*, Report RR-91-8, Unite d'Informatique, University of Louvain, 1991.

A SURVEY OF FORMAL METHODS

The use of formal methods has been much heralded as the way forward for the production of good quality software but this view has not been widely accepted in industry. The National Physical Laboratory was asked by the Department of Trade and Industry to investigate why formal methods have not been more widely taken up by industry. It did this by conducting surveys of literature, academic institutions and industry in the summer of 1992.

A report was produced on the literature survey. This report lists the perceived benefits, barriers and limitations to formal methods. For all items in the list it discusses by rational argument and reported experimental evidence their validity. This report then summarises what we concluded from this.

The survey of academic institutions was aimed at finding out what formal methods were being taught in them. A brief report summarises this survey.

The aims of the survey of industry were to collect information from people using or considering using formal methods as to the:

- benefits, barriers and limitations to formal methods;
- means of assessing the contribution of formal methods to the software life cycle.

The survey also gives information on what formal methods are being used, how they are being used and what other methodologies they are being used with. We have produced two reports on this — a detailed one and an overview.

It is hoped that an article concerning this survey will appear in the next issue of FACS FACTS.

For further information, please contact:

DITC Office,
Formal Methods Survey,
National Physical Laboratory,
Teddington, Middlesex, TW11 0LW.
Tel: 081–943 7002
Fax: 081–977 7091

EATCS

HISTORY AND ORGANIZATION

EATCS is an international organization founded in 1972. Its aim is to facilitate the exchange of ideas and results among theoretical computer scientists as well as to stimulate cooperation between the theoretical and the practical community in computer science.

Its activities are coordinated by the Council of EATCS, out of which a President, a Vice President, a Treasurer and a Secretary are elected. Policy guidelines are determined by the Council and the General Assembly of EATCS. This assembly is scheduled to take place during the annual International Colloquium on Automata, Languages and Programming (ICALP), the conference of EATCS.

MAJOR ACTIVITIES OF EATCS

- Organization of ICALP's
- Publication of the "Bulletin of the EATCS"
- Publication of the "EATCS Monographs in Theoretical Computer Science"
- Publication of the journal "Theoretical Computer Science"
- Other activities of EATCS include the sponsorship or the cooperation in the organization of various more specialized meetings in theoretical computer science. Among such meetings are: CAAP (Colloquium on Trees in Algebra and Programming), TAPSOFT (Conference on Theory and Practice of Software Development), STACS (Symposium on Theoretical Aspects of Computer Science), LICS (Logic in Computer Science), Conference on Structure in Complexity Theory, SPAA (Symposium on Parallel Algorithms and Architectures), Workshop on Graph Theoretic Concepts in Computer Science, International Conference on Application and Theory of Petri Nets, International Conference on Database Theory, Workshop on Graph Grammars and their Applications in Computer Science.

BENEFITS

Benefits offered by EATCS include:
- Receiving the "Bulletin of the EATCS" (over 1200 pages per year)
- Reduced registration fees at various conferences
- Reciprocity agreements with other organizations
- 25% discount in purchasing ICALP proceedings
- 25% discount in purchasing books from "EATCS Monographs on Theoretical Computer Science"
- About 70 % discount (equals about US $800) per annual subscription to "Theoretical Computer Science".
- About 70 % discount (equals about US $200) per individual annual subscription to "Fundamenta Informaticae".

(1) THE ICALP CONFERENCE

ICALP is an international conference covering all aspects of theoretical computer science and now customarily taking place during the second or third week of July.

