FACS

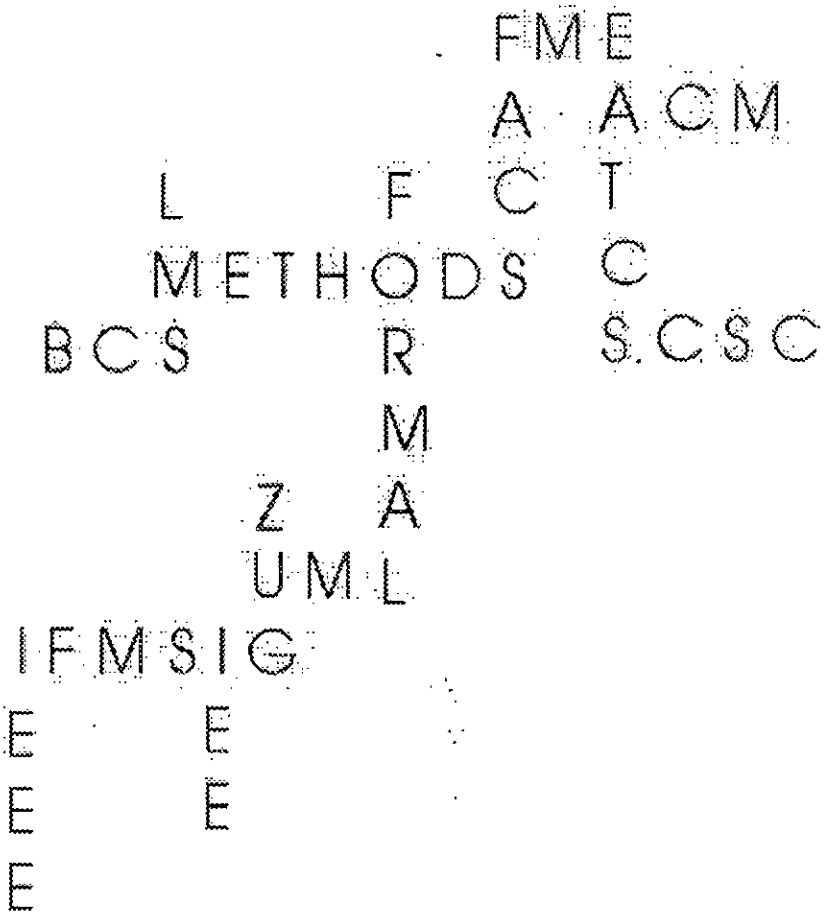
A

C

T

S

Issue 2005-4
December 2005



BCS FACS

The Newsletter of the Formal Aspects of Computing Science (FACS) Specialist Group

ISSN 0950-1231

About *FACS FACTS*

FACS FACTS [ISSN: 0950-1231] is the newsletter of the BCS Specialist Group on Formal Aspects of Computing Science (FACS). *FACS FACTS* is distributed in electronic form to all FACS members.

FACS FACTS is published four times a year: **March**, **June**, **September** and **December**. Submissions are always welcome. Please see the advert on **page 36** for further details or visit the newsletter area of the FACS website [http://www.bcs-facs.org/newsletter].

Back issues of *FACS FACTS* are available to download from:

http://www.bcs-facs.org/newsletter/facsfactsarchive.html

The *FACS FACTS* Team

Contributors to this Issue

Kelly Androutsopoulos, Richard Banach, Dines Bjørner, Paul Boca, Jonathan Bowen, Rob Delicata, John Derrick, Caroline Dickerson, George Eleftherakis, José Fiadeiro, John Fitzgerald, Richard Paige, F.X. Reid, Judi Romijn, Margaret West

Contents

The activities of FACS (i.e. **sponsoring conferences and workshops**, **offering student bursaries** and **hosting evening seminars**) is funded solely from membership subscriptions. The more paid-up FACS members we have, the more we can do ☺

If you would like to become a FACS member – or renew your lapsed membership – please complete the membership form on **page 63** of this issue of *FACS FACTS*. See the **SPECIAL OFFER** on the journal too.

If you have any questions about FACS, please send these to Paul Boca [Paul.Boca@virgin.net], the Membership Secretary.

Editorial

Jonathan Bowen & Paul Boca, BCS-FACS

FACS has had a good year. Our Evening Seminars pilot scheme has turned out to be very popular. Since the last newsletter, we have had seminars from Professor Muffy Calder (using process algebras to model biochemical pathways), Professor Martin Henson (a Z-like logic for specification and program development) and Professor Richard Bornat (separation logic, a reworking of Hoare logic with pointers). If you missed these or any of the preceding seminars – and they were all excellent – the slides are available from the BCS-FACS website [http://www.bcs-facs.org/events/EveningSeminars]. FACS would like to thank all five speakers for giving excellent seminars and for lending their support. They have set some very high standards, which we plan to maintain.

We are continuing the evening seminar series in 2006, and the organizers (Paul Boca, Jonathan Bowen and Jawed Siddiqi) are currently putting together the programme. The first seminar, entitled *Formal Methods in the Last 25 Years,* will take place on **30 January 2006** (see the advert on **page 58** of this issue). Details of the other seminars will be announced soon on the FACS mailing list, in the newsletter and on the above website. If you have ideas for speakers that you would like to hear and subjects that you think would be of particular interest, please contact one or more of the organizers.

FACS has just held its Christmas Meeting on *Formal Methods and Testing*. Thanks to Rob Hierons, Chair of the FORTEST subgroup of FACS, and Paul Boca for organizing the event. We have already started discussing plans for the 2006 Christmas meeting. Details will be announced on the usual channels.

We hope 2006 will be just as successful as 2005 for FACS. If anyone would like to organize a FACS event, please contact Jonathan Bowen, the BCS-FACS Chair, on jonathan.bowen@lsbu.ac.uk. Also contact Jonathan Bowen if you would like to apply for FACS support, moral or otherwise. FACS has limited funds that can be made available with committee approval for sponsorship of suitable conferences. You are welcome to discuss this informally at first to see what might be appropriate. In any case, it is relatively easy to provide best paper prizes in the form of FACS membership and subscription to the *Formal Aspects of Computing* journal.

Speaking of which, in 2006 the FAC journal comes of age. To mark its 18th birthday, Springer is generously giving all those renewing their FACS memberships, and joining FACS for the first time, **free** online access to the *Formal Aspects of Computing* journal. In addition, for just £5 you can subscribe to the paper copy of the journal too. However, please note that the £5 special offer expires at the end of February 2006. From 1 March 2006, the paper copy will cost £48. So do fill out the registration form in this issue and return it without delay! Thank you to Beverley Ford and Christiane Notarmarco at Springer, and FACS committee member John Cooke for making this happen.

It remains to wish you all *Season's Greetings* on behalf of the FACS Committee. We hope to see you at one (or more) of the seminars in 2006.

Call for Proposals

FM'08: The 15th International Symposium on Formal Methods

Formal Methods Europe invites proposals from organisations interested in hosting the 15th International Symposium on Formal Methods (FM'08), to be held in Europe in Spring 2008.

Formal Methods Europe (FME) [http://www.fmeurope.org] is a worldwide association bringing together researchers and practitioners in all aspects of formal methods for the development, analysis and maintenance of computing systems and software. Its activities include the dissemination of research findings and industrial experience through symposia and sponsored events, notably the long running and highly regarded series of "FM" symposia. The 14th symposium in the series [http://fm06.mcmaster.ca] is being organized by McMaster University, Ontario, Canada for August 2006. We invite enquiries and proposals from potential hosts for the 15th Symposium (FM'08), which will be held at a European location, in spring 2008.

Symposia take place approximately every 18 months and have at their core an international technical conference based on high quality refereed papers with published proceedings. A programme of workshops, tutorials and exhibitions adds to the event. However, there are few fixed rules regarding the format and potential hosts are invited to discuss options with FME. Attendances typically range from 125 to 260 participants.

Important Dates

January 15, 2006	Symposium proposals due *Following receipt of proposals, FME's board will discuss proposals by email with the proposers.*
April 15, 2006	Notification of proposal acceptance

Proposal Submission

Proposals should be regarded as expressions of interest and will serve as a basis for discussion. They should be no more than 3 pages in length and should contain:

- A brief description of why hosting the symposium is of interest to the proposing institution.
- The names and contact information (web page, email address) of the proposed organising committee.
- Previous experience in events organisation.
- A description of the facilities that will be available for the symposium.

We welcome informal enquiries. Please send your proposals and any enquiries by electronic mail to: info@fmeurope.org. Proposals should be submitted in PDF.

Train Column
Dines Bjørner

Transportation Nets*

A Domain Model

Dines Bjørner†

This is a technical note in draft form, published in the Train Column of Issue 2005-4 of *FACS FACTS*. With this document I wish to again illustrate the concept of domain modelling. I challenge readers to come up with similar models, informally phrased in altogether different ways and/or formalised in specification languages (B, CafeOBJ, Casl, VDM-SL, Z, etcetera) alternative to the present specification language (RSL). And I challenge readers to help "complete" the present modelling effort: express well-formèdness criteria (axiomatically or otherwise), additional facets, etc.

Abstract

Multi-modal transportation nets consists of segments and junctions. Segments are single-modal and are either roads, or rail tracks, or air-lanes or shipping lanes, but only exactly one of these. Junctions are multi-modal and can be thought of as a non-empty subset of road intersections, train stations, airports and harbours. We present a narrative and a formalisation of multi-modal transportation nets while exploring notions of single or multi-modal paths and routes, composition and decomposition of nets from, respectively into several such, cost of transports between junctions, and so on.

In this report we develop — according to the triptych principles of software engineering — a variety of software systems for transporation applications.

This is a draft report. It is issued for the December 2005 issue of the electronic newsletter *FACS FACTS*. It is issued in order to see whether there are "other people out there" who are interested in developing domain models, and, if so, whether they might contribute to, say some form of domain theory for transportation nets.

†Email:bjorner@gmail.com

Contents

1 Introduction

1.1 Multi-Modal Transportation Nets

By *transport* we mean that something, say *freight* is being *moved* along some *route* driven so by some *conveyour* propelled by some *motive force*.

Thus there are five phenomena involved in *transport*. They are *freight*, *movement*, *route*, *conveyour* and *motive force*.

Routes are part of *transportation nets*. The nets are seen as a set of junctions connected by segments. The junctions are either street (or road) intersections, or are train stations, or are airports, or are harbours. The corresponding segments, consequentially, are roads, rail lines, air corridors (air lanes), or shipping lanes. A route is then an alternating sequence of junctions and segments.

Conveyours consequentially are either cars (private automobiles, taxis, busses, and trucks), or are trains (passenger, freight, or mixed (for example auto trains)), or are aircraft, or are ships.

Multi-modality seemingly complicates matters: the ability to move along a route, by a car along a road, transferring to a train at a junction which is both a street intersection and a train station, then transferring the train onto a ship at a station/harbour junction, and so on.

1.2 The TripTych Paradigm of Software Development

There is the dogma:

- Before software can be designed its requirements must be understood.
- Before requirements can be prescribed the underlying application domain must be understood.
- To understand the requirements their prescription must both
  - be in the natural, professional language of the domain and of software engineering
  - and must be formalised.
- To understand the domain its description must both
  - be in the natural, professional language of the domain
  - and must be formalised.

And there are therefore the consequences of the dogma:

- In software development

  - we must first — and carefully — describe the domain,
  - then, based on the domain description, prescribe the requirements,
  - and, finally, based on the requirements prescription, design the software.

- The phases outlined above are linked:
  - precise, formal relations can be established between the domain description and the requirements prescription, and
  - precise, formal relations can be established between the requirements prescription and the software specification.

The consequences, we believe, of adhering to the above phase-wise development are:

- That the emphasis
  - on domain prescription as a prerequisite for requirements prescription secures the right computing system;
- and the emphasis
  - on requirements prescription as a prerequisite for software specification secures that the computing system is right.

1.3 Aims and Objectives

- **Aims:** to present a domain description.
- **Objectives:** to hopefully spur further research into and use of domain theories cum domain engineering.

The main reference to the triptych approach is [1].

2 Net Topology

We conceptualise as segments the physically manifest phenomena of roads (between adjacent street intersections), rail tracks (between adjacent train stations), air lanes (between adjacent airports) and shipping lanes (between adjacent harbours). We likewise conceptualise as junctions street intersections, train stations, airports and harbours.

2.1 Nets, Segments and Junctions

1. Nets consist of one or more segments and two or more junctions:

type
  N, S, J
value
  obs_Ss: N → S-**set**
  obs_Js: N → J-**set**
axiom
  ∀ n:N • **card** obs_Ss(n) ≥ 1 ∧ **card** obs_Js(n) ≥ 2

Annotations:

- N, S, J are considered abstract types, i.e., sorts. N, S and J are type names, i.e., names of types of values. Values of type N are nets, values of type S are segments and values of type J are junctions.
- One can observe from nets, n, their (one or more) segments (obs_Ss(n)) and their (two or more) junctions (obs_Js(n)); n is a value of type N.
- Functions have names, obs_Ss, and obs_Cs, and functions, f, have signatures, f: A → B (not illustrated), where A and B are type names. A designates the definition set of f and B the range set.
- A-**set** is a type expression. It denotes the type whose values are finite, possibly empty set of A values.
- These observer functions are postulated.
- They cannot be formally defined.
- They are "defined" once a net has been pointed out[1]
- The axiom expresses that in any net there is at least one segment and at least two junctions.

Applying the observer functions to the net of Fig. 1 yields:

obs_Ss(n) = {sa,sb,sc,sd,se,sf,sg,sh,sj,sk}
obs_Js(n) = {j1,j2,j3,j4,j5,j6,j7,j8}

Nets, segments and junctions are physically manifest, i.e., are phenomena.

[1]Take the transportation net Europe. By inspecting it, and by deciding which segments and which associated junctions to focus on (i.e., "the interesting ones") we know which are all the interesting roads, rail tracks, air lanes and shipping lanes, respectively the interesting (associated) street intersections, train stations, airports and harbours.

