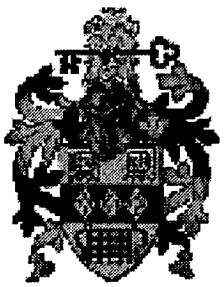
FORMAL METHODS
EUROPE

FACS Europe



BCS
FACS

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

Series I Vol. 3, No. 3, Winter 1997 ISSN 1361-3103

1 Editorial

Happy '97, everybody...Here is the Xmas workshop special edition; many thanks to the Requirements Engineering SIG for co-hosting what everyone I spoke to agreed was an interesting and thought-provoking event, despite some late withdrawals (thanks to the 'flu season, mainly) leading to a more London-based event than originally intended.

Perhaps the most significant outcome of this workshop was that as a result of discussions there about the practical difficulties of obtaining material for research, Praxis plc have obtained permission for the output of one of their projects to be made available to bona fide academic researchers. Publication of work referring to this material will need to be authorized by Praxis, but permission will not be witheld unreasonably. The material consists of a substantial semi-formal requirements specification, including traceability

relations, and supporting material. For further information, please contact Rosamund Rawlings, *rmr@praxis.co.uk.*

Is there anyone else out there who is willing to make a substantial piece of real work available for academic analysis by a wider community of researchers?

1.1 From the FACS committee...

Dr. Ian Maung, Department of Computer Science, University of Warwick, Coventry CV4 7AL. email: *im@dcs.warwick.ac.uk*

Home page:*http://www.dcs.warwick.ac.uk/ im/*

I am a mathematician originally, having done a PhD in Mathematical Logic in Manchester. After that, I spent a very enjoyable 3.5 years in Brighton becoming a computer scientist, and doing research in formal object-oriented methods with Richard Mitchell and Dan Simpson.

For the last 1.5 years, I have been a lecturer in Computer Science at Warwick, where I teach formal methods (Z), software engineering, object technology and C++ to vast numbers of (mostly) very bright (but not too conscientious) students, which leaves me with precious little time for research (boo-hoo), most of which is spent doing PhD supervision and ESPRIT project management.

At the last committee meeting, I was given the honour of constructing and maintaining a distribution network for FACS publicity. So I would like to ask you all to let me know of any new or established channels that we could use so I can pool them together and improve our publicity.

1.2 Formalities

Contributions express the opinions of contributors, not of FACS, FME or any other organization with which they are associated (unless they say otherwise!). Letters are welcome and should be sent to the Editor.

Advertisements are welcome, as full or half page printed ads, or as inserts (i.e. loose sheets or booklets mailed with the Newsletter). Advertisements and inserts will only be accepted where they are clearly of specific interest to the FACS/FME community. Please contact the editor for current rates and due dates for copy. We normally publish four issues a year, but cannot guarantee specific dates.

2 FACS Events

2.1 Convergence of System Theories, 7-8 April 1997

This is now confirmed as a joint event of BCS-FACS and the IEE Professional Group M5 (Systems Engineering). It will be an informal workshop to bring together several research communities which use and combine theories which are intended to describe parts or aspects of an application or domain. Full details will be available as soon as practicable via the FACS website. We are still waiting for final confirmation of our venue, so please accept our apologies for lack of fuller information at this point.

If you are interested in taking part, please contact one of the organisers:
John Boarder, *100064.1533@CompuServe.COM*
Alan Wills, *alan@trireme.com*
Ann Wrightson, *a.m.wrightson@hud.ac.uk*

2.2 Northern Formal Methods Workshop, Ilkley, 14-15 July 1997

This workshop follows on from the highly successful event held for the first time in 1996 in Ilkley, West Yorkshire.

The workshop is intended to be an open forum for the exchange of ideas and results between researchers involved in the theory and application of formal methods. For example, papers in the 1996 workshop covered areas such as specification methods, concurrency, real-time, methods integration, process algebras, experience reports from industrial trials, hardware, and the teaching of formal methods.

The programme will consist of invited talks by three leading figures in the use and application of formal methods, contributed papers, and discussion groups.

You should find a full call for papers as a loose insert in this newsletter. If someone else got it first, then contact Andy Evans, *a.s.evans@comp.brad.ac.uk*, or David Duke, *duke@minster.york.ac.uk*.