Typical topics discussed during recent ICALP conferences are: computability, automata theory, formal language theory, analysis of algorithms, computational complexity, mathematical aspects of programming language definition, logic and semantics of programming languages, foundations of logic programming, theorem proving, software specification, computational geometry, data types and data

structures, theory of data bases and knowledge based systems, cryptography, VLSI structures, parallel and distributed computing, models of concurrency and robotics.
Sites of ICALP meetings:

- Paris, France 1972
- Saarbrücken, Germany 1974
- Edinburgh, Great Britain 1976
- Turku, Finland 1977
- Udine, Italy 1978
- Graz, Austria 1979
- Noordwijkerhout, The Netherlands 1980
- Haifa, Israel 1981
- Aarhus, Denmark 1982
- Barcelona, Spain 1983
- Antwerp, Belgium 1984
- Nafplion, Greece 1985
- Rennes, France 1986
- Karlsruhe, Germany 1987
- Tampere, Finland 1988
- Stresa, Italy 1989
- Warwick, Great Britain 1990
- Madrid, Spain 1991
- Wien, Austria 1992
- Lund, Sweden 1993

(2) THE BULLETIN OF THE EATCS

Three issues of the Bulletin are published annually appearing in February, June and October respectively.

The Bulletin is a medium for *rapid* publication and wide distribution of material such as:

- EATCS matters
- Information about the current ICALP
- Technical contributions
- Surveys and tutorials
- Reports on conferences
- Calendar of events
- Reports on computer science departments and institutes
- Listings of technical reports and publications
- Book reviews
- Open problems and solutions
- Abstracts of Ph.D.Theses
- Information on visitors at various institutions
- Entertaining contributions and pictures related to computer science.

Contributions to any of the above areas are solicited. All written contributions should be sent to the Bulletin Editor:

Prof.Dr. G. Rozenberg
Dept. of Mathematics and Computer Science
University of Leiden
P.O. Box 9512
2300 RA Leiden, The Netherlands
Email: rozenber@rulcri.leidenuniv.nl
Fax: +31-71-276985

Deadlines for submissions to reach the Bulletin Editor are: January 15, May 15 and September 15 for the February, June and October issue respectively.

All pictures (preferably black and white) including text of what they are showing should be sent to the Picture Editor:

Prof.Dr. M. Kudlek
FB Informatik, Universität Hamburg
Schlüterstr. 70
2000 Hamburg 13, Germany

Deadlines are 2 weeks *before* those for written contributions, indicated above.

(3) EATCS MONOGRAPHS ON THEORETICAL COMPUTER SCIENCE

This is a series of monographs published by Springer-Verlag and launched during ICALP 1984; since then 15 volumes were published. The series includes monographs as well as innovative textbooks in all areas of theoretical computer science, such as the areas listed above in connection with the ICALP conference. The volumes are hard-cover and ordinarily produced by type-setting. To ensure attractive prices other production methods are possible.

The editors of the series are W. Brauer (Munich), G. Rozenberg (Leiden), and A. Salomaa (Turku). Potential authors should contact one of the editors.

The advisory board consists of G. Ausiello (Rome), M. Broy (Passau), S. Even (Haifa), J. Hartmanis (Ithaca), N. Jones (Copenhagen), T. Leighton (Cambridge, Mass.), M. Nivat (Paris), C. Papadimitriou (Athens and San Diego), and D. Scott (Pittsburgh).

Updated information about the series can be obtained from the publisher, Springer-Verlag.

(4) THEORETICAL COMPUTER SCIENCE

The journal *Theoretical Computer Science*, founded in 1975, is published by Elsevier Science Publishers, Amsterdam, currently in 16 volumes (32 issues) a year.

Its contents are mathematical and abstract in spirit, but it derives its motivation from practical and everyday computation. Its aim is to understand the nature of computation and, as a consequence of this understanding, provide more efficient methodologies.

All kinds of papers, introducing or studying mathematical, logical and formal concepts and methods are welcome, provided that their motivation is clearly drawn from the field of computing.

Papers published in *TCS* are grouped in two sections according to their nature. One section, "Algorithms, automata, complexity and games", is devoted to the study of algorithms and their complexity using analytical, combinatorial or probabilistic methods. It includes the whole fields of abstract complexity (i.e., all the results about the hierarchies that can be defined using Turing machines), of automata and language theory (including automata on infinite words and infinitary languages), of geometrical (graphic) applications and of system performance using statistical models.

A subsection is the Mathematical Games Section, which is devoted to the mathematical and computational analysis of games.