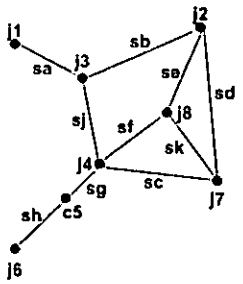


Figure 1: A simple net of segments and junctions

2.2 Segment and Junction Identifications

2. We now assume that segments and junctions have unique identifications.

type
  Si, Ji
value
  obs_Si: S → Si
  obs_Ji: J → Ji

Segment and junction identifications are mental concepts.

3. No two segments have the same segment identifier. And no two junctions have the same junction identifier.

axiom
  ∀ n:N • **card** obs_Ss(n) ≡ **card** {obs_Si(s)|s:S • s ∈ obs_Ss(n)}
  ∀ n:N • **card** obs_Js(n) ≡ **card** {obs_Ji(c)|j:J • j ∈ obs_Js(n)}

Annotations:

- **card** set expresses the cardinality of set, i.e., its number of distinct elements.
- {f(a)|a:A • p(a)} expresses the set of all those B elements f(a) where a is of type A and has property p(a) [where we do not further state f, A and B. p is a predicate, i.e., a function, here from A into truth values of type **Bool**, for Boolean].
- The axioms now express that the number of segments in n is the same as the number of segment identifiers of n — which is a circumscription for: No two segments have the same segment identifier.
- Similarly for junctions.

The constraints that limit identification of segments and junctions can be physically motivated: Think of the geographic $(x, y, z$ co-ordinate) point spaces "occupied" by a segment or by a junction. These points must necessarily be distinct for otherwise physically distinct segments and junctions. Segments may thus cross each other without the crossing point (in x, y space) being a junction, but, for example, one segment may, at the crossing point be physically above the other segment (tunnels, bridges, etc.).

2.3 Segment and Junction Reference Identifications

4. Segments are delimited by two distinct junctions. From a segment one can also observe, obs_Cis, the identifications of the delimiting junctions.

type
  Jip = {|{ji,ji′}:Ji-**set** • ji≠ji′|}
value
  obs_Jis: S → Jip

Annotations:

- {|a:A • p(a)|} is a subtype expression. It expresses a subset of type A, namely those A values which enjoy property p(a) [p is a predicate, i.e., a function, here from A into truth values in the type **Bool**]. In the above p(a) is ji≠ji′.
- In this case Jip is the subtype of Ji-**set** whose values are exactly 2 element sets of Ji elements.

5. Any junction has a finite, but non-zero number of segments connected to it. From a junction one can also observe, obs_Sis, the identifications of the connected segments.

type
  Si1 = {|sis:Si-**set**•**card** sis ≥1|}
value
  obs_Sis: J → Si1

Annotations:

- Si1 is the type whose values are non-empty, but still finite sets of Si values.

One cannot from a segment alone observe the connected junctions. One can only refer to them. Similarly: one cannot from a junction alone observe the connected segments. One can only refer to them. The identifications serve the role of being referents.

6. In any net, if s is a segment connected to connectors identified by ji and ji′, respectively, then there must exist connectors j and j′ which have these identifications and such that the identification si of s is observable from both j and j′.

axiom
∀ n:N, s:S • s ∈ obs_Ss(n) ⇒
let {ji,ji′} = obs_Jis(s) **in**
∃! j,j′:J • {j,j′}⊆obs_Js(n) ∧ j≠j′ ∧
obs_Si(s) ∈ obs_Sis(c) ∩ obs_Sis(c′) **end**

Annotations:

- We read the above axiom:
  - for all nets n and for all segments s in n
  - let ji and ji′be the two distinct junction identifications observable from s, then
  - there exists exactly two distinct junctions, j and j′ of the net, such that
  - the segment identification of s is in both the sets of segment identifications observable from j and j′.

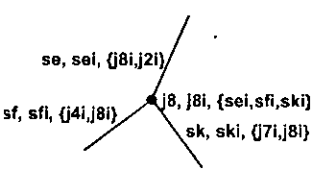


Figure 2: One junction and its connected segments

Figure 2 illustrates the relation between observed identifications of segments and junctions.

The above constraints take on the mantle of being laws of nets: If segments and junctions have distinct identifications, then the above must follow as a law of man-made artifacts.

7. Vice-versa: In any net, if j is a junction connecting segments identified by si, si′, ..., si″ then there must exist segments s, s′, ..., s″ which have these identifications and such that the identification ji of j is observable from all s, s′, ..., s″.

axiom
∀ n:N, j:J • j ∈ obs_Js(n) ⇒
let sis = obs_Sis(c), ji = obs_Ji(j) **in**
∃! ss:S-**set** • ss⊆obs_Ss(n) ∧ **card** ss=**card** sis ∧
sis = {|obs_Si(s)|s:S•s ∈ ss|} **end**

Annotations:

- Let us read the above axiom:
  - for all nets, n, and all junctions, j, of that net
  - let sis be the set of segment identifications observed from j, and let ji be the junction identifier of j, then
  - there exists a unique set, ss, of segments of n with as many segments as there are segment identifications in sis, and such that
  - sis is exactly the set of segment identifications of segments in ss.

2.4 Paths and Routes

8. By a path we shall understand a triplet of a junction identification, a segment identification and a junction identification.

```
type
  P = Ji × Si × Ji
value
  paths: N → P-set
  paths(n) ≡
    {(ji,si,ji′)|s:S,ji,ji′:Ji,si:Si• s ∈ obs_Ss(n)∧{ji,ji′} ∈ obs_Jis(s)∧si=obs_Si(s)}
```

Annotations:

- Paths are modelled as Cartesians.
- One can generate all the paths of a net.
- It is the set of path triplets, two for each segment of the net and such that the pair of junction identifications, ji and ji′, observable from a segment is at either "end" of the triplet, and such that the segment identification is common to the two triplets (and in the "middle").

Paths, and as we shall see next, routes are mental concepts.

9. By a route of a net we shall understand a list, i.e., a sequence of paths as follows:
   - A sequence of just one path of the net is a route.
   - If r and r′ are routes of the net such that the last junction identification, ji, of the last path, (_,_,ji) of r and the first junction identification, ji′, of the first path (ji′,_,_) of r′ are the same, i.e., ji=ji′, then $r^\frown r'$ is a route.
   - Only routes that can be generated by uses of the first (the basis) and the second (the induction) clause above qualify as proper routes of a net.

```
type
  R = {|r:P*•wf_R(r)|}
value
  wf_R: P* → Bool
  wf_R(r) ≡
    ∀ i:Nat • {i,i+1}⊆inds(r) ⇒
      let (_,_ji)=r(i), (ji'_,_)=r(i+1) in ji = ji' end

  routes: N → R-infset
  routes(n) ≡
    let rs = {⟨p⟩|p:P•p ∈ paths(n)}
           ∪ {r^r'|r,r':R•{r,r'}⊆rs∧wf_R(r^r')} in
    rs end
```

Annotations:

- Routes are well-formed sequences of paths.
- A sequence of paths is a well-formed route if adjacent path elements of the route share junction identification.
- Give a net we can compute all its routes as follows:
  - let rs be the set of routes to be computed. It consists first of all the single path routes of the net.
  - Then rs also contains the concatenation of all pairs of routes, r and r', such that these are members of rs and such that their concatenation is a well-formed route.
  - If the net is circular then the set rs is an infinite set of routes. The least fix point of the recursive equation in rs is the solution to the "routes" computation.

2.5 Segment and Junction Identifications of Routes

10. For future purposes we need be able to identify various segment and junction identifications as well as various segments and junctions of a route.

```
value
  xtr_Jis: R → Ci-set, xtr_Sis: R → Si-set
  xtr_Jis(r) ≡ case r of ⟨⟩ → {}, ⟨(ji,_,ji')⟩^r' → {ji,ji'}∪ xtr_Jis(r') end
  xtr_Sis(r) ≡ case r of ⟨⟩ → {}, ⟨(_,si,_)⟩^r' → {si}∪ xtr_Sis(r') end

  xtr_Ss: N × Ji → S-set
  xtr_Ss(n,ji) ≡ {s|s:S•s ∈ obs_Ss(n) ∧ ji ∈ obs_Jis(s)}
```

xtr_C: N × Ji → C, xtr_S: N × Si → S
xtr_C(n,ji) ≡ **let** j:J • j ∈ obs_Js(n) ∧ ji=obs_Ji(j) **in** j **end**
xtr_S(n,si) ≡ **let** s:S • s ∈ obs_Ss(n) ∧ si=obs_Si(s) **in** s **end**

first_Ji: R ~→ Ji, last_Ji: R ~→ Ji
first_Ji(r) ≡ **case** r **of** ⟨⟩ → **chaos**, ⟨(ji,_,_)⟩^r′ → ji **end**
last_Ji(r) ≡ **case** r **of** ⟨⟩ → **chaos**, r′^⟨(_,_,ji)⟩ → ji **end**

first_Si: R ~→ Si, last_Si: R ~→ Si
first_Si(r) ≡ **case** r **of** ⟨⟩ → **chaos**, ⟨(_,si,_)⟩^r′ → si **end**
last_Si(r) ≡ **case** r **of** ⟨⟩ → **chaos**, r′^⟨(_,si,_)⟩ → si **end**

first_J: R × N ~→ J, last_J: R × N ~→ J
first_J(r,n) ≡ xtr_J(first_Ji(r),n)
last_J(r,n) ≡ xtr_J(last_Ji(r),n)

first_S: R × N ~→ S, last_S: R × N ~→ S
first_S(r,n) ≡ xtr_S(first_Si(r),n)
last_S(r,n) ≡ xtr_S(last_Si(r),n)

Annotations:

- Given a route one can extract the set of all its junction identifications.
  - If the route is empty, then the set is empty.
  - If the route is not empty then it consists of at least one path and the set of junction identifications is the pair of junction identifications of the path together with set of junction identifications of the remaining route.
  - Possible double "counting up" of route adjacent junction identifications "collapse", in the resulting set into one junction identification. (Similarly for cyclic routes.)
- Given a route one can similarly extract the set of all its segment identifications.
- Given a net and a junction identification one can extract all the segments connected to the identified junction.
- Given a net and a junction identification one can extract the identified junction.
- Given a net and a segment identification one can extract the identified segment.
- Given a route one can extract the first junction identification of the route.
  - This extraction should not be applied to empty routes.
  - A non-empty route can always be thought of as its first path and the remaining route. The first junction identification of the route is the first junction identification of that (first) path.

- Given a route one can similarly extract the last junction identification of the route.
- Given a route one can similarly extract the first segment identification of the route.
- Given a route one can similarly extract the last segment identification of the route.
- And similarly for extracting the first and last junctions, respectively first and last segments of a route.

2.6 Circular and Pendular Routes

11. A route is circular if the same junction identification either occurs more than twice in the route, or if it occurs as both the first and the last junction identification of the route. Given a net we can compute the set of all non-circular routes by omitting from the above pairs of routes, r and r′, where the two paths share more than one junction identification.

```
non_circular_routes: N → R-set
non_circular_routes(n) ≡
  let rs = {⟨p⟩|p:P•p ∈ paths(n)}
           ∪ {r⌢r′|r,r′:R•{r,r′}⊆rs∧wf_R(r⌢r′)∧non_circular(r,r′)} in
  rs end
non_circular: R×R → Bool
non_circular(r,r′) ≡ card xtr_Jis(r) ∩ xtr_Jis(r′) =1
```

Annotations:

- To express the finite set of all non-circular routes
  - is to re-express the set of all routes
  - except constrained by the further predicate: non_circular.
- An otherwise well-formed route consisting of a first part r and a remaining part r′
  - is non-circular if the two parts share at most one junction identification.

12. Let a path be (ji_f, si, ji_t), then (ji_t, si, ji_f) is a *reverse path*. That is: the two junction identifications of a path are reversed in the reverse path. A route, rr, is the reverse route of a route r if the ith path of rr is the reverse path of the $n-i+1$'st path of r where n is the length of the route r, i.e., its number of paths. A route is a *pendular* route if it is of an even length and the second half (which is a route) is the reverse of the first half route.

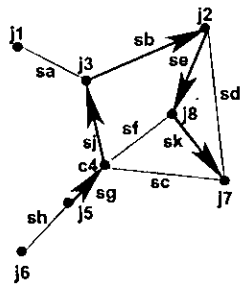


Figure 3: A route, graphically and as an expression

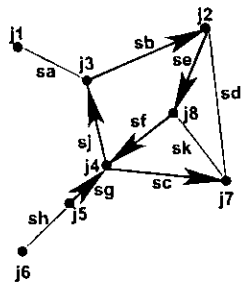


Figure 4: A circular route, graphically and as an expression

```
value
  reverse: P → P
  reverse(jif,si,jit) ≡ (jit,si,jif)

  reverse: R → R
  reverse(r) ≡
    case r of
      ⟨⟩ → ⟨⟩,
      ⟨(jif,si,jit)⟩^r′ → reverse(r′)^⟨(jit,si,jif)⟩
    end

  reverse(r) ≡ ⟨reverse(r(i))|i in [n..1]⟩

  pendular: R → R
  pendular(r) ≡ r^reverse(r)

  is_pendular(r) ≡ ∃ r′,r″:R • r′^r″ = r ∧ r″=reverse(r′)
```

Annotations:

- The reverse of a path is a path with the same segment identification, but with reverse junction identifications.
- The reverse of a route, r, is
  - the empty route if r is empty, and otherwise
  - it is the reverse route of all of r except the first path of r concatenated (juxtaposed) with the singleton route of the reverse path of the first path of r.
- Given a route, r, we can construct a pendular route whose first half is the route r and whose second half is the reverse route of r.
- A (an even length) route is a pendular route if it can be expressed as the concatenation of two (equal length) routes, r′ and r″ such that r″ is the reverse of r′, that is, if its second half is the reverse of its first half.

2.7 Connected Nets

13. A net is connected if for any two junctions of the net there is a route between them.

```
value
  is_connected: N → Bool
  is_connected(n) ≡
    ∀ j,j':J • {j,j'}⊆obs_Js(n) ∧ j≠j' ⇒
      let (ji,ji') = (obs_Ji(j),obs_Ji(j')) in
      ∃ r:R • r ∈ routes(n) ∧
        first_Ji(r) = ji ∧ last_Ji(r) = ji' end
```

Annotations:

- A net n is connected if
  - for all two distinct connectors of the net
  - where ji and ji′ are their junction identifications,
  - there exists a route, r, of the net,
  - whose first junction identification is ji and whose last junction identification is ji′.