2.3 Specification Patterns

FACS/FME have been invited to join in with the Specification Patterns group's meeting on *Patterns in discrete event modelling* on Wed. June 18 at 6.30 pm at the University of Westminster.

Graham Berrisford (group co-ordinator) says:

> This event is about drawing event models to specify the 'business objects' that synchronise concurrent information objects—the key issue in combining federated systems or distributing a single one; Separating concurrency from communication; Simple patterns in object interaction analysis.
>
> This is as formal a topic as we get, whether it is formal enough for your group should be a matter for debate on the day.

I hope we can muster a (friendly!) presence for this, since this is an area where formal techniques are potentially practically useful for supporting business-oriented computing—not traditionally a strong area for formality—through rigorous representations of reusable patterns.

Further details from Roger Carsley, *roger@westminster.ac.uk.*

2.4 BCS-FACS Xmas Workshop 1997

This will happen, somewhere in London, at the usual time. Please contact John Cooke for further information.

2.5 FACS Anniversary

FACS will be 20 years old on 30 Nov 1997 (if you count the semi-official inaugural meeting) or 16 May 1998 (if you take the first meeting date we were 'official' with respect to the BCS). The Autumn issue of the newsletter will be a special 'FACS at 20' edition. Serious and no-so-serious pieces will be welcome, from members and friends old and new (we already have a promised contribution from our old friend and patron F.X.Reid).

3 FME events

3.1 FMEIndSem - Formal Methods Europe Industrial Seminars

This roadshow has now started; full details available from Tim Denvir, or see *http://www.ifad.dk/projects/fmeindsem.html.*

3.2 FME '97

Formal Methods: Their Industrial Applications and Strengthened Foundations, Graz, Austria, 15-19 September 1997. See insert in last issue for full details, or FME website.

3.3 FME Information Resources

This project, sponsored by the European Commission under ESSI - European Systems and Software Initiative, Software Best Practice - is developing on-line databases of Formal Methods tools, applications, frequently asked questions. The databases are being expanded during the course of the project.

http://www.cs.tcd.ie/FME/

3.4 FMEGuides

This project, sponsored by the European Commission under ESSI - European Systems and Software Initiative, Software Best Practice - is developing multimedia management guide books accessible through servers. It is intended to enable managers to read success-stories about the use of formal methods. The material will include videos, magazine articles, and images and video sequences usable by TV producers.

http://demain.cgs.fr/formal/

4 (A few) Other Events

4.1 Problem Solving with Computer Systems Projects

A half-day colloquium on 12 March, on spotting and countering failure in systems projects, hosted by the IEE Computing and Control Division. One of a number of events which can be found on the IEE website, *http://iee.org.uk/*

4.2 TAPSOFT is dead...long live ETAPS

TAPSOFT 97, April 14-18 1997, is to be the last in its current format of a two-part meeting consisting of CAAP (Colloquium on Trees in Algebra and Programming) and FASE (Colloquium on Formal Approaches in Software Engineering).

Full details are available on *http://www.lifl.fr/tapsoft97*, or by email from *Anne-Cecile.Caron@univ-lille1.fr.*

The successor conference will be known as ETAPS (European joint conferences on Theory and Practice of Software), and will run in early Spring from 1998.

4.3 CADE-14

The 14th International Conference on Automated Deduction July 13-17, 1997, Townsville, Australia.

CADE is the major forum for presentation of research in all aspects of automated deduction. Logics of interest include propositional, first order, equational, higher order, classical, intuitionistic, constructive, type theory, nonstandard, and meta-logics. Methods of interest include resolution, paramodulation, unification, term rewriting, tableaux, constraints, decision procedures, induction, interactive systems, and frameworks. Applications of interest include hardware and software development, systems verification, artificial intelligence, logic, set theory, mathematics, applicative programming, and logic programming. Special topics of interest include proof translation, human-computer interfaces, distributed deduction, and search heuristics.

Full details and an electronic registration form are on the WWW:
http://www.cs.jcu.edu.au/~cade-14/

5 Electronic Information Sources

5.1 Formal Methods Applications Database

In the context of the project FMEInfRes (Formal Methods Europe Information Resources, partially funded by the EC under the ESSI programme, project number 21375) a database has been set up with descriptions of industrial applications of formal methods.