The other section, "Logic, semantics and theory of programming", is devoted to formal methods to check properties of programs of implement formally described languages; it contains all papers dealing with semantics of sequential and parallel programming languages. All formal methods treating these problems are published in this section, including rewriting techniques, abstract data types, automatic theorem proving, calculi such as SCP or CCS, Petri nets, new logic calculi and developments in categorical methods.

The editor-in-chief of "Theoretical Computer Science" is:
Prof.M. Nivat
26 rue du Poitou
94550 Chevilly-Larue
France

ADDITIONAL INFORMATION

Please contact the Secretary of EATCS:
Prof.Dr.B. Monien
Fachbereich Mathematik - Informatik
Universität-GH Paderborn
4790 Paderborn, Germany
Email: eatcs@uni-paderborn.de

Faculty of Information Technology

University of Brighton

Department of Computing
Professor Dan Simpson
Head of Department

Watts Building
Moulsecoomb
Brighton BN2 4GJ England
Telephone 0273 600900
Facsimile 0273 642405

Direct line 0273 642451

Date Wednesday, February 10, 1993

Dear Colleague

EATCS Fund

Over the past few years Dines Bjørner has coordinated this scheme which allows colleagues in countries with hard currencies to help our computer science colleagues in currency constrained countries. This scheme has been very successful, currently it allows about 100 individuals in currency constrained countries to be members of EATCS and enjoy all the benefits of such membership. As many of you will know Dines has taken up a new post in Macao and I am honoured to accept the job of continuing this scheme.
The former Eastern European countries are undergoing tremendous changes which are causing enormous financial difficulties in these countries. As usual such difficulties are being felt particularly in sections of society such as education and science. This is why your help to our TCS colleagues in this region is now more important than ever. Such help is also appreciated more than ever.

The scheme operates as follows. Members in countries with hard currencies are asked to make a donation to provide membership for those in currency controlled countries. To do this is very easy. You simply send me, at the above address, a cheque made out to "*EATCS: European Association for Theoretical Computer Science*". The value of the cheque should be for n annual subscriptions where n is a number of years. The current annual subscription is US $18, 30DM or equivalent in any hard currency. Of course it helps with continuity if you can make n as large as possible. The EATCS bank in Antwerp will cash these cheques in a very favourable way. If you wish to sponsor more than one person simply send integer multiples of your donation. We keep a pool of names of people who we wish to support, your donations allow the choice from this pool to be made. If you are aware of someone who you think is particularly worth supporting please send me details so that we can consider them for adding to the pool of names.

Over the past years many colleagues have helped other computer scientists in this way. Needless to say the donations are very helpful as they allow people to function in a less constrained way. In particular recipients receive the Bulletin, make new contacts, attend conferences and generally benefit from being full members of the community. To those who have been donors in the past I urge you to continue to do so, to others I suggest that this is a most worthwhile way of helping others.
I look forward to hearing from you and, particularly, to being able to pass on your cheques to this good cause.

Best wishes

Dan Simpson

FORTHCOMING EVENTS

1993

The following is a list of forthcoming events which broadly lie in a period of 8 months. Earlier events may be found in the previous issue.

June 21–3 July
Fifth International School for Computer Science Researchers "Specification and Validation Methods for Programming Languages and Systems",
Lipari Island, Italy. *Contact:* Prof. A. Ferro, Dipartimento di Matematica, Citta Universitaria, Viale A. Doria, 6, 95125 Catania, Italy. Email school@mathct.cineca.it

June 22
Computer Supported Cooperative Work, Petri Nets and Related Formalisms,
Chicago, USA *Contact:* G. De Michells, University of Millano, Via Comelico 39, 20135 Milano, Italy. Email gdemich@hermes.dsi.unimi.it. **Or** C. Ellis, University of Colorado at Boulder, Boulder 80309, USA. Email skip@cs.colorado.edu

June 22
ACM SIGPLAN'93 Workshop on State in Programming Languages,
Albuquerque, New Mexico, USA. *Contact:* Prof. P. Hudak, State Workshop'93, Department of Computer Science, Yale University, 51 Prospect Street, New Haven, CT 06520-2158, USA. Email state-workshop@cs.yale.edu