2.8 Net Decomposition

14. One can decompose a net into all its connected subnets. If a net exhaustively consists of m disconnected nets, then for any pair of nets in different disconnected nets it is the case that they share no junctions and no segments. The set of disconnected nets is the smallest such set that together makes up all the segments and all the junctions of the ("original") net.

value
decompose: N → N-**set**
decompose(n) **as** ns
obs_Ss(n) = ∪{obs_Ss(n′)|n′:N•n′ ∈ ns} ∧
obs_Js(n) = ∪{obs_Js(n′)|n′:N•n′ ∈ ns} ∧
{} = ∩{obs_Ss(n′)|n′:N•n′ ∈ ns} ∧
{} = ∩{obs_Js(n′)|n′:N•n′ ∈ ns} ∧
∀ n′:N•n′ ∈ ns ⇒ connected(n′) ∧ ...

Annotations:

- A set ns of nets constitutes a decomposition of a net, n,
  (a) if all the segments of n appear in some net of ns,
  (b) if all the junctions of n appear in some net of ns,
  (c) if no two or more distinct nets of ns share segments,
  (d) if no two or more distinct nets of ns share junctions, and
  (e) if all nets of ns are connected.
- **Comment:** It appears that items 3 and 4 are unnecessary, that is, are properties once items 1, 2 and 5 hold.

That is, we have the following:

Lemma:
∀ n:N •
let ns = decompose (n) **in**
∀ n′,n″:N • {n′,n″}⊆ns ∧ n′≠n″ ⇒
obs_Ss(n′) ∩ obs_Ss(n″) = {} ∧
obs_Js(n′) ∩ obs_Js(n″) = {} **end**

The above 14 items define a lot of what there is to know about transportation nets if we only operate with the sorts that have been introduced (N, S, Si, J, Ji) and the observer functions that have likewise been introduced (obs_Ss, obs_Js, obs_Si, obs_Ji, obs_Jis and obs_Sis). The relationships between sorts, i.e., net, segment, segment identification, junction and junction identification values are expressed by the axioms. The above is a so-called property-oriented model of the topology of transportation nets. That model is abstract in that it does not hint at a mathematical model or at a data structure representation of nets, segments and junctions, let alone their topology. By topology we shall here mean how segments and junctions are "wired up". The axioms above guarantee that no segment of a net is left "dangling": It is always connected to two distinct junctions; and no junction of a net is left isolated: It is always connected to some segments of the net.

We have tacitly assumed that all segments are two way segments, that is, transport can take place in either direction. Hence a segment gives rise to two paths.

3 Multi-Modal Nets

Interesting transportation nets are multi-modal. That is, they consist of segments of different transport modalities: roads, rails, air-lanes, shipping lanes, and, within these different categories. Thus roads can be either freeways, motor-ways, ordinary highways, and so on.

3.1 General Issues

15. We introduce a concept, M, of transport mode. M is a small set of distinct, but otherwise further undefined tokens. An m in M designates a transport modality.

```
type
  M
```

3.2 Segment and Junction Modes

16. With each segment, s, we can associate a single mode, m, and with each junction we can associate the set of modes of its connected segments.

```
value
  obs_M: S → M
  obs_Ms: J → M-set
axiom
  ∀ n:N, j:J • j ∈ obs_Js(n) ⇒
    let ss = xtr_Ss(n,obs_Ji(j)) in
    obs_Ms(j) = {obs_M(s)|s:S • s ∈ ss} end
  ∀ n:N, s:S • s ∈ obs_Ss(n) ⇒
    let {ji,ji'} = obs_Jis(s) in
    let {j,j'} = {xtr_J(n,ji),xtr_J(n,ji')} in
    obs_M(s) ∈ obs_Ms(j) ∩ obs_Ms(j') end end
```

Annotations:

- From a segment one can observe its mode.
- From a junction one can observe its set of modes.
- Let us read the first axiom:
  - for all net, n, and all junctions, j, of that net
  - let ss be the set of segments connected to j,
  - now the set of modes of c is equal to the set of modes of the segments in ss.
- Let us read the second axiom:

  - for all net, n, and all segments, s, of that net
  - let ji and ji′ be the junction identifiers of the two junctions to which s is connected, and
  - let j and j′ be the two corresponding junctions,
  - then the segment mode is in both the set of modes of the two junctions.
- We can define a function, xtr_Ss, which from a net, n, and a junction identification, ji, extracts the set of segments, ss, connected to the junction identified by ji.
- xtr_Ss(n,ji) yields the set of segments, ss, in the net n for which ji is one of the observed junction identifications of s.
- And we can define a function, xtr_J, of signature N × Ji → J, which when applied to a net, n, and a junction identification, ji,
- extracts the junction in the net which has that junction identifier.

3.3 Single-Modal Nets and Net Projection

17. Given a multi-modal net one can project it onto a set of single modality nets, namely one for each modality registered in the multi-modal net.

```
type
  mmN = {|n:N • card xtr_Ms(n) > 1|}
  smN = {|n:N • card xtr_Ms(n) = 1|}
value
  xtr_Ms: N → M-set
  xtr_Ms(n) ≡ {obs_M(s) | s:S • s ∈ obs_Ss(n)}

  projs: N → smN-set
  projs(n) ≡ {proj(n,m) | m:M • m ∈ xtr_Ms(n)}

  proj: N × M → smN
  proj(n,m) as n′
    post
      let ss = obs_Ss(n), ss′ = obs_Ss(n′),
          js = obs_Js(n), js′ = obs_Js(n′) in
      ss′ = {s | s:S • s ∈ ss ∧ m=obs_M(s)} ∧
      js′ = {j | j:J • j ∈ js ∧ m ∈ obs_Ms(j)}
      end
```

Annotations:

- A multi-modal net is a net with more than one mode. mmN is thus the subtype of nets, n:N, which are multi-modal.

- A single-modal net is a net with exactly one mode. smN is thus the subtype of nets, n:N, which are multi-modal.
- The xtr_Ms function extracts the mode of every segment of a net.
- The projs function applies to any net, n:N, and yields the set of single-modal subnets of n, one for each mode of n. The projs function makes use of the proj function.
- The proj function applies to any n, n:N, and any mode of that net, and yields the single-modal subnet on n whose mode is the given mode.
  - The proj function is expressed by a post condition, i.e., a predicate that characterises the necessary and sufficient relation between the argument net, n, and the result net n′.
  - In a single-modal net, n′, projected from a multi-modal net, n, and of mode m, we keep exactly those segments, ss′, of n whose mode is m,
  - and we keep exactly those junctions, js′, of n whose mode contains m.
  - No more is needed in order to express the necessary and sufficient condition for a single-modal net to be a subnet of a proper net.
  - That is, some single-modal nets are not proper nets since in proper nets every junction have the set of modes of all the segments connected to the junction.

4 Segment and Junction Attributes

4.1 Segment and Junction Attribute Observations

We now enrich our segments and junctions.

18. Segments have lengths.
19. Junctions have modality-determined lengths between pairs of (same such modality) segments connected to the junction.
20. Segments have standard transportation times, i.e., time durations that it takes to transport any number of units of freight from one end of the segment to the other.
21. Junctions have standard transfer time per modality of transport between pairs of segments connected to the junction.
22. Junctions have standard arrival time per modality of transport.
23. Junctions have standard departure times per modality of transport.
24. Segments have standard costs of transporting a unit of freight from one end of the segment to the other end.

25. Junctions have standard costs of transporting a unit of freight from the end of one connecting segment to the beginning of another connecting segment.

We can now assess

- (i) length of a route,
- (ii) shortest routes between two junctions,
- (iii) duration time of standard transport along a route, including transfer, stopover and possible reloading times at junctions, and
- (iv) shortest duration time route of standard transport between two junctions.

type
  L, TI
value
  ms:M-**set**, **axiom** ms≠{}
  obs_L: S → L
  obs_L: Si × J × M × Si → L
  obs_TI: S → TI
  obs_TI: Si × J × Si → TI
  obs_TI: J × M $\xrightarrow{\sim}$ TI, **pre** obs_TI(j,m): m ∈ obs_Ms(j)
  obs_TI: J × M × M $\xrightarrow{\sim}$ TI, **pre** obs_TI(j,m,m′): {m,m′}⊆obs_Ms(j)
  obs_arr_TI: J × M $\xrightarrow{\sim}$ TI, **pre** obs_arr_TI(j,m): m ∈ obs_Ms(j)
  obs_dep_TI: J × M $\xrightarrow{\sim}$ TI, **pre** obs_dep_TI(j,m): m ∈ obs_Ms(j)
  +: L × L → L
  +: TI × TI → TI

Annotations:

- L and Ti are sorts designating length and time values.
- ms denotes a non-empty set of modes.
- From a segment one can observe, obs_L, its length.
- From a segment one can observe, obs_TI, a time duration for a normal conveyour of the mode of the segment to travel the length of the segment.
- From a junction and a mode (of that junction) one can observe, obs_TI, a time duration for a normal conveyour of the mode to cross, i.e., to travel through the junction.
- From a junction and a pair of modes (m and m′ of that junction) one can observe, obs_TI, a time duration which represents the normal time it takes to transfer freight from a conveyour of mode m to a conveyour of mode m′. (The two modes may be the same.)

- From a junction and a mode (of that junction) one can observe, obs_arr_TI, a time duration for an item of freight destined for a normal conveyour of the mode to arrive and be "entry" processed (including loaded) at that junction.
- From a junction and a mode (of that junction) one can observe, obs_dep_TI, a time duration for an item of freight destined for a normal conveyour of the mode to arrive and be "exit" processed (including unloaded) at that junction.
- One can add lengths.
- One can add time durations.

4.2 Route Lengths

26. One can compute the length of a route of a net and one can find the shortest such route between two identified junctions.

```
value
  length: R → N ~→ L
  length(r)(n) ≡
    case r of
      ⟨⟩ → 0,
      ⟨(jf,si,jt)⟩ → obs_L(xtr_S(si,n)),
      ⟨(ji1,sii,ji2),(jj1,sij,jj2)⟩^r' →
        let si=xtr_S(sii,n),sj=xtr_S(sij,n) in
        obs_L(si) + obs_L(sii,xtr_J(ji2,n),sij) + length(⟨(jj1,sij,jj2)⟩^r') end
    end
    pre: r ∈ routes(n) ∧ ji2=jj1

value
  shortest_route: Ji × Ji → N ~→ R
  shortest_route(jf,jt)(n) ≡
    let rs = routes(n) in
    let crs = {r|r:R•r ∈ rs ∧ first_Ji(r)=jf ∧ last_Ji(r)=jt} in
    let sr:R • sr ∈ crs ∧ ~∃ r:R • r ∈ crs ∧ length(r)(n)<length(sr)(n) in
    sr end end end
    pre: {jf,jt}⊆obs_Jis(n) ∧ jf≠jt
```

Annotations:

- The length of a single modality route of a net
  - is 0 if the route is empty,

  - otherwise it is the length of the first segment of the route plus the length of the rest of the route computed as follows:
    * If the route consists of just one segment, then 0,
    * else, the length of the junction from incident segment to emanating segment plus
    * the length of the rest of the route computed as otherwise specified above.

- The shortest route of a net between two of its identified junctions (the precondition) can be abstractly determined as follows:
  - First we find all the routes, rs, of the net.
  - Then we find those routes, crs, whose first and last connection identifications are the given ones, cf and ct.
  - Amongst those we find a shortest one, that is, one, in crs, for which there are no shorter routes, r, in crs.

4.3 Route Traversal Times

27. One can find the total time it takes to traverse a route, including the times it takes to pass through a junction, and one can find the quickest route between two identified junctions.

```
all_time: R → N → TI
all_time(r)(n) ≡
  obs_arr_TI(xtr_J(first_J(r),n),obs_M(first_S{r}))
  + time(r)(n)
  + obs_dep_TI(xtr_J(last_J{r},n),obs_M(last_S(r)))

time: R → N → TI
time(r)(n) ≡
  case r of
    ⟨⟩ → 0,
    ⟨(jf,si,jt)⟩ → obs_TI(xtr_S(si,n)),
    ⟨(ji1,sii,ji2),(jj1,sij,jj2)⟩^r′ →
      let si=xtr_S(sii,n),sj=xtr_S(sij,n) in
      obs_TI(si) + obs_TI(sii,xtr_J(ji2,n),sij) + time(⟨(jj1,sij,jj2)⟩^r′) end
  end
  pre: r ∈ routes(n) ∧ ji2=jj1

quickest_route: Ji × Ji → N → R
quickest_route(jf,jt)(n) ≡
  let rs = routes(n) in
  let crs = {r|r:R•r ∈ rs ∧ first_Ji(r)=jf ∧ last_Ji(r)=jt} in
```

```
    let qr:R • qr ∈ crs ∧ ~∃ r:R • r ∈ crs ∧ all_time(r)(n)<all_time(qr)(n) in
    qr end end end
```

4.4 Function Lifting

28. Notice how the two functions shortest_route and quickest_route differ only by the length, respectively the time functions. Hence:

```
type
  Q
  FCT = R → N → Q
value
  less: Q × Q → Bool
  lowest: Ji × Ji → N → FCT → R
  lowest(jf,jt)(n)(fct) ≡
    let rs = routes(n) in
    let crs = {r|r:R•r ∈ rs ∧ first_Ji(r)=jf ∧ last_Ji(r)=jt} in
    let lr:R • lr ∈ crs ∧ ~∃ r:R • r ∈ crs ∧ less(fct(r)(n),fct(qr)(n)) in
    lr end end end
```

29. Similarly one could also lift the 'less' predicate:

```
  Q
  PRE = Q × Q → Bool
  FCT = R → N → Q
value
  best: Ji × Ji → N → FCT → PRE → R
  best(cf,ct)(n)(fct)(pre) ≡
    let rs = routes(n) in
    let crs = {r|r:R•r ∈ rs ∧ first_Ji(r)=cf ∧ last_Ji(r)=ct} in
    let br:R • lr ∈ crs ∧ ~∃ r:R • r ∈ crs ∧ pre(fct(r)(n),fct(qr)(n)) in
    br end end end
```

And so on.