If you know of any applications of formal methods/specification languages worth considering for inclusion in the database (i.e. you have been involved yourself or you know someone who was involved with such an application) then it would be appreciated if you would send details to (e-mail preferred): Nico Plat, Cap Volmac, P.O. Box 2575, 3500 GN Utrecht, The Netherlands. Fax: +31-30-2522234. E-mail: Nico.Plat@ACM.org (please use the format requested; this was given in the last issue of this newsletter, and is also on the website).

5.2 Rigorous Object-oriented Development

A repository on rigorous methods for object-oriented development is under construction at Manchester University Department of Computer Science. *http://www.cs.man.ac.uk/rood/*. Small so far, but looks well worth keeping an eye on if you have an interest in such things.

5.3 Computer-related Incidents

One of the best reads I know on the Web is Peter Ladkin's website on computer-related incidents and accidents in civil and military aviation. Lots of thought-provoking reflections, good information, and links to related papers worldwide. *http://www.rvs.uni-bielefeld.de/~ladkin/Incidents/*

6 Xmas Workshop 1996. Formal Methods and Requirements Engineering: Challenges and Synergies

Report by Sara Jones and David Till, with contributions by the speakers.

The aim of this workshop was to encourage communication and understanding between academics and practitioners in the fields of requirements engineering and formal methods, and to identify areas where work in each field might contribute to the other. The workshop was jointly organised by RESG (Requirements Engineering Specialist Group) and FACS (Formal Aspects of Computing Science), both of which are BCS special interest groups.

Around 70 people attended the workshop to hear a distinguished panel of invited speakers presenting Requirements Engineering (RE) and Formal Methods (FM) perspectives on:

- Change in Software and System Requirements and Specifications
- Requirements Traceability
- Requirements Elicitation and Validation
- Non-functional Requirements
- Inconsistency in Software/System Requirements and Specifications
- Use of Multiple Notations
- Architecture
- Domain Knowledge

There were also group discussion sessions on each day in which delegates were divided into smaller groups to discuss issues arising from the presentations, as well as more specific questions about what FM could deliver for RE practitioners, and *vice versa*. In the final plenary session, delegates were invited to put any further questions directly to the speakers.

Michael Jackson began by **Setting the Scene**. His own abstract for the talk summarises best the main points:

> The essence of formality is reasoning with symbols without considering what the symbols stand for. The essence of informality is the unboundedness of the set of considerations that may prove significant. This unboundedness can invalidate otherwise sound logic applied to an informal domain.
>
> Machines are formal. Requirements are informal. To deal with requirements we must explore the problem context far from the machine. We formalise requirements to understand how a formal machine behaviour at the interface affects the world. This formalisation is necessarily imperfect. We must minimise the approximation error by appropriate choices.

One can say the same thing in different ways. When we write software we are specifiying some very definite behaviour on the part of the machine, and whether we like it or not the software system is a formal system. On the other hand, the real world is immensely complex and we cannot expect to describe it completely as a formal system. But unless we do formalise some properties of the real world which are of interest for the system concerned to a sufficient degree of accuracy there is no way that we can hope to demonstrate that the formal system we specify and build will, in combination with the relevant part of the real world, result in those desirable properties being realised.

The first RE/FM pair of speakers were Ken Eason and John Wordsworth who discussed aspects of **Change in Software and System Requirements and Specification**. Ken Eason set the scene here by presenting a case study scenario in which a major bank wanted help in specifying a large change programme. He pointed out that stakeholders do not have ready-made requirements at the beginning of a project, but generate them over time. Support for this generation of requirements must therefore be provided as, for example, in the ORDIT method, where increasingly realistic forms of scenario (involving, for example, static paper-based constructions, role play, or trial implementations) are used to assist clients in understanding possible solutions to their problems. He ended his presentation with a challenge to the formal methods community: could formal methods help in managing on-going changes in requirements through the course of a project?