June 28 – July 1
Fifth Conference on Computer-Aided Verification,
Heraklio, Crete, Greece. *Contact:* Costas Courcoubetia, University of Crete, Department of Computer Science and Institute of Computer Science, FORTH, PO BOX 1385, GR-7110, Heraklion, Crete, Greece. Email courcou@csi.forth.gr

June 28 – July 3
International Conference on Formal Methods in Programming and Their Applications,
Novosibirsk, Russia. *Contact:* M. Bulyonkov, Institute of Informatics Systems, 6, Acad. Lavrentyeva av., 630090 Novosibirsk, Russia. Email mike@isi.itfs.nsk.su

June 28 – July 9
First International Summer School in Logic for Computer Science,
Le Bourget-du-Lac, France. *Contact:* M. Parigot, School LCS, Laboratoire de Logique, UFR de Mathematiques, Universite Paris 7, 2 place Jussieu, 75251 Paris Cedex 05, France. Email school@logique, jussieu.fr

June 29 – July 2
International Conference on Number Theoretic and Algebraic Methods in Computer Science,
Moscow, Russia. *Contact:* I. Shparlinski, Dept. no. 4, ICSTI, Kuusinena str., 21 B, Moscow, 125252, Russia, fax +95-9430089. Email shparplb.icsti.su

July 5–9 ICALP'93
20th International Colloquium on
Automata, Languages, and Programming,
Lund, Sweden. *Contact:* Prof. Rolf Karlsson, Department of Computer Science, Lund University, S-221 00 Lund, Sweden. Email: icalp93@dna.lth.se

July 7–9 SEE'93
Conference on Software Engineering Environments,
Reading, United Kingdom. *Contact:* Ray Welland, Comp. Sci. Dept., Univ. of Glasgow, Glasgow, G12 8QQ, UK, tel +44 41 330 4968, fax +44 41 330 4913. Email ray@dcs.glasgow.ac.uk

July 12–15
Developments in Language Theory,
Turku, Finland. *Contact:* A. Salomaa, Mathematics Department, University of Turku, SF-20700 Turku, Finland, fax +358-21-6336595

July 13–20 LPAR'93
Fourth International Conference on Logic Programming and Automated Reasoning,
St. Petersburg, Russia. *Contact:* LPAR'93 ECRC, Arabelleastrasse 17, 8000 Munich 81, Germany.

July 19–23 CASE 93
Sixth International Workshop Computer-Assisted Software Eng.,
Singapore. *Contact:* John Junkins, City Univ. London, School of Informatics, Northampton Sq., London, EC1 0HB, UK, phone 44 (71) 477-8410, fax 44 (71) 477-8588. Email sb308@city.ac.uk

July 20–August 1
Marktoberdorf Summer School "Proof and Computation",
Marktoberdorf, Germany. *Contact:* Faculty of Computer Science, Technical University of Munich, Summer School, PO BOX 202420, W-8000 Munich 2, Germany.

July 26–30 ECOOP'93
Seventh European Conference on Object-Oriented Programming,
Kaiserslautern, Germany. *Contact:* W. Olthoff, ECOOP'93, Organizing Chair, German Research Center for AI, PF 2080, 6750 Kaiserslautern, Germany, fax +49-631-2053210. Email ECOOP93@dfkl.uni-ki.de

July 26–30
7th European Conference on Object Oriented Programming,
Kaiserslautern, Germany. *Sponsor:* DFKI in coop. w/SIGPLAN. *Contact:* Gerhard Barth, DFKI, PO BOX 2080 Kaiserslautern 6750 Germany, tel +49 631 205 3213. Email barth@dfki.uni-kl.de

August 13–24
Nato Advanced Study Institute, Constraint Programming,
Pamu, Estonia *Contact:* J. Penjam, Institute of Cybernetics, 21 Akadeemia tee, EE-0026 Tallinn, Estonia. Email natoasi@ioc.ee