4.5 Transportation Costs

30. We can further assess (i) transport costs, (ii) lowest (per unit) freight cost between two junctions, etc. We assume that if a freight item is transported into a junction

and out of that junction by the same modality conveyour, then it is not reloaded, i.e., along segments of the same modality.[2]

```
type
  K, F
value
  obs_K: (S|J) → K
  obs_F: (S|J) → F

  +: K × K → K

  cost: R → N → K
  cost(r)(n) ≡
    case r of
      ⟨⟩ → 0,
      ⟨(jf,si,jt)⟩ →
        obs_K(xtr_J(jf,n))+obs_K(xtr_S(si,n))+obs_K(xtr_J(jt,n))
      ⟨(jf,si,jt),(jf′,si′,jt′)⟩^r′ → assert: jt=jf′
        obs_K(xtr_J(jf,n))+obs_K(xtr_S(si,n))+...+cost(r′)
    end

  cheapest: Ji×Ji → N → ((K×K)→K) → ((K×K)→Bool) → R
  cheapest(jf,jt)(n) ≡
    best(jf,jt)(n)(λ(k1,k2):(K×K)•k1+k2)(λ(k1,k2):(K×K)•k1<k2)
```

5 Road Nets

We wish to view road nets at different levels of abstraction. At a most detailed such level we make no distinction between the road kinds, whether community roads, provincial roads, motor roads or freeways. At another level of abstraction we wish to make exactly those distinctions. And at the least detailed level of abstraction we consider certain road junctions to designate road nets of smaller or larger communities.

31. Figure [A] 5 on the next page shows a road net. Instead of showing junctions J1, J2 and J3 as small black disks we show them as larger circles for reasons that transpire from Fig. [B] 5 on the facing page.

32. Junctions J1, J2 and J3 are considered composite, that is, to represent communities.

[2]This grossly simplifying assumption will be removed later. For the time being it allows us to operate with the simple notion of routes that was introduced above. For the reloading case we need to decorate the route notion, effectively making it into a bill of ladings notion: one that prescribes possible reloading at junctions.

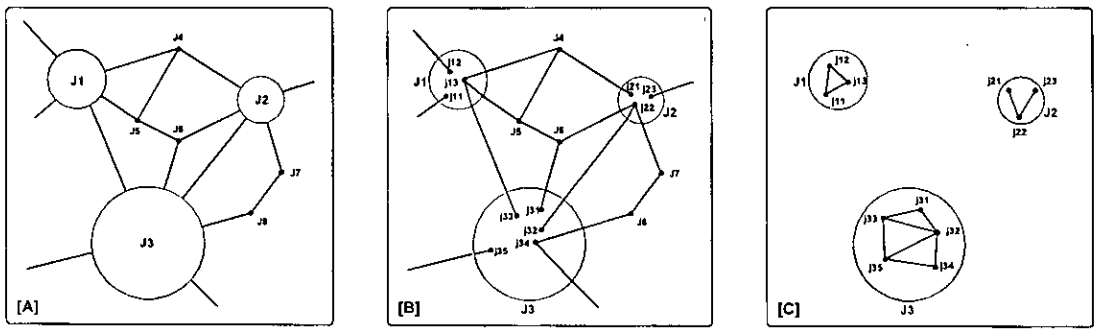


Figure 5: Gross [A] versus semi-detailed [B] road net and community road nets [C]

33. We may consider the road net of Fig.[A] 5 to be an abstraction of the road net hinted at in Fig.[B] 5.

34. Junctions j11, j12, ..., j35 are considered simple embedded junctions.

35. We decide to allow three kinds of junction:

    (a) composite,

    (b) simple embedded and

    (c) simple.

    They are as follows:

    (a) Composite junctions stand for road nets themselves. The junctions of those road nets are all simple embedded junctions.

    (b) Simple embedded junctions are the junctions, hence, of composite junction road nets.

    (c) Simple junctions are those junctions which are not composite (that is: are not standing for road nets) and are not simple embedded junctions (that is: simple, hence un-embedded junctions are those remaining junctions of a net which include modality road).

36. In Fig. [B] 5 we have left out the internal roads, that is, segments of junctions J1, J2 and J3, that is between the simple embedded junctions j11, j12 and j13, between j21, j22 and j23, and between j31, j32, j33, j34 ans j35.

37. The internal segments of junctions J1, J2 and J3 are shown in Fig. [C] 5. They are to be considered complete nets "in and by" themselves.

38. We may consider the implied junction identifications Ji1, Ji2 and Ji3 to be names of communities.

39. We may consider the implied junction identifications ji11, ji12 and ji13 to abstract to J1, ji21, ji22 and ji23 to abstract to J2, and ji31, ji32, ji33, ji34 and ji35 to abstract to J3.

40. We shall assume that from these junction identifications, say jikℓ, one can observe the more abstract junction identifications, i.e., Jik.

41. We shall, conversely, assume that from segment junction identifications one can observe whether they are identifications of composite, of simple embedded or of simple junctions, and, if of composite junctions, that one can further observe which simple embedded junction of the composite junction the segment is connected to.

42. In summary: When considering any multi-modality net and from it project, that is, consider only the net, n_r, of modality road, then we may find that some junctions are composite while some are simple. When then examining the road nets, r_n, contained in composite junctions then we will find that their junctions are simple embedded. The embedded road nets, r_n, otherwise satisfy all the properties (i.e., axioms) of nets in general. To link up the segments of n_r incident upon, that is, connected to composite junctions (in n_r) we provide their junction identifications with two levels of observability: the abstract one that made us see that they were connected to composite junctions (cf. Fig. [A] 5 on the previous page), and a concrete one that enables us to decide which ones of the simple embedded junctions they are "finally" linked to (cf. Fig. [B] 5 on the preceding page).

```
type
  M == road | ...
  Jc, Js, Jse
  Jic, Jis, Jise
  J = Jc | Js | Jse
  Cn
value
  is_composite_J: J → Bool
  is_simple_J: J → Bool
  is_simple_embedded_J: J → Bool
  obs_N: Jc → N
  obs_Jic: Jc → Jic, obs_Jis: Js → Jis, obs_Jise: Jse → Jise
  obs_Cn: Jic → Cn, obs_Cn: Jise → Cn
  obs_Jise: Jic → Jise
axiom
  ∀ j:Jc • is_composite_J(j) ∧ xtr_Ms(obs_N(j,road))={road},
  ∀ j:Js • is_simple_J(j),
```

∀ j:Jse • is_simple_embedded_J(j)

∀ n:N,j:J • j ∈ obs_Js(n) ∧ is_composite_J(j) ⇒
  let rn = obs_N(j) **in**

  end

6 Railway Nets

In Issue 2005-2 of *FACS FACTS* we presented rudiments of a domain model for railways. In this section we shall show how the railway modality of transportation nets can lead us directly into the model given in that earlier issue of *FACS FACTS*.

A transportation net of modality railway has segments (lines between stations) and has junctions (stations).

43. We concretise the concept of modes. Mode m=railway will now designate railway nets:

type
  M == road | railway | ...

44. From a multi-modal transportation net we can project the railway net, rn:RN:

value
  proj: N × {railway} → RN

45. Junctions of a transportation net of modality railway have sub-junctions which are stations:

value
  proj: J × {railway} → ST

46. Segments of a transportation net of modality railway become lines:

value
  proj: S × {railway} → LI

6.1 Lines, Stations, Units and Connectors

The postulated projection functions give us the basic entities of the railway model presented earlier [2]. The model of that reference now "takes over"!

7 Net Dynamics

By net dynamics we shall mean the changing possibilities of flow of conveyors (cars, trains, aircraft, ships, etc.) along segments and through junctions. We speak of direction of flow along segments in terms of *"from the junction at one end of the segment to the junction at the other end"*. And we speak of flow through a junction as *"proceeding from one segment incident upon the junction into a (usually different) segment emanating from that junction"*. Segments connected to a junction are both incident upon that junction and emanates from that junction.

7.1 Segment and Junction States

47. Segments may be open for traffic in either or both directions (between the segments' two junctions [identified by ji_x and ji_y]) or may be closed.

48. We model the state, $s\sigma : S\Sigma$, of a segment, $s : S$, as a set of pairs of junction identifications, namely of the two identifications of the junctions that the segment connects. This state, $s\sigma : S\Sigma$, is

    (a) either empty, i.e., the segment is closed ($\{\}$),

    (b) or has one pair, $\{(ji_x, ji_y)\}$, that is, the segment is open in the direction from junction ji_x to junction ji_y,

    (c) or another pair $\{(ji_y, ji_x)\}$,

    (d) or both pairs $\{(ji_x, ji_y), (ji_y, ji_x)\}$, that is, is open in both directions.

49. Junctions may direct traffic from any subset of incident segments to any subset of emanating segments.

50. We model the state, $j\sigma : J\Sigma$, of a junction, $j : J$, as a set of pairs of segment identifications, namely of identifications of segments connected to the junction.

    (a) Let the set of identifications of segments connected to junction j be $\{si_1, si_2, ..., si_m)\}$.

    (b) If, in some state, $j\sigma$ of the junction, it is possible (allowed) to pass through the junction from the segment identified by si_j to the segment identified by si_k, then the pair (si_j, si_k) is in $j\sigma$.

    (c) The junction state may be empty, i.e., closed: no traffic is allowed through the junction.

(d) Or the junction state may be "anarchic full", that is, it contains all combinations of the pairs of identifiers of segments incident upon the junction.

type
  SΣ = (Ji×Ji)-**set**
  JΣ = (Si×Si)-**set**
value
  obs_SΣ: S → SΣ
  obs_JΣ: J → JΣ

  xtr_Jis: SΣ → Ji-**set**, xtr_Jis(sσ) ≡ {ji|ji:Ji • (ji,_) ∈ obs_sσ ∨ (_,ji) ∈ obs_sσ}
  xtr_Sis: JΣ → Si-**set**, xtr_Sis(jσ) ≡ {si|si:Si • (si,_) ∈ obs_jσ ∨ (_,si) ∈ obs_jσ}
axiom
  ∀ s:S • xtr_Jis(obs_SΣ(s)) ⊆ xtr_Jip(s),
  ∀ j:J • xtr_Sis(obs_JΣ(j)) ⊆ xtr_Sis(j)

Observations:

- A junction, $j : J$, of just one segment, $s : S$, that is, s is a cul de sac, may either be closed, and vehicles trying to enter j will be queued up, or it is open, and vehicles entering j will be lead back to s.
- As a consequence segment s, in order for this latter routing to happen, must be open in both directions when j is "open".
- In general, if the state of a junction j (identified by ji) contains a pair (si_x, si_y) then the state of the designated segments, sx and sy, must respectively contain pairs (ji', ji), respectively (ji, ji''), where $\{ji, ji'\}$ and $(ji, ji''\}$ are the pairs of junction identifications associated with si_x and si_y respectively.
- And this must hold for all states of junctions and adjacent segments.
- This is captured in the axioms below.

axiom
  ...

51. The junction of Fig. 6 shows four segments, identified by A, B, C and D.
52. The figure also suggests a state in which traffic lights prohibit movements from A into J, from B into J,
53. from C via J into A, and from D via J into B.

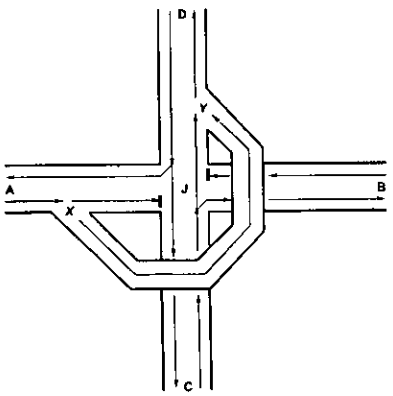


Figure 6: A Special "Carrefour" Junction

54. The "bypass" from A/X into Y/D appears to be such that traffic can always pass from A into D.

55. The current state alluded to in Fig. 6 appears to be:

$$j\sigma_J : \{(A, D), (C, B), (C, D), (D, A), (D, C)\}$$

56. (A, D) is a member of every state in $j\omega_J$ see next section.

7.2 Segment and Junction State Spaces

type
  SΩ = SΣ-**set**
  JΩ = JΣ-**set**
value
  obs_SΩ: S → SΩ
  obs_JΩ: J → JΩ
axiom
  ∀ s:S • obs_SΣ(s) ⊆ obs_SΩ(s),
  ∀ j:J • obs_JΣ(j) ⊆ obs_JΩ(j)

7.3 Open and Closed Routes

And so on. Yes, here we stop. For a contribution to *FACS FACTS* this ought to be enough.

8 Closing

This draft technical note represents work in progress.

These are the plans for further, leisurely technical development work: (1) To "complete" the above domain description estimated at adding a similar amount of material, perhaps 150% more. (2) To complement such a domain model with (typically) three diverse sets of requirements: narrated and formalised for example for (2.1) net maintenance (road maintenance, rail line maintenance); (2.2) logistics, that is, for freight transport planning and tracing; and (2.3) traffic services planning (timetabling, rostering, etc.). And then, finally, (3) to implement each of these three sets of requirements, fitting them on a common transportation net platform, arguing correctness of implementation, providing tests, proofs and model checking and to publish all this as a textbook for teaching software development in courses where students "copy" the ideas of the textbooks while applying them to entirely different infrastructure domains such as health care, the financial service industry, "the market" (including supply chains, etc.), manufacturing (of one kind or another), etc.

Anyone care to join?

9 Bibliographical Notes

References

[1] Dines Bjørner. *Software Engineering*, volume 1: Abstraction and Modelling, vol. 2: Specification of Systems and Languages, vol. 3: Domains, Requirements and Software Design of *Texts in Theoretical Computer Science, the EATCS Series*. Springer Verlag, 2005. Chapters 12 14 of vol.2 primarily authored by Christian Krog Madsen.

[2] Dines Bjørner. *An Example Railway Domain*, *FACS FACTS*, Issue 2005-2, pages 29-39, June 2005. Available to download from the BCS-FACS website: `http://www.bcs-facs.org/newsletter/facts200506.pdf`.