John Wordsworth's claim was that formal methods encourage engineers to circumvent changes to requirements by promoting thorough exploration of requirements before they are agreed between client and supplier. He saw animation and execution of formal specifications as being important tools in

allowing clients to see the effects of a proposed system function in advance of development and claimed that formal methods in general were helpful in allowing project managers to assess the risk of changes to system requirements or functions as well as the effort required to make the changes. On a more cautionary note, however, he noted that the use of formal methods demands substantial intellectual investment, and that system specifiers are therefore likely to resist change once a specification has been developed.

Laurence James spoke about **Requirements Traceability** and addressed what he calls the higher-level principles of requirements traceability: closure (connections between requirements all the way through to delivered system); comtinuity (consistent information flow between different project viewpoints); logic (solutions developed are correct); emotion (this seems to be the requirements engineering version of the feel-good factor!) – "traceability is emotion backed up by logic". At a more pragmatic level, he emphasised the importance within an organisation of knowing what are its requirements for a traceability tool, and especially understanding the likely impact of the use of such a tool on their development process. He also claimed that traceability tools can support the necessary bridges between informal requirements and formal specifications of components and their properties. He said that "logic and traceability are the two faces of requirements"; another way to express this is to say that formal languages can be used to specify precisely and unambiguosly what is needed, while traces help to answer questions about why such precisely described compoents are needed and how they relate to the informally expressed requirements. Francisco Pinheiro showed how the requirements for traceability tools can themselves be formally defined, but pointed out that any such formal framework should not be too prescriptive because traceability needs should be guided by necessity rather than predefined structure. He introduced the TOOR approach to traceability as an example of an appropriately formal attack on the problem.

The topic considered after lunch was that of **Requirements Elicitation and Validation**. Lee McCluskey had the challenge of presenting a formal methods perspective on these issues. He began with some observations regarding the different sets of jargon used by those in the RE and FM communities. Although there is some overlap, with languages such as RML straddling the jargon divide to a certain extent, he noted that there are apparently also many differences in the concerns of the two communities, judging from keywords used in conference proceedings of the respective communities.

Lee's talk then focused on the validation of formal requirements specifications which he felt was an important and little researched area. He suggested that the design of formal specification languages should perhaps be influenced

to a greater extent by the need to validate specifications, and by the characteristics of particular applications or domains. While validation of formal specifications is still so difficult, no single method is likely to be sufficient on its own, particularly in the development of safety critical applications where reliable validation procedures are crucial. The solution he proposed was to combine a number of different methods. He ended his talk by describing an approach which he had developed in projects funded by the UK Civil Aviation Authority and other bodies, and which involved combining manual checking with automated syntax and type checking, simulation or animation of specifications, and formal proofs.

John Dobson's talk adopted a wider perspective and presented a common framework for model building which was intended to show how sets of vocabulary items and composition rules for different representations relate to common conceptual models and architectural principles according to which a system is to be built. His framework consisted of five levels:

- the natural language level, at which requirements are articulated
- the conceptual level, at which the concerns of a client are framed and alternative models of requirements are explored
- the logical level, where appropriate calculi are identified and formal models are defined
- the design level, where real-world interpretations of operations described in the calculi are defined
- and finally, the descriptive level, at which the implications of these interpretations are explored

The final session of the first day was devoted to so-called **Non-Functional Requirements**. Ian Sommerville began the session by declaring himself as someone who didn't believe in the existence of non-functional requirements as a kind of requirements which is fundamentally different from functional requirements. In fact, he claimed, assuming such a distinction could be made had been largely responsible for the lack of uptake of academic formal methods in industry. This was because formal methods research had focused exclusively on functional requirements, ignoring their inextricable links with non-functional requirements, while industry had to be very concerned with so-called non-functional aspects of the systems they produced. The solution he proposed was to forget trying to make any distinction between functional and non-functional requirements and to focus research on developing formal notations which are 'usable by real people working on real systems'. The

emphasis should be on sending industry a simple message about wha techniques and notations to use under what circumstances, and on providing good support, for example, in the form of tools, for notations which are already available.

David Robertson also discussed the fact that the term 'non-functional' can, at best, be only loosely defined. However, taking a definition encompassing ideas such as 'aspects of a system which don't influence its mechanics', 'descriptions of problems, not all aspects of which may influence directly the systems we build to solve them', and 'various other things to do with higher levels of design', he was able to point to several examples in which formal methods had been used to help reveal argument and model structures, and to assist in developing formal interpretations of knowledge whose structure can not be easily accessed. In general, however, his feeling was that the use of formal methods in dealing with non-functional requirements was still at a 'pre-engineering' stage, so that examples of good practice are scarce, and reliable guidelines for their use in this context are non-existent.