August 15–18
12th Annual ACM Symposium on Principles of Distributed Computing,
Ithaca, N.Y., USA. *Sponsor:* SIGOPS, SIGACT. *Contact:* James Anderson, Dept. of Comp. Sci., Univ. of Maryland-College Park, A.V. Williams Building, College Park, MD 20742-3255, tel (301) 405-2701. Email 1ha@cs.umd.edu

August 23–26 CONCUR'93
Fourth International Conference on Concurrency Theory,
Hildesheim, Germany. *Contact:* CONCUR'93, attn. E. Best, Institut fur Informatik, Universitat Hildesheim, Marlenburger Platz 22, D-3200 Hildesheim, Germany. Email E.Best@informatik.uni-hildeshiem.de

August 23–27 FCT'93
Fundamentals of Computation Theory,
Szeged, Hungary. *Contact:* T. Gaizer or J. Viragh, FCT'93 Bloyai Institute, A. Jozsef University, 6721 Szeged, Aradi v. tere 1., Hungary, fax 36-62-12292. Email: h754esi@ella.hu, h1299gai@ella.hu, J68A004@HUSZEG11

August 25–27 PLILP'93
Fifth International Symposium on Programming Language Implementation and Logic Programming,
Tallinn, Estonia. *Contact:* PLILP 93, Jaan Penjam, Software Department, Institute of Cybernetics of Estonian Academy of Sciences, Akadeemia tee 21, EE-0108 Tallinn, Estonia. Email plilp@ioc.ee

August 30–September 3 MFCS'93
Eighteenth International Symposium on Mathematical Foundations of Computer Science,
Gdansk, Poland. *Contact:* MFCS'93, Institute of Computer Science of the Polish Academy of Sciences, ul. Jaskowa Dolina 31, PO BOX 562, 80-252 Gdansk, Poland, fax +48-58-416912

September 1–3 The Mathematics of Dependable Systems,
London, UK. *Sponsor:* The Institute of Mathematics and its Applications *Contact:* Dr Victoria Stavridou, Department of Computer Science, Royal Holloway and Bedford New College, University of London, Egham, Egham Hill, Surrey TW20 0EX, UK, tel (+44) 784 443429/21, fax (+44) 784 443420. Email victoria@uk.ac.rhbnc.cs

September 7–10
Category Theory and Computer Science,
Amsterdam, The Netherlands. *Contact:* Dr. D. Pitt, Department of Mathematics, University of Surrey, Guildford, Surrey GU2 5XH, United Kingdom.

September 13–17 ESEC'93
4th European Software Engineering Conference,
Garmisch-Partruchen, Germany. *Sponsor:* AFCET, AICA, ATI, BCS, GI, OGI and SI. *Contact:* ESEC'93, c/o Uta Weber, Technical Univ., Inst. fur Informatik, Orlensstr. 34, D-2000 Munchen 80 Germany, tel +49 48095 142, fax +49 89 48095 160. Email esec@informatik.tu-muenchen.de

September 13–17 CSL'93
Conference of the European Association for Computer Science Logic,
Swansea, Wales. *Sponsor:* EACSL. *Contact:* K. Meinke, Dept. of Comp. Sci., Univ., College of Swansea, Swansea, Great Britain, fax +44 792 295708. Email csl93@pyr.swan.ac.uk

September 23–24 HOA'93
An International Workshop on Higher Order Algebra, Logic and Term Rewriting,
Amsterdam, The Netherlands. *Contact:* Prof. dr. B. Moller (HOA'93), Institut fur Mathematik, Universitat Augsburg, Universitatsstrasse 2, W-8900 Augsburg, Germany, fax +49-821-5982200. Email moeller@uni-augsburg.de

September 26–October 1 OOPSLA'93
Conference on Object Oriented Programming Systems Languages and Applications,
Washington, DC., USA. *Sponsor:* SIGPLAn. *Contact:* Timlynn Babitsky, JFS Consulting, 5 Wise Ferry Ct., Lexington, SC 29072, tel (803) 957-5779.