FACS FACTS Issue 2006-1

Call For Submissions

Deadline 17 February 2006

We welcome contributions for the next issue of *FACS FACTS*, in particular:

- Letters to the Editor
- Conference reports
- Reports on funded projects and initiatives
- Calls for papers
- Workshop announcements
- Seminar announcements
- Formal methods websites of interest
- Abstracts of PhD theses in the formal methods area
- Formal methods anecdotes
- Formal methods activities around the world
- Formal methods success stories
- News from formal methods-related organizations
- Experiences of using formal methods tools
- Novel applications of formal methods
- Technical articles
- Tutorials
- Book announcements
- Book reviews
- Adverts for upcoming conferences
- Job adverts
- Puzzles and light-hearted items

Please send your submissions (in Microsoft Word, LaTeX or plain text) to Paul Boca [editor@facsfacts.info], the Newsletter Editor, by **17 February 2006**.

If you would like to be an official *FACS FACTS* reporter or a guest columnist, please contact the Editor.

FAST2005: Formal Aspects of Security and Trust Workshop

Rob Delicata

Harry Potter and Formal Methods are terms which rarely enjoy lexical proximity. On the surface, at least, they seem positively antithetical — the unexplained phenomenon of magic being difficult to reconcile with the stolid precision of computer science. Certainly, the readers of Harry Potter are rather higher in number (though, one suspects, lower in brow) than those of the BCS-FACS newsletter, *FACS FACTS*. The intersection of the readerships is not disjoint, however, as shown by my headlong dive into Harry Potter and the Half-Blood Prince – sixth episode in the series – during a long journey to Newcastle (by train from London Kings Cross, but platform 3, not 9 ¾.)

In Harry Potter 6, the evil Lord Voldemort is waging war on the world, indiscriminately killing wizards and muggles (non-wizards) alike. Murders and misdeeds fill the pages of wizarding paper 'The Daily Prophet' and students at Hogwarts School of Witchcraft and Wizardry are in danger. Security at Hogwarts has been tightened: students are routinely checked for the possession of harmful magic, and all mail sent to the school is searched. Well, not quite *all* mail. Certain addresses in the nearby town of Hogsmeade are designated as *trusted*, and packages originating at these addresses are not examined. This trust is shown to be misplaced when one resident of a trusted address is placed under a mind-control spell (the *Imperius* curse) and forced to send a bottle of poisoned wine to the school. It is rather fortunate that the effects of this attack are not fatal, but it does serve to highlight a gaping security hole, and in doing so perfectly illustrates the important link between the concepts of security and trust.

As I sat reading, it struck me as rather appropriate that the book was released (apparently) to coincide with the Workshop on Formal Aspects in Security and Trust (FAST2005), held in Newcastle upon Tyne (and the reason for my train journey). The third edition of this international workshop was co-located with the Formal Methods Symposium (FM'05), 18–19 July 2005.

FAST2005 aimed to foster cooperation among researchers in the areas of security and trust. That there is a need for such cooperation is based on the premise that security and trust describe separate issues: the former seeks to *guarantee* that a given infrastructure is secure while the latter deals with the *perception* that it is secure. As security evangelist Bruce Schneier notes in his blog, inadequate security can be worse than no security at all. In the case of Harry Potter, the belief that mail was being checked engendered a (misappropriated) feeling of trust in the school's security. As a result, no one felt the need to take any further precautions. Hogwarts' mail-checking scheme would fail any reasonable formal model of either security or trust. Perhaps researchers in this area, muggles though they be, have something to teach the wizarding world.

Harry's absence at the conference was keenly felt, but offset by the presence of a small but diverse delegation from Britain, Ireland, mainland Europe, the United States and Australia. The program of 18 papers (16 full and

2 short) was selected by the program committee from 37 submissions and supported by two invited talks. The workshop was chaired by Theo Dimitrakos, Fabio Martinelli, Peter Ryan and Steve Schneider, with local organisation from Alessandro Falleni and Ilaria Matteucci.

The invited speakers addressed two very different aspects of security. Dr. Cédric Fournet from Microsoft Research discussed recent work carried out in Cambridge on formal tools for securing web services. In particular, he described TulaFale: a pi-calculus based specification language for writing machine-checkable descriptions of SOAP-based security protocols. Used in congress with Bruno Blanchet's ProVerif tool, the approach allows the automated verification of authentication and secrecy properties. The second invited speaker was Professor Brian Randell, Professor Emeritus at the University of Newcastle upon Tyne. He spoke about the socio-technical issues surrounding the neoteric field of electronic voting, particularly with regard to garnering public support for (and trust in) the technique. The conclusion, that an automated voting scheme could gain a degree of trust equivalent to that accorded to current paper ballots, was convincing. To my mind, however, there is an equally important need to address socio-economic roadblocks to public acceptance.

Presentations in the main workshop covered many of the central issues (and approaches) relevant to security and trust: information flow, trust management, access control and anonymity, and protocol analysis. One does not have to venture back too far to find a time when security conferences were dominated by papers on protocol analysis (one can even make a case for protocol verification being a 'killer app' for formal methods). In FAST2005, protocol analysis was confined to a single session. Perhaps this is symptomatic of a general shift of focus new topics; perhaps it is a testament to the significant research effort that brought the field to maturity. Either way, the first paper, "On the Formal Analyses of the Zhou Gollmann Non-repudiation Protocol" served as a timely reminder that the area is by no means complete. The presenter, Steve Schneider (neither of the paper's authors – Susan Pancho-Festin and Dieter Gollmann – could attend), argued that different analyses of the same protocol may yield different results, some guaranteeing correctness where others expose attacks. The discrepancy is explained by differing (and often implicit) assumptions made by those performing the analysis, highlighting the need for such underlying assumptions to be investigated further. Another presentation in this session – an interesting and fast-paced talk on the formal analysis of specification-based intrusion detection schemes – was as notable for its technical merit as for its attempt to shoehorn seventy slides into a twenty-five minute slot.

The session on Information flow dealt with such eclectic topics as the use of type-checking in eliminating implicit flows[1], the use of abstract interpretation in verifying secure information flows[2], and an investigation in the equivalence of the concept of opacity with anonymity and non-interference[3].

---

[1] "Eliminating Implicit Information Leaks by Transformational Typing and Unification", Boris Koepf and Heiko Mantel.

[2] "Abstract Interpretation to Check Secure Information Flow in programs with input-output security annotations", Nicoletta De Francesco and Luca Martini.

[3] "Opacity Generalised to Transition Systems", Jeremy Bryans, Maciej Koutny, Laurent Mazare and Peter Ryan.

Elsewhere, Konstantinos Chatzikokolakis delivered an impassioned talk on *probable innocence*, a new notion of probabilistic anonymity, and Hongbin Zhou described a logic for determining whether delegation schemes are able to withstand attempts at subterfuge. Of the two short papers accepted, only one was presented: Kun Wei ended the workshop much as it began, with a consideration of the Zhou-Gollmann non-repudiation protocol. The talk described the use of CSP stable-failures in reasoning about a timed version of the protocol, with the aid of model-checker FDR.

The full list of papers can be found at the FAST2005 website [http://www.iit.cnr.it/FAST2005], and post proceedings are planned with Springer's LNCS series.

Despite two no-shows, the workshop ran smoothly and generated interesting discussion between delegates. Any conference which brings together different communities is subject to a dangerous pitfall: that physical proximity does not always result in the mutually beneficent exchange of ideas. My feeling is that FAST2005 did achieve its goal. Let us hope that the dialogue continues.

Joining Other Societies and Groups

London Mathematical Society
http://www.lms.ac.uk/contact/membership.html

Formal Methods Europe
http://www.fmeurope.org/fme/member.htm

European Association for Theoretical Computer Science
http://www.eatcs.org/organization/membership.html#how_to_join

Association for Computing Machinery
https://campus.acm.org/Public/QuickJoin/interim.cfm

IEEE Computer Society
http://www.computer.org/join/

Back to Contents page

CALCO 2005: 1st Conference on Algebra and Coalgebra in Computer Science

José Fiadeiro

The first Conference on Algebra and Coalgebra in Computer Science (CALCO) was held at the University of Wales Swansea, 2–6 September 2005, with the co-sponsorship of BCS-FACS. The conference was preceded by a well-attended workshop dedicated to young researchers (CALCO-jnr) organized by Peter Mosses and John Power, and followed by a three-day meeting of IFIP WG1.3 (Foundations of System Specification) hosted by Peter Mosses.

A committee chaired by José Fiadeiro and Jan Rutten put together a very strong programme that reported both theoretical work on the mathematics of algebras and coalgebras and the way these results can support methods and

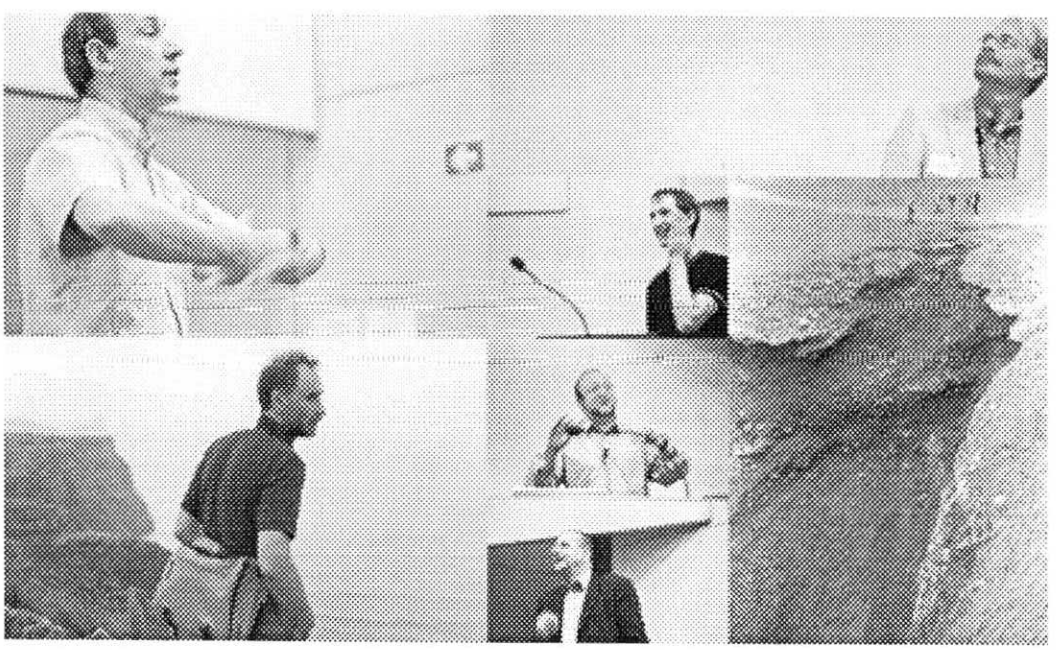
@PJL USTATUS JOB = ON
@PJL USTATUS PAGE = OFF
@PJL USTATUS DEVICE = ON
@PJL USTATUS TIMED = 0

CALCO 2005: 1st Conference on Algebra and Coalgebra in Computer Science

José Fiadeiro

The first Conference on Algebra and Coalgebra in Computer Science (CALCO) was held at the University of Wales Swansea, 2–6 September 2005, with the co-sponsorship of BCS-FACS. The conference was preceded by a well-attended workshop dedicated to young researchers (CALCO-jnr) organized by Peter Mosses and John Power, and followed by a three-day meeting of IFIP WG1.3 (Foundations of System Specification) hosted by Peter Mosses.

A committee chaired by José Fiadeiro and Jan Rutten put together a very strong programme that reported both theoretical work on the mathematics of algebras and coalgebras and the way these results can support methods and techniques for software development. The 25 papers, selected from 68 submissions, together with three invited talks by Samson Abramsky, Peter Mosses and Vladimiro Sassone, attracted 77 participants from all over the world. Neal Harman, Markus Roggenbach and Monika Seisenberger were highly praised for the enthusiasm and energy that they put into the organization of CALCO. With the help of a number of other members of the Department of Computer Science, they created a perfect environment in which ideas were freely exchanged and new collaborations initiated. Springer, through the publication of LNCS 3629, and Will Harwood, through a collection of superb photographs (see below), ensured that the memory of CALCO will not be lost. Hopefully, they will also stimulate participation in the next conference, which is scheduled to take place in Bergen, Norway, 2007. More details, including the photographs, can be found online at http://www.cs.swan.ac.uk/calco.



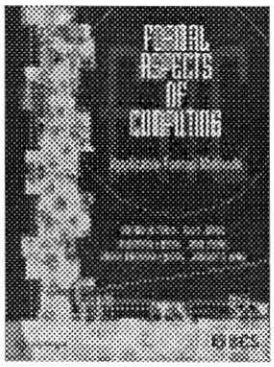
Formal Aspects of Computing

Applicable Formal Methods

ISSN: 0934-5043 (print version)
ISSN: 1433-299X (electronic version)
Journal no. 00165
Online version available | Online First articles available
2006 Volume 18 | 4 issues

Abstracted/Indexed in Current Contents

Founded by BCS-FACS



Formal Aspects of Computing publishes contributions at the junction of theory and practice. The objective is to disseminate applicable research. Thus new theoretical contributions are welcome where they are motivated by potential application; applications of existing formalisms are of interest if they show something novel about the approach or application.

The principal aim of this journal is to promote the growth of computing science, to show its relation to practice and to stimulate applications of apposite formalisms to practical problems. One significant challenge is to show how a range of formal models can be related to each other.