After the two presentations, there was some discussion of the idea that it may not always be desirable to formalise non-functional requirements relating to, for example, the political implications of a new system. However one delegate described how her organisation had used FUSION to specify at least some non-functional requirements relating to users and stakeholders along with requirements relating more directly to the system required.

The two speakers on **Inconsistency in Software/System Requirements and Specifications** were Alfonso Fuggetta and Tony Hunter. Alfonso Fuggetta pointed out that though we ultimately would like a consistent and complete specification of the problem to be solved, the process of reaching such a happy state of affairs is likely to pass through intermediate states in which our knowledge is partial and even inconsistent. This is inevitable and indeed necessary because the detection of inconsistency is one of the major driving forces to discover more salient information about the domain and the problem to be solved within that domain. Of course if we want to define what we mean by inconsistency in a given context we will have to formulate formal consistency rules or rely on the inference mechanisms of a logical language. However it is important to realise that often we will not detect this kind of formal inconsistency becuase our formal descriptions do not take account of enough of the significant properties of the domain. It is also possible that inconsistency will be detected as a result of different usage of the same terms within different viewpoints, when in fact there is no real underlying conflict. Tony Hunter also illustrated how inconsistency can be a spur to action; in his examples, upon further investigation it transpires that there were unstated assumptions. He also described how we can reason in

the presence of inconsistency by using quasi-classical logic. We do not want to eliminate inconsistency; we want to manage it. In particular, we may want to investigate the sources of inconsistency, we may want to ignore it or delay its resolution. If we do in the end resolve it, it is important to keep a trace of it, since this will remind us of what conflicts had to be addressed, and indeed who was involved in the decisions which led to its resolution.

Tom Maibaum and Jose Fiadeiro have worked together over many years on the **Use of Multiple Notations**. Tom Maibaum took up enthusiastically the role of the Requirements Engineer looking for help from the Formal Methodist. He motivated the use of multiple notations by different views (stakeholders) in order to reduce complexity and to allow for independent and natural expression of different aspects of a problem to be solved. His specific challenge to formal methods was:

> How can we relate descriptions in different formalisms with possibly different structuring principles and almost certainly different structures?

Jose Fiadeiro recast the problem:

> How can we relate and integrate different notations so that we can understand the whole, detect interferences and support incremental development and evolution?

Many authors have explored the idea of translating notations into a common style of predicate logic and then integrating on the basis of a common semantic domain. Maibaum and Fiadeiro have proposed as an alternative the use of category theory as a convenient mathematical framework for interaction: this allows what they call 'exoskeletal' representation of interaction (the explicit linkage of entities in different views which must be treated as identical) and translation between structuring principles.

The **Software Architecture** theme was addressed by Jeff Kramer and Susan Eisenbach. In his abstract, Jeff Kramer says that software architecture is in the solution domain, while requirements are in the problem domain, though requirements themselves often do describe an architecture of sorts, perhaps best represented by Michael Jackson's problem frames. He also suggests that it can be very difficult to state or to understand requirements clearly until some feasible architecture has been contemplated. Thus, in many cases, the cycle of activities most likely to produce better understanding of how high-level system goals may be operationalised is to push forward towards design of a feasible implementation based on well-tried software architectures and then to step back again, having achieved greater insight into

what is possible and the kind of behavioural characteristics to be expected from such an implementation. Of course, this assumes that the software architectures concerned are such that certain important behavioural characteristics – for example, performance measures – can be inferred without knowing the details of the plug-in, application-specific modules which will be needed. Thus Jeff was concerned to point out the utility of software architecture within the requirements process; Susan Eisenbach went on to show how such architectures can be formally described, using the π-calculus and logic, illustrating with a simple example.