September 27–30
Conference on Software Maintenance '93,
Montreal, Quebec, Canada. *Contact:* Marc Kellner, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA, tel 412 268 7721, fax 412 268 5758. Email mik@sei.cmu.edu

October 6–8
12th Symposium on Reliable Distributed Systems,
Princeton, New Jersey, USA. *Contact:* Prof. David Taylor, Department of Computer Science, University of Waterloo, Waterloo, Ontario, CANADA N2L 3G1, tel (519) 888-4432. Email dtaylor@grand.uwaterloo.ca

October 19–22 PNPM'93
Fifth International Workshop on Petri Nets and Performance Models,
Toulouse, France. *Contact:* G. Juanole, LAAS-CNRS, 7, Avenue du Colonel Roche, 31077 Toulouse Cedex, France, fax +33-61-336411. Email juanole@laas.fr

October 26–29 ILPS'93
International Logic Programming Symposium,
Vancouver, Canada. *Contact:* Dale Miller, Department of Computer Science, 200 South 33rd Street, University of Pennsylvanis, Philadelphia, PA 19104-6389, USA, fax +1-215-8980587. Email dale@saui.cis.upenn.edu

November 3–6 ISSRE 93
Fourth International Symposium on Software Reliability Engineering,
Denver. *Cosponsor:* IEEE Computer Soc. Technical Committee on Software Eng., IEEE Reliability Soc. Denver Chapter. *Contact:* Anneliese von Mayrhauser, Computer Science Dept., Colorado State Univ., Ft. Collins, CO 80523, phone (303) 491-7016, fax (303) 491-6639. Email avm@cs.colostate.edu. **Or** Yoshihiro Tohma, Computer Science Dept., Tokyo Inst. of Technology, 2-12-1 Oakayama Meguro-ku, Tokyo 152, Japan, phone 81 (3) 3726-1111, ext. 2566. Email tohma@cs.titech.ac.jp

December 15–17 FSTTCS'93
Thirteenth Conference on the Foundations of Software Technology and Theoretical Computer Science,
Bombay, India. *Contact:* Prof. R.K. Shyamasundar, FST&TCS'13, Tata Institute of Fundamental Research, Bombay 400 005, India, fax +91-22-215-2181. Email fsttcs@tifrvax.bitnet

1994

January 5–7
The 6th FACS Refinement Workshop on Theory and Practice of Formal Software Development,
London, United Kingdom. *Contact:* R. Shaw, Lloyd's Register, Lloyd's Register House, 29 Wellesley Road, Croydon, CR0 2AJ, United Kingdom. Email roger.shaw@ale.lreg.co.uk

Guidelines for Newsletter Contributions

Contributions may be in the form of **single-sided** camera-ready copy, suitable for layout and sub-editing. They can also be sent to us using electronic media (i.e. by floppy disk (MS DOS or Mac)/e-mail/etc.), to be formatted in the house style. As a rule, we generally accept pure ASCII text or TEX/LATEX in order to avoid complications involving interchange between wordprocessing formats. We regret that we are unable to offer typesetting facilities for handwritten material.

If contributions are sent using proprietary wordprocessor/markup language formats (i.e. MicroSoft Word 5, FrameMaker), then these will be treated as though they were camera-ready copy. If we are unable to print them adequately or to otherwise convert to another more suitable form then the authors may be asked to provide paper copies of appropriate reproduction quality.

Artwork can be provided for appropriate inclusion, either using general formats (such as DVI files or Encapsulated PostScript) or by sending camera-ready paper copy. Generally, line drawings and other high-contrast graphical diagrams will be acceptable.

Material must be of adequate quality for reproduction. Output from high quality printers with at least 300 DPI resolution is generally acceptable. Output from printers with lesser resolution (i.e. dot-matrix printers) tends not to reproduce very well and will not be of sufficiently good print quality. The Editorial Panel reserves the right to refuse publication for contributions which cannot be reproduced adequately.