In particular, the scope of Formal Aspects of Computing includes, but is not limited to:
Well-founded notations for the description of systems | Verifiable design methods | Elucidation of fundamental computational concepts | Approaches to fault-tolerant design | Theorem-proving support | State-exploration tools | Formal underpinning of widely used notations and methods | Formal approaches to requirement analysis |

To order without BCS membership:
Americas: call (toll free) 1-800-SPRINGER | Fax (201) 348-4505 | Email orders-ny@springer.com |
ROW: call +49 (0) 6221-345-0 | Fax +49 (0) 6221-345-4229 | Email SDC-journals@springer.com |

Springer is part of Springer Science + Business Media
www.springer.com

Springer
the language of science

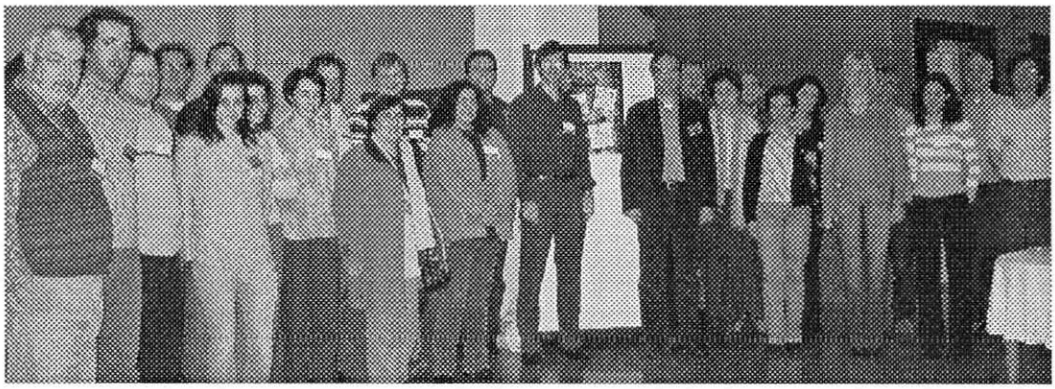
SEEFM 2005: 2nd South-East European Workshop in Formal Methods

George Eleftherakis

The second South-East European Workshop in Formal Methods took place in Ohrid on 18–19 November 2005. SEEFM05 was organized by SEERC and CITY College and locally organized by Sts. Cyril and Methodius University and hosted at Metropol hotel in Ohrid. The event was sponsored by BCS-FACS.

The workshop attracted participation from scientists, academics and researchers from Institutes and Universities all over the world, but especially from Europe. More specifically, out of the 17 papers accepted for presentation (10 full papers and 7 discussion papers), 3 of them were from Greece, 3 from Romania, 3 from Germany, 2 from France, and 1 from each of UK, Finland, Austria, Russia, Brazil, Algeria. We also had two invited papers from UK. In 8 papers, at least one author is associated with South-East Europe. The accepted papers were:

- Jari Veijalainen, Eleni Berki, Jari Lehmonen, Pasi Moisanen, "Implementing a New International Paper Mill Efficiency Standard - Using Computational Correctness Criteria to Model and Verify Timed Events"
- Maria Kourkouli, George Hassapis, "Application of the timed automata abstraction to the performance evaluation of the architecture of a bank on-line transaction processing system"
- Dimitris Dranidis, George Eleftherakis, Petros Kefalas, "Object-based language for generalized state machines"
- Christopher Thomson, Mike Holcombe, "Using a formal method to model software design in XP projects"
- Fevzi Belli, Christof J. Budnik, Axel Hollmann, "Formalization of Modeling, Analysis and Testing of Interactive Systems using Statecharts"
- Peter Massuthe, Wolfgang Reisig, Karsten Schmidt, "An Operating Guideline Aproach to the SOA"
- Richard Banach, Jean-Paul Bodeveix, Mamoun Filali, Michael Poppleton, "Dynamic aspects of retrenchments through temporal logic"
- Cristina Luca, "Context-free grammar for a two-dimensional language"
- Carlos Bazilio, Edward H. Haeusler, and Markus Endler, "Binding Network Topologies to Specifications via Pronouns"
- Victor V. Kuliamin, Vitaliy A. Omelchenko, Olga L. Petrenko, "Active Learning Facilitates Success of Formal Methods in Practice"



- Michael Barth, "A Formal Model for Performance Assessment in a Simulative Environment"
- Laura Ildiko Kovacs, Nikolaj Popov, Tudor Jebelean, "A Verification Environment for Imperative and Functional Programs in the Theorema system"
- Miloud Rached, Jean-Paul Bodeveix, Mamoun Filali, Odile Nasr, "A Timed B Method for Modelling Real Time Reactive Systems"
- Thouraya Bouabana-Tebibel, and Mounira Belmesk, "Object-oriented workflow formalization"
- Valentina Vujocevic, George Eleftherakis, "Improving Formal Methods' Tools Usability"
- Anca Vasilescu, "Algebraic model for the JK flip-flop behaviour"
- Oana Georgescu, "Problem Solving with Different Models"

Registration was free. There were 48 people registered for the workshop. The nationalities of the participants are as follows: FYRoM 15, Greece 8, UK 6, Romania 5, Germany 3, France 3, Algeria 1, Turkey 1, Hungary 1, Bulgaria 1, Serbia & Montenegro 1, Slovenia 1, Russia 1, Brazil 1 and others.

The event began with welcome speeches from Dr.G.Eleftherakis co-chair of SEEFM05 and Dr.P.Ketikidis, Vice-Principal of CITY College and Director of SEERC. Both expressed their satisfaction about the number of participants and their willingness to find ways of promoting and organizing similar workshops in the future as well as establishing links between academics in this research area in South-East European states.

Professor Jonathan Bowen (pictured left) from London South Bank University and Professor John Derrick from the University of Sheffield were the invited speakers. Bowen (BCS-FACS Chair) re-examined the original ten important requirements (or "commandments") for formal developers to consider and follow, based on knowledge of several industrial application success stories, considering their validity in the light of a further decade of industrial best practice and experiences.

Derrick (pictured right) described recent work which seeks to apply model-checking techniques to the verification of Erlang code.

There were three sessions that followed. Sessions A and C were presentations of full papers and session B was a presentation of discussion papers. Session A was more on applied formal methods. Sessions B and C were on theoretical foundations, verification and tools. All papers were very interesting and raised stimulating discussions among the participants. Each presentation of a full paper lasted for 20 minutes allowing 5 minutes for questions and each presentation of a discussion paper lasted for 10 minutes allowing 10 minutes of questions and discussion.

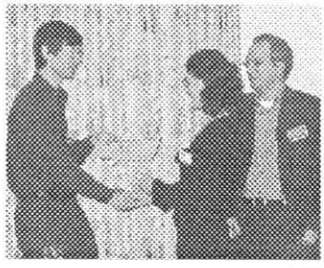
At the end, there was a panel discussion, which also involved questioning from the participants, chaired by Dr. G. Eleftherakis, on the general topic: "Formal Methods, Practical dimensions: Challenges in the Business World". The panel consisted of Professor J. Bowen (UK), Professor J. Derrick (UK), Professor W. Reisig (Germany) and Dr E. Berki (Finland). The outcomes of this discussion are summarized as follows:

- Formal methods is a hot research area but there is still no evidence of wide acceptance.
- Industry needs to be convinced about the importance of using Formal Methods.
- Participants from the industry should be invited to express their opinions.
- Researchers from South-East Europe should join the existing formal methods organizations in Europe.
- Education on Formal Methods should be provided to students who will identify the future needs of the industry.
- The particular characteristics of people in SEE could be identified and due to their strong mathematical background, there might be a chance there to use these people in disseminating the practice on Formal Methods.
- The workshop provided a good chance to establish strong links between academics in SEE countries and those links should be exploited.

SEEFM05 was once again a satellite workshop of the Balkan Conference in Informatics (2nd). The proceedings of the workshop were published as a CD, available on request by emailing seefm05_secretariat@city.academic.gr. The proceedings will be published by SEERC as an edited volume under the title: "Formal Methods: Challenges in the business world" .

A number of selected papers will be peer reviewed once more and will be included in a post-proceedings volume in the forthcoming issue of the International Journal *Annals of Mathematics, Computing and Teleinformatics* (AMCT) [http://journals.teilar.gr/amct].

Best Paper / Student presentation prizes, sponsored by BCS-FACS, each consisting of one year's FACS membership and one year's subscription to the *Formal Aspects of Computing* journal, were awarded based on a voting procedure between all the participants. The best paper presentation was awarded to Professor Wolfgang Reisig (pictured left) for the paper "An Operating Guideline Aproach to the SOA" authored by Peter Massuthe, Wolfgang Reisig, Karsten Schmidt. The best student presentation prize went to Ms. Laura I.Kovacs (pictured right) for the paper "A Verification Environment for Imperative and Functional Programs in the Theorema system" authored by Laura Ildiko Kovacs, Nikolaj Popov and Tudor Jebelean.



In addition, BCS-FACS sponsored a student bursary of £150 to help with attendance and presentation at SEEFM by a researcher based in the UK. The researcher was Christopher Thomson, a PhD student from the Computer Science department of the University of Sheffield, who presented the paper "Using a formal method to model software design in XP projects", authored by Christopher Thomson and Mike Holcombe.

The workshop included a gala-dinner in a traditional restaurant in Ohrid offered to all the participants of SEEFM05.

Conclusions

- The workshop was the second attempt to bring people from SEE together, based on their common interest in Formal Methods.
- All participants expressed the opinion that the workshop is to be continued on a biannual basis.
- It is thought that the workshop could alternate between Thessaloniki and other interested parties (e.g., Timisoara, Romania).
- The Steering Committee could be extended keeping it still small and flexible.
- A web page [http://www.seefm.info] containing all links and email addresses of people in SEE should be constructed as a means to disseminate information and practices. The page should initially contain information collected at the two workshops, e.g. people, institutions, research groups, interests, projects, tools, courses etc.
- Common projects should be sought between various institutions in SEE with common research interests in Formal Methods.
- Every effort should be made to involve people from industry. ▒

Back to Contents page

THE UNIVERSITY of York

1st International Symposium on Concurrency, Real-Time, and Distribution in Eiffel-like Languages (CORDiE'06)

http://www.cs.york.ac.uk/~paige/cordie06.htm

King's Manor, York, United Kingdom, 4-5 July 2006

The Eiffel method provides a pure, object-oriented language for specifying and implementing systems. It is widely regarded as one of the cleanest object-oriented languages available, in terms of the thoroughness of the specification of its underlying semantics, its tool support for engineering and analysing systems, and in the principles used for its design. Recently, interest has been focused on proposals for extending Eiffel to concurrent, real-time, and distributed systems. At the forefront of these proposals is SCOOP (Simple Concurrent Object-Oriented Programming), a concurrency mechanism that adds a single keyword, **separate**, to the language. These proposals have in turn attempted to address real-time and distributed programming extensions. The proposals pose a number of challenging research questions, both theoretical and practical. Practical questions include efficient design and implementation of all aspects of the proposals while allowing extension to real-time and multi-processor distribution. Theoretical questions include how to extend reasoning for Eiffel to support concurrency, real-time and distribution.

This is the first event to focus specifically on concurrency, real-time and distribution extension for Eiffel-like languages. To this end, original research contributions are sought in all areas related to extending Eiffel and Eiffel-like languages to concurrent, real-time, and distributed systems development. Topics of interest include, but are not limited to:

- Semantics of models of concurrency in Eiffel-like languages.
- Implementations of models of concurrency in Eiffel and Eiffel-like languages.
- Real-time and Distributed extensions to Eiffel-like languages, including handling security and exceptions.
- Reasoning about concurrent programs in Eiffel-like languages.
- Case studies in implementing concurrent systems in Eiffel-like languages.
- Empirical/quantitative comparisons of concurrency models in Eiffel-like languages
- Comparisons of concurrency in Eiffel with other languages, e.g., Java, Ada, C++.

The symposium is also a venue for papers that seek to relate novel concurrency mechanisms from other languages where there is a clear flavour of formality, e.g., Java/JML, Spec#, etc.

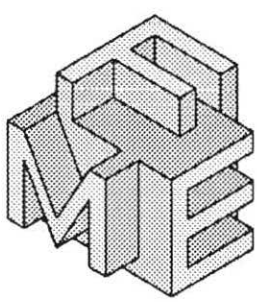
Important Dates and Organisation

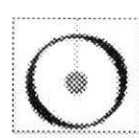
- Submission of Papers: 16 February 2006
- Notification to Authors: 15 March 2006
- Symposium: 4-5 July 2006

Programme Committee:

Simon Dobson (UC Dublin)
Jin Song Dong (NUS Singapore)
Chris Gill (Washington, USA)
Michael Gonzalez Harbour (Cantabria, Spain)
Jeremy Jacob (York, UK)
Jeff Magee (Imperial College, UK)
Bertrand Meyer (ETHZ Switzerland)
Jan Vitek (Purdue, USA)
Jonathan Ostroff (York, Canada)
Piotr Nienaltowski (ETHZ, Switzerland)
Alan Wood (York, UK)
Jim Woodcock (York, UK)
Phil Brooke (Teesside, Co-chair)
Richard Paige (York, Co-chair)

Sponsors and Supporters





ICFEM 2005: 7th International Conference on Formal Engineering Methods

Richard Banach

The seventh International Conference on Formal Engineering Methods (ICFEM 2005) was held in Manchester this year on 2–4 November 2005. Actually ICFEM 2005 is the seventh in a series of conferences, beginning in Hiroshima in 1997, and continuing since then, making approximately equal visits to the Far East and the West. In this time, it has become established as one of the major forums for the presentation of new ideas for the application of formal techniques to the engineering of real systems. Currently, the challenge is to encourage general acceptance of these methods within industry as a part of the development of high quality systems.

This year ICFEM accepted 30 technical papers from 74 submitted; in addition there were three invited talks. Amongst many fascinating presentations throughout the conference, ranging over topics as diverse as development, communications, specification, testing, verification and security, some particular favourites of the author were those on revealing "unclarities" in the semantics of UML 2.0, model checking real communications protocols using "sweeplines", and using Stalmarck's algorithm to prove inequalities.

A good flavour of the essence of a conference like ICFEM 2005 can best be conveyed by briefly describing the contributions of our keynote speakers:

- Anthony Hall (Independent Consultant, UK)
  Realising the Benefits of Formal Methods

  Hall gave a provocative talk in which he posed the question: "what have formal methods ever done for us?" He rightly pointed out that we need to address the risks and costs inherent in using formal methods as well as those that occur when such methods are not used. There was also an interesting section on the sociological and pedagogical issues surrounding the widespread uptake of formal methods by industry.

  This talk was sponsored by BCS-FACS.

- Egon Börger (University of Pisa, Italy)
  A Compositional Framework for Service Interaction Patterns and Interaction Flows

  Börger presented an abstract state machine intended for use in transaction processing. His particular focus is on business process management in multi-party collaborative environments. There are eight basic interaction patterns in the model which can then be composed to perform more complex tasks. Inherent in the system being presented is a model for rigorous execution/platform independent analysis, with a particular contribution being a benchmarking for web services functionality testing.

  This talk was sponsored by Microsoft Research.