The final session of the workshop was devoted to the role of **Domain Knowledge** in requirements specification. Neil Maiden's talk described the work of the European NATURE project on determining the way in which domain knowledge could enable reuse of information about common user requirements or system contexts. However, he suggested there was still a need for firmer theoretical foundations for defining the notion of a domain. Jeremy Holland described his work on a project in which a formal representation of a particular domain of activity (that of clinical activity in a London hospital) was used in the development of new information systems for the hospital. Based on his experience of that project, he felt that the existence of a formal domain model had assisted in reasoning about the domain, and hence about the system to be developed, exposing contradictions and inconsistencies in information given to him by system stakeholders, and had led to greater control in designing the system through progressive refinement of the model. He concluded that the process had been time-consuming, but that the time had been well spent.

It is of course more difficult to summarise what emerged from the **discussion groups** since they were intended to be more open-ended. Here are some of the questions and points which arose during the discussions on the first day of the meeting:

- has the focus of formal methods been too narrow?

- it was suggested that a pragmatic view might be: informal, structured methods give some rapid initial results; formal models should then be built, but kept secret from the client (though the analyst should be able to read from the model into the client'w world and terminology); formal proofs only done in mission or safety critical areas

- is requirements capture/analysis/definition an engineering disciplne? engineering implies process with feedback, starting and stopping criteria

- does RE ever stop? can we avoid thinking about solutions?
- what are the skills needed by a Requirements Engineer?
- what are the building blocks for RE?
- the main benefit of FM for RE is in posing questions
- we need a better understanding of what RE problems are
- we need to be able to classify problems to see which techniques may help
- it is hard to talk about problem space as distinct from solution space
- the concerns of FM and RE are not keeping pace with changes in the real world; we need to look more at re-engineering and product families
- how can we achieve better synergy of academic expertise and the needs of industry? controlled case study experiments, improved metrics, cost/benefit analysis in different areas
- how do we know we are making progress? how can we measure the quality of requirements? how can we compare methods? how do we select the right combination of techniques?

On the second day, discussion groups were asked to consider what formal methods could deliver to requirements engineering practitioners in the next 1, 5 or 10 years, and vice versa.

In answer to the first question (What can FM deliver for RE?), it was suggested that:

- FM might deliver a basis for problem and solution classifications, perhaps within 1 or 5 years, which would then enable a mapping between appropriate problem and solution patterns.
- within 5 years, FM may be able to deliver better natural language paraphrasing of formal specifications to facilitate validation, and more usable formal methods!
- within 10 years it was hoped that FM might be able to assist in problems of feature interaction, and that techniques for theorem proving might also be made more usable

It was also pointed out that different formal methods were likely to have different impacts in different types of system development. For example, it was thought that formal methods would have different impacts in the development of control, business information and highly interactive systems, with safety critical and control systems being likely to benefit most quickly, and business information systems being unlikely to benefit within the timeframe considered.

The second question (What can RE deliver for FM?) elicited the following suggestions:

- a better understanding of approaches to negotiation which could be used in discussing trade-offs between different formal models
- a basis for formal modelling of impact, sensitivity and consequence analysis regarding changes to specifications
- a better way of maintaining links between a formal specification and the 'real world'

A need for greater co-operation between the two communities of FM and RE in terms of transferring skills and techniques was also identified by three of the four groups.

In conclusion, our impression was that all the participants felt they had heard something useful and stimulating during the two days of the workshop. Of course it would be unrealistic to expect a meeting such as this to achieve major breakthroughs in an area as challenging as this one, but it must be of benefit to the members of the two overlapping communities present at the workshop if they understand better each other's concerns.

7 FACS Co-ordinates

7.1 FACS Central

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

FACS Officers

Chairman	John Cooke	D.J.Cooke@lboro.ac.uk
Treasurer	Roger Stone	R.G.Stone@lboro.ac.uk
Committee Secretary	Roger Carsley	roger@westminster.ac.uk
Membership Secretary	John Cooke	D.J.Cooke@lboro.ac.uk
Newsletter Editor	Ann Wrightson	a.wrightson@hud.ac.uk
Liaison with BCS	Margaret West	m.m.west@hud.ac.uk
Liaison with FME	Tim Denvir	t-denvir@dircon.co.uk
Publicity Co-ordinator	Ian Maung	im@dcs.warwick.ac.uk

Contributions to the Newsletter on any relevant topic are welcome. Please send them by email if possible, in LaTeX or plain text, to the Editor.

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