Page definition information

If possible, contributions should be designed to fit standard A4 paper size, leaving a margin of at least one inch (1") on all sides. Camera ready copy should be sent in **single-sided** format, with page numbers written lightly **on the back**. Ideally, all fount sizes used should be no smaller than 10pt for clarity. Contributions should attempt to make adequate use of the space, filling at least 60% of each page, including the last one. Authors should note that all contributions may be sub-edited appropriately to make efficient use of space.

Deadlines

The production deadlines for the coming year are:

Summer	end of May, 1993	**Winter**	end of November, 1993
Autumn	end of August, 1993	**Spring**	end of February, 1994

Disclaimer

The views and opinions expressed within articles included in the BCS FACS FACTS newsletter are the responsibility of the authors concerned and do not necessarily represent the opinions or views of the editorial panel.

Addresses

Editors:

Dr. Jawed Siddiqi
Dept. of Computing and Management Sciences
Sheffield Hallam University
100 Napier Street
Sheffield, S11 8HD
United Kingdom

Tel: +44 742 533141
E-mail: J.I.Siddiqi@shu.ac.uk

Dr. Brian Monahan
Dept. of Computer Science
University of Manchester
Oxford Road
Manchester, M13 9PL
United Kingdom

Tel: +44 61 275 6137
E-mail: brianm@cs.man.ac.uk

BCS FACS Committee 1992/93

General

General enquiries about the BCS FACS group, the newsletter or its meetings can be made to:

BCS FACS
Department of Computer Studies
Loughborough University of Technology
Loughborough, Leicestershire
LE11 3TU
Tel: +44 509 222676
Fax: +44 509 211586
E-mail: FACS@lut.ac.uk

Membership fees 1993

Standard (i.e. non-BCS members) : £25
BCS members : £10

Discount subscription rates 1993

EATCS : £10
FACS Journal : £33 (6 issues, Vol. 5)

Officers

Chair	Tim Denvir
Treasurer	Roger Stone
Committee Secretary	Richard Mitchell
Membership Secretary	John Cooke
Newsletter Editors	Jawed Siddiqi & Brian Monahan
Publicity	Brian Monahan
BCS SIG representative	David Blyth
BCS SE TC representative	John Boarder (Roger Shaw)
Liaison with FACS Journal	John Cooke
Liaison with BCS FMIS group	Ann Wrightson

Committee Members

Name	Affiliation	Tel:	E-mail
R. Barden	Logica Cambridge Ltd	0223-66343	rosalind@logcam.co.uk
D. Blyth	Incord Ltd.	0202-896834	DBlyth@cix.compulink.co.uk
J. Boarder	Buckinhamshire	0494-22141	
Dr. D.J. Cooke	Loughborough	0509-222676	D.J.Cooke@lut.ac.uk
B.T. Denvir	Translimina Ltd.	081-882 5853	timdenvir@cix.compulink.co.uk
Prof. S.J. Goldsack	Imperial	071-589-5111x5099	sig@ic.doc.ac.uk
Dr. A.J.J. Dick	Bull Research		J.Dick@uk03.bull.co.uk
R.B. Jones	ICL Winnersh	0734-693131x6536	R.B.Jones%win0109.uucp@uknet.ac.uk
Dr. R.J. Mitchell	Brighton	0273-642458	rjm4@unix.brighton.ac.uk
Dr. B.Q. Monahan	Manchester	061-275-6137	brianm@cs.man.ac.uk
Prof. A. Norcliffe	Sheffield Hallam	0742-720911x2473	A.Norcliffe@scp.ac.uk
R.C.F. Shaw	Lloyd's Register	081-681-4040x4818	Roger.Shaw@aie.lreg.co.uk
Dr. J.I.A. Siddiqi	Sheffield Hallam	0742-533141	J.I.Siddiqi@SHU.AC.UK
Prof. D. Simpson	Brighton	0273-600900x2450	ds33@unix.bton.ac.uk
Dr. R.G. Stone	Loughborough	0509-222686	R.G.Stone@lut.ac.uk
D.R. Till	City	071-477-8552	till@cs.city.ac.uk
A. Wrightson	Central Lancashire	0772-893242	annw@sc.uclan.ac.uk