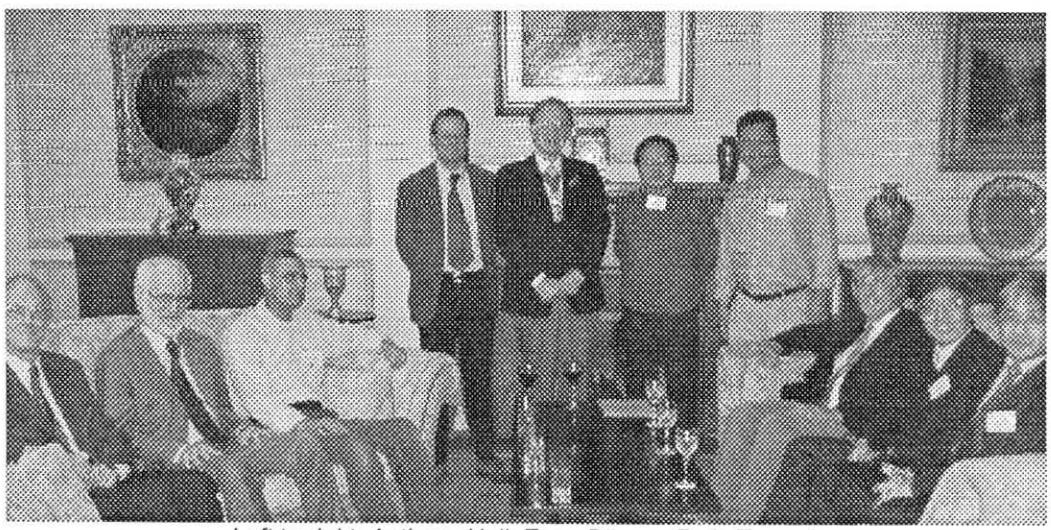
- John Rushby (SRI, USA)
  An Evidential Tool Bus

  In his talk, Rushby presented thoughts about the future direction that might be taken by FM tools. What he outlined was a possible method to integrate many different tools into a unified tool platform for doing formal methods. One intriguing example would be an architecture in which Mathematica (or Maple) was used in conjunction with Isabelle-HOL (or PVS) to verify that the computer algebra system had correctly solved a problem, by checking its solution using a theorem prover to validate the results.

  This talk was sponsored by FME.

The technical part of the conference was supplemented with a social programme. This began with a reception in the Fossil Gallery of the Manchester Museum, which contains a rather dramatic cast of the full skeleton of a Tyrannosaurus Rex. This and the other exhibits provided a stimulating backdrop for informal conversations amongst the delegates. During the reception, the University's Vice-President for External Affairs warmly welcomed the participants. The other component of the social programme was the Conference Banquet. This was held in the splendid surroundings of the Banqueting Room of Manchester Town Hall, a venue enjoyed by all the delegates. The banquet itself was preceded by a Civic Reception hosted by the Deputy Lord Mayor of Manchester, at which he welcomed delegates to the City of Manchester in general and to the Town Hall in particular.

The usual fade-out sensation typically experienced at the end of a conference was jolted somewhat due to the coincidence of the last part of the conference with the Muslim Eid festival, which caused a temporary shortage of taxi drivers. Fortunately, the in depth local knowledge of the staff at Hulme Hall (venue for the conference) meant that enough taxi resources could be located, in order that all those with trains or planes to catch after the conference ended could be accommodated. ▒

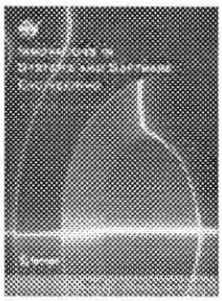


Left to right: Anthony Hall, Egon Börger, Faris Taweel, Richard Banach, James Ashley, Kung-Kiu Lau, Jin Song Dong, John Rushby, Shaoying Liu, Kenji Taguchi

Springer www.springer.com

Innovations in Systems & Software Engineering

A NASA Journal



Journal no. 11334
ISSN: 1614-5046 (print version)
ISSN: 1614-5054 (electronic version)
Online version available
Online First articles available
2006 Volume 2 | 4 issues

Innovations in Systems and Software Engineering: A NASA Journal will address issues and innovations in Systems Engineering, Systems Integration, Software Engineering, Software Development and other related areas that are specifically of interest to NASA. The journal will include peer-reviewed world-class technical papers on topics of research, development and practice related to NASA's missions and projects, topics of interest to NASA for future use, and topics describing problem areas for NASA together with potential solutions. Papers that do not address issues related to NASA are of course very welcome, provided that they address topics that NASA might like to consider for the future.

Papers are solicited from NASA and government employees, contractors, NASA-supported academic and industrial partners, and non-NASA-supported academics and industrialists both in the USA and worldwide. The journal will include updates on NASA innovations, articles on NASA initiatives, papers looking at educational activities, and a State-of-the-Art section that will give an overview of specific topic areas in a comprehensive format written by an expert in the field.

To order:

Americas:
Phone: (toll free) 1-800-SPRINGER
Fax (201) 348-4505
Email orders-ny@springer.com

ROW:
Phone: +49 (0) 6221-345-0
Fax +49 (0) 6221-345-4229
Email SDC-journals@springer.com

Springer is part of Springer Science + Business Media
www.springer.com

Springer
the language of science

RefineNet Workshop at ICFEM 2005

John Derrick

A lively refinement workshop was held at the ICFEM meeting in Manchester (see page 47) at the end of October. Technical contributions were given by Thai Son Hoang, David Streader, Joe Morris, Mike Poppleton, Steve Schneider, Georg Struth, Colin Snook, Heike Wehrheim and John Derrick. The invited talk was by Jean Raymond Abrial on *Using Formally Defined Design Patterns to Improve System Developments*, which was enjoyed by a packed audience. Thanks go to the ICFEM organizers for the local arrangements.

The next RefineNet meeting is scheduled for January/February; see http://www.refinenet.org.uk for more details.

SPECIAL OFFERS FOR FACS MEMBERS

- To celebrate the **18th birthday** of the ***Formal Aspects of Computing* (FAC) journal**, FACS members renewing their membership, or joining the first time, will receive **FREE online access to the FAC journal until the end of December 2006**.
- In addition, FACS members can subscribe to the **paper copy** of Vol 18 of the FAC journal for **£5**. This special price of £5 is **valid until the end of February 2006**. From 1 March 2006, the price for the paper copy will be £48. So return your membership forms with payment without delay!!
- FACS members will receive **30% discount on Springer titles**. To claim your discount, please contact Springer directly by email on journalslondon@springer-sbm.com . Please mention that you are a paid-up FACS member. Springer will contact the FACS Membership Secretary to confirm your membership status before processing any orders.
- FACS members can **subscribe to the ISSE journal (see advert on page 49) for £30**
- FACS members can **subscribe to the requirements engineering journal (see advert on page 52) for £30**

If you have any questions about these offers, please contact Paul Boca [paul.boca@virgin.net]

Back to Contents page

IFM 2005: 5th International Conference on Integrated Formal Methods

Judi Romijn

The fifth conference on Integrated Formal Methods (IFM2005) was held from 29 November to 2 December 2005, at the Technische Universiteit Eindhoven, the Netherlands.

The main conference was preceded by a invited tutorial by Holger Hermanns on the combination of statecharts and stochastic analysis. In the afternoon, we had a doctoral symposium with no fewer than 13 presentations. The audience was a nice size, with about 30 people. The day ended with a welcome reception (sponsored by IPA) with a celebration of the Dutch traditional Sinterklaas event: All present received a chocolate initial of their first name.

On Wednesday 30 November, the main conference started with the FME-sponsored talk by David Parnas who gave the conference a nice provocative start. This was followed by seven of the 19 regular presentations (see the IFM website for details [http://www.win.tue.nl/ifm/]). In the afternoon we had an excursion to the Van Abbe Museum for modern art, with a guided tour past rooms with early 20th century work placed next to pieces from the inter-bellum. Afterwards we had the conference dinner in the restaurant of the museum, with a view of a truck loaded with soccer balls in the midst of a pond.

On Thursday and Friday, we had the well-received invited presentations by Doron Peled and Patrice Godefroid, and regular presentations ranging from rather theoretical perspectives on event systems with fairness to the application of formal methods to the hardware domain.

On Friday, we closed the conference with the traditional election of the best regular/student presentations, awarding the winners with a year's membership of BCS-FACS and a one-year subscription to the *Formal Aspects of Computing* journal. The best student presentation prize was won by Pontus Boström (Åbo Akademi University, Finland), and the best regular presentation prize was once again won by Steve Schneider (University of Surrey, UK).

It was an enjoyable and inspiring conference!

The next IFM conference will be hosted by Oxford in June/July 2007 and chaired by Jim Davies.

Archiving *FACS FACTS*

If you have any old issues of *FACS FACTS* or *FACS Europe*, please contact Paul Boca [Paul.Boca@virgin.net]. A list of the issues to be archived can be found on the BCS-FACS website [http://www.bcs-facs.org/newsletter/facsfactsarchive.html].

Back to Contents page

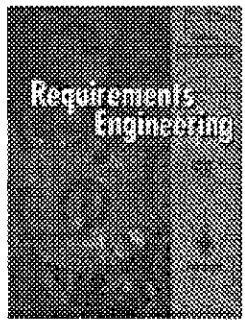
Springer

www.springer.com

Requirements Engineering

ISSN: 0947-3602 (print version)
ISSN: 1432-010X (electronic version)
Journal no. 00766
Online version available
Online First articles available
2006 Volume 11 | 4 issues

Abstracted/Indexed in Current Contents



Requirements Engineering provides a focus for the dissemination of new results about the elicitation, representation and validation of requirements of software intensive information systems or applications. Theoretical and applied submissions are welcome, but all papers must explicitly address:

- the practical consequences of the ideas for the design of complex systems
- how the ideas should be evaluated by the reflective practitioner

The journal is motivated by a multi-disciplinary view that considers requirements not only in terms of software components specification but also in terms of activities for their elicitation, representation and agreement, carried out within an organisational and social context. To this end, contributions are sought from fields such as software engineering, information systems, occupational sociology, cognitive and organisational psychology, human-computer interaction, computer-supported cooperative work, linguistics and philosophy for work addressing specifically requirements engineering issues.

To order:

Americas: Phone: (toll free) 1-800-SPRINGER
Fax (201) 348-4505
Email orders-ny@springer.com

ROW: Phone: +49 (0) 6221-345-0
Fax +49 (0) 6221-345-4229
Email SDC-journals@springer.com

News from BCS: Specialist Groups' Assembly, 25 October 2005, London

Margaret West, BCS Liaison Officer for FACS

The recent Specialist Groups' Assembly was held at the BCS London Offices, Southampton Street, on 25 October 2005. The opening remarks and first talks concerned the increase in BCS membership (and hence increase of revenue) and the forthcoming 50th anniversary of the BCS (2007). A 3-year plan had the objective of ensuring that by 2008 an IT professional would be as highly regarded as any other professional. Plans for the Golden Jubilee would include National, Regional and Local events. The targeted audience would include Government, Companies (Senior Management etc.) and the Education Sector – the latter to involve primary and secondary schools as well as tertiary education.

The rest of the morning session involved some aspect of "professionalism". The first speaker on the topic remarked on the continuing problems with IT projects — for example that 75% exceed budgets and schedule. He highlighted the issue with an example: Jaguar recalled 55,000 vehicles because of software problems. The original item was from the Daily Telegraph (17 April 2004). However the following is from

http://www.accidentreconstruction.com/news/apr04

"All are equipped with six-speed ZF automatic transmissions and the spokesman said the problem involved an electronic module that controls the gearbox.

"In a rare event of major loss of oil pressure in the transmission, it can result in the transmission selecting reverse gear," he said."

Another speaker asserted that in order to promote professionalism "Best Practice" should be more widely known and accepted and that the cooperation of other professions is required. In the question and answer session which followed I asked if any of the speakers were aware of the "best practices" or "guidelines" adopted for the development of the software for the problematic Jaguar cars. I asked this as I was indirectly associated with a set of guidelines for automotive software produced by the Motor Industry Software Reliability Association (MISRA). The guidelines recommend formal notations for the specification of high integrity automotive systems. The original article did not mention any guidelines – although I checked the MISRA website [http://www.misra.org.uk] – and Jaguar do belong to the association.

The afternoon session of the assembly consisted of workshops — the one I attended involved discussion about event promotion and improvement. During the day there was also a brief presentation about the new BCS website, which should be completed next year. The new web site does not yet affect specialist groups.

Back to Contents page

Conference Announcements

The following are sponsored by BCS-FACS and/or considered of special interest to BCS-FACS members:

January 2006

POPL 2006 – 33rd Annual Symposium on Principles of Programming Languages
11–13 January
Charleston, USA
http://www.cs.princeton.edu/~dpw/popl/06

February 2006

UITP 06 – 1st International Symposium on Unifying Theories of Programming
5–7 February
Walworth Castle, County Durham
http://www.scm.tees.ac.uk/utpsymposium

March 2006

CMCS 2006 - 8th International Workshop on Coalgebraic Methods in Computer Science
25–27 March
Vienna, Austria
http://conferences.inf.ed.ac.uk/cmcs06/cmcs06.html

ETAPS 2006: European Conference on Theory And Practice Of Software
25 March – 2 April
Vienna, Austria
http://www.complang.tuwien.ac.at/etaps06

TACAS 2005 – Tools and Algorithms for the Construction and Analysis of Systems
25 March – 2 April
Vienna, Austria
http://depend.cs.uni-sb.de/index.php?id=329

MBT 2006 – 2nd Workshop on Model Based Testing
25–26 March
Vienna, Austria
http://react.cs.uni-sb.de/mbt2006

Back to Contents page

March 2006

FASSE 2006 – Fundamental Approaches to Software Engineering
27–29 March
Vienna, Austria
http://www.elet.polimi.it/conferences/fase06

FOSSACS – Foundations of Software Science and Computation Structures
27–31 March
Vienna, Austria
http://fossacs06.ru.is

SPIN 2006 – 13th International SPIN Workshop on Model Checking of Software
30 March – 1 April
Vienna, Austria
http://www.cs.tut.fi/SPIN2006

April 2006

BCTCS 2006 – 22nd British Colloquium for Theoretical Computer Science
4–7 April
Swansea, UK
http://www.cs.swan.ac.uk/BCTCS2006

June 2006

WADT 2006 - 18th International Workshop on Algebraic Development
1-3 June
Submission: 15 April
La Roche en Ardenne, Belgium
http://www.info.fundp.ac.be/~pys/WADT06

DisCoTec 2006 – Distributed Computing Techniques
13 – 16 June
Submission: 10 January (abstract), 17 January(paper)
Bologna, Italy
http://www.discotec06.cs.unibo.it/satellite.htm

CiE 2006 - Computability in Europe 2006. Logical Approaches to Computational Barriers.
30 June – 5 July
Submission: 6 January
Swansea, Wales
http://www.cs.swan.ac.uk/cie06

Back to Contents page

August 2006

MFCSIT 2006 – 4th Irish Conference on Mathematical Foundations of Computer Science and Information Technology
1–5 August
Submission: 17 April
Cork, Ireland
http://www.ucc.ie/info-mfcsit

FM2006 – Formal Methods 2006
21–27 August
Submission: 24 February
Ontario, Canada
http://fm06.mcmaster.ca

October 2006

ICFEM 2006 – 8th International Conference on Formal Engineering Methods
30 October – 3 November
Submission: 12 May
Macao, China
http://www.iist.unu.edu/icfem06

November 2006

ICTAC 2006 – 3rd International Colloquium on Theoretical Aspects of Computing
20–24 November
Submission: 1 May 2006
Gammart/Tunis, Tunisa
http://www.iist.unu.edu/ICTAC2006

December 2006

BCS-FACS Christmas Meeting
TBA
London, UK
http://www.bcs-facs.org/events/xmas2006.html

For further conference announcements, please visit the **Formal Methods Europe** website [http://www.fmeurope.org], the **EATCS** website [http://www.eatcs.org] and the **Virtual Library Formal Methods** website [http://vl.fmnet.info/meetings].

Back to Contents page

PhD Abstracts

Name	Kelly Androutsopoulos
Thesis Title	Specification and Verification of Reactive Systems Using RSDS
Supervisor	Dr. Kevin Lano
Institute	King's College London
Examiners	Dr. Krysia Broda and Dr. Michael Poppleton
Awarded	April 2005
URL	http://www.dcs.kcl.ac.uk/pg/kelly/publications.html
Keywords	RSDS, statemachines, model checking, translation, proof of correctness, reactive systems

Formal methods have been applied to reactive systems in order to capture errors early on in the development life-cycle and reduce redesign costs. The Reactive Systems Development Support (RSDS) method provides support for the analysis and design of reactive systems and generates code from these specifications. An RSDS system is specified by a set of invariants, a set of statemachines and a Data Control Flow Diagram (DCFD), which are then verified using the B theorem-prover. B however requires user interaction and is not capable of proving temporal properties easily. This thesis extends RSDS by integrating model checking so that temporal properties can be verified. The model checker used is the Symbolic Model Verifier (SMV).

There are two distinct semantic views of statemachines in RSDS: the coarse-grain and the fine-grain, with the key difference between them being the granularity of a step. We describe a translation to SMV for each semantic view and we guarantee the quality of the translations by formally proving their correctness. This proof is a vital part in our provision of transparent formal method support for system design. To overcome the state explosion problem of model checking, we propose some natural ways of using the RSDS decomposition techniques for dividing the system into subsystems; these can then be model checked independently as separate SMV programs. We have tested our translations with various case studies.

RSDS/UML is an object-oriented version of RSDS that uses a restricted subset of UML for specification. It aims to bridge the gap between formal methods and mainstream software development techniques. For the same reasons as with RSDS, we integrate model checking with RSDS/UML by defining a translation for the coarse-grain and proving its correctness. The properties verified can reason over the dynamic instantiation of classes. The translation is illustrated on the gas burner system.

Back to Contents page

BCS FACS

BCS-FACS / FME Evening Seminar

Formal Methods in the Last 25 Years

Jean-Raymond Abrial, ETH Zurich
Ian Hayes, University of Queensland
Cliff Jones, University of Newcastle
John Tucker, University of Wales

30 January 2006

Start time: 5.30 pm

Refreshments from 5pm

BCS London Offices
First Floor
The Davidson Building
5 Southampton Street
London WC2E 7HA

Mathematically-based "formal" methods for developing software and systems have had an interesting history. Over the past twenty-five years, the subject has moved from controversies surrounding code verification, through work on data types, design methodology, refinement and "Lightweight" Formal Methods, to automated proof and model-checking technology.

This event brings together four computer scientists who have been active as leading researchers and practitioners in the field over the last quarter century. It provides an opportunity to learn about the motivations behind some of the major developments in the field, to discuss trends, fashions, successes and failures and set them in their recent historical context. The meeting will be of interest to researchers, students and practitioners in software and systems development, specialists in formal methods and anyone with an interest in the history of computing.

The Panel:

Jean-Raymond Abrial, ETH Zurich
Ian Hayes, University of Queensland
Cliff Jones, University of Newcastle upon Tyne
John Tucker, University of Wales, Swansea

Chairman: John Fitzgerald, Formal Methods Europe (FME)

There will be an opportunity for participants to raise issues for discussion at the event. In order to make best use of the time, participants are invited to email John Fitzgerald [John.Fitzgerald@ncl.ac.uk] with issues or questions that they would like the panel to discuss. It may not be possible to deal with all the issues raised, but there will be an opportunity for additional questioning and discussion at the event.

Refreshments will be served from 5pm

The seminar is free of charge and open to everyone. If you would like to attend, please email Paul Boca [Paul.Boca@virgin.net] by **26 January 2006.**

Pre-registration is required, as security at the BCS Offices is tight.

Location of the venue

http://www.bcs.org/NR/rdonlyres/B5872B38-3FBB-46E8-9CE7-6F43212E1198/0/londonss.jpg

FME website

http://www.fmeurope.org

FACS website

http://www.bcs-facs.org

FACS Evening Seminars website

http://www.bcs-facs.org/events/EveningSeminars

Back to Contents page

F.X. Reid Answers all your Problems

[*Editors' note: Despite his insistence of not becoming an 'agony aunt', we persisted, aided by information received from Birmingham of his partiality for Mouton Cadet. At great expense (id est, a crate of the stuff from Oddbins), we elicited the following copy. The editors take no responsibility for what follows. Honest!*]

Dear Auntie Reid,

I am a senior academic at a provincial university, which is being forced to lay people off because of the drop in income from foreign students. As an admissions tutor for such students, who has not published the requisite 20 papers in top journals for over a year, I feel myself to be particularly vulnerable. I am now trying for voluntary retirement on the principle of 'jump before you're pushed'.

The trouble is that I enjoy doing research and I feel that I still have a lot to offer. What should I do?

Name and Address Supplied.

Dear N & A S.

I know who you are and don't call me 'auntie', you worm! What should you do? You can start by returning my copy of the *Tractatus*, pay the coffee club and stop playing Solitaire every hour G*d sends. As for your research, I'd rather not comment.

On a more helpful note, I suggest that you apply for an R.A. post somewhere. Preferably, far away from me.

Alternatively, you could adopt the peripatetic *modus vivendum* of Paul Erdos. Don't be surprised, however, if, on approaching a colleague with the announcement that your mind is open, to receive the reply 'well this door isn't, so push off'.

FXR

Dear Auntie Reid,

I am a radical socialist and a specialist in non-interleaving parallelism who has fallen in love with his professor, a wonderful woman who does a mean lasagne.

My worry is that she is a devout believer in CSP. I tried telling her that the trace/failure/divergence semantics is a Gothic monstrosity, designed merely to

ensure full abstraction for a congruence based on a set of dodgy equations (you know the score: surjective initial algebra semantics → normal form → full abstraction) but she just laughs and hits me with volume three of *A Handbook of Logic in Computer Science*.

I am at my wits end. What can I do?

Tod Grudge, Rummage

Dear Tod,

Don't keep calling me 'auntie'!

How interesting that a 'radical socialist', as you so alarmingly describe yourself, has not made a beeline for *Kapital* or the *Communist Manifesto*. Or perhaps you have.

I fully sympathise with your intellectual predicament. Interleavers are, after all, sad cases who have not been weaned off formal language theory and λ-calculus, while the equational reasoning approach, although elegant, is ultimately sterile.

As for your emotional problems, what can I say? My own success with the female sex is, of course, legendary. (I would always recommend a bottle of *Mouton Cadet*, but as a radical socialist, you're probably a real-ale-and-a-packet-of-crisps man.) I suggest that you become an expert in CSP (if your stomach is strong enough) and then formulate your observations in the form of theorems. If expressed subtly enough, you should be able to logically demonstrate the 'Gothic monstrosity' of the semantics. Subversion in this form should be more than acceptable to a person of your political persuasion.

FXR

Dear Auntie Reid,

I am an attractive 19 year old female research student with big ideas [*Editors' note: the editors reserve the right to substitute alternative words where appropriate. We reserved the right here!]* working on neural networks. My supervisor, Dr. de Sade, says that my work is good, but keeps inviting me back to his home at the *Chateau la Coste* for what he calls an 'in depth exploration of back propagation'.

I'm not sure what to do. He tells me that he can 'get me through it' – I suppose he means my PhD – but I don't like the way he leers when he says it. Please help.

Lolita Lovebody

Dear Lolita,

For the last time, will you people stop calling me 'auntie'!

I am profoundly shocked by your letter. For an attractive 19 year old female, particularly one with big "ideas" to be subjected to neural networks, of all things, causes my gorge to rise. I suggest that you try something more consistent with the delicate feminine mind, say complexity theory.

As for your supervisor, this is the old, old story; an ambitious young PhD student strutting her stuff before a moribund valetudinarian.

Wear a longer skirt, put your hair in a bun and express a fervent interest in Per Martin Löf intutionistic type theory. If that doesn't work, tell him that his adherence to the Widrow-Hoff rule makes it impossible for you to work together.

Incidentally, do you like *Mouton Cadet*?

FXR

Dear Auntie Reid,

I'm at the end of my tether!

FXH, Oxford

Dear FXH

I am not your bloody auntie!! Get a life!

FXR

CiE 2006

Computability in Europe 2006:
Logical Approaches to Computational Barriers

30 June – 5 July 2006
Swansea University

http://www.cs.swansea.ac.uk/cie06

cie06@swansea.ac.uk

Call for Grant Applications

Deadline: 31 March 2006

A number of grants are available for attending CiE 2006. They are mainly intended for supporting UK based PhD students and for participants from the former Soviet Union. Also, student members of the ASL may apply for travel funds. For more details see our website.

BCS FACS

FACS membership application/renewal (2006)

Title (Prof/Dr/Mr/Ms) _____ **First name** _____ **Last name**_____

Email address (required for options * below)_____

BCS membership No. (or sister society name + membership number)

_____

Address _____

_____

_____

Postcode _____ **Country** _____

I would like to take out **membership to FACS** at the following rate:

- ☐ £15 (Previous member of BCS-FACS now retired, unwaged or a student)
- ☐ £15 (Member of BCS or sister society with web/email access)*
- ☐ £30 (Non-member or member of BCS or sister society without web/email access)

ALL MEMBERS WILL RECEIVE FREE ELECTRONIC ACCESS TO THE *FORMAL ASPECTS OF COMPUTING* JOURNAL UNTIL THE END OF DECEMBER 2006

I would like to subscribe to **Volume 18 of the FAC journal** (paper copy) at the following rate:

- ☐ £5 THIS OFFER IS VALID UNTIL THE END OF FEBRUARY 2006. FROM 1 MARCH 2006, THE PRICE WILL BE £48

The total amount payable to BCS-FACS in **pounds sterling** is **£ 15 / 20 / 30 / 35** (delete as appropriate). I am paying by:

- ☐ Cheque made payable to **BCS-FACS (in pounds sterling)**
- ☐ Credit card via PayPal (instructions can be found on the BCS-FACS website)
- ☐ Direct transfer (in **pounds sterling**) to:

  Bank: Lloyds TSB Bank, Langham Place, London
  Sort Code: 30-94-87
  Account Number: 00173977
  Title of Account: BCS-FACS

If a receipt is required, please tick here ☐ and **enclose** a stamped self-addressed envelope.

Please send completed forms to:

Dr Paul P Boca
PO BOX 32173
LONDON N4 4YP
UK

For FACS use only

Received by FACS	Date:	Initials:
Sent to Springer	Date:	Initials
Actioned by Springer	Date:	Initials

FACS Committee

Jonathan Bowen
FACS Chair
ZUG Liaison

Jawed Siddiqi
Treasurer

Paul Boca
Secretary and
Newsletter Editor

Roger Carsley
Minutes
Secretary

John Cooke
FAC Journal
Liaison

Ali Abdallah
Events Coordinator

John Fitzgerald
FME Liaison
SCSC Liaison

Margaret West
BCS Liaison

Rick Thomas
LMS Liaison

Judith Carlton
Industrial Liaison

Kevin Lano
UML Liaison

Rob Hierons
Chair, FM and
Testing Subgroup

FACS is always interested to hear from its members and keen to recruit additional Committee members. Presently we have vacancies for officers to handle publicity and help with fund raising, and to liaise with other specialist groups such as the Requirements Engineering group and the European Association for Theoretical Computer Science (EATCS). If you are interested in helping the Committee, please contact the FACS Chair, Professor Jonathan Bowen, at the contact points below:

BCS FACS
c/o Professor Jonathan Bowen (Chair)
London South Bank University
Faculty of BCIM
Borough Road
London SE1 0AA
United Kingdom

T +44 (0)20 7815 7462
F +44 (0)20 7815 7793
E info@bcs-facs.org.uk
W www.bcs-facs.org

You can also contact the other Committee members via this email address.

Please feel free to discuss any ideas you have for FACS or voice any opinions openly on the FACS mailing list [FACS@jiscmail.ac.uk]. You can also use this list to pose questions and to make contact with other members working in your area. Note: only FACS members can post to the list; archives are accessible to everyone at http://www.jiscmail.ac.uk/lists/facs.html .

Coming Soon in FACS FACTS....

TRain Column

Conference reports

Report on FM05 Industry Day

Details of upcoming FACS Evening Seminars

Report on FM05 Grand Challenges Workshop

And More...

Back to Contents page