Formal Methods: The National Physical Laboratory’s Experiences

Keith Lines
Data Science Department
National Physical Laboratory, UK

BCS, FACS meeting 06/04/2021    Draft v3 06/04/2021
1. Aims
2. About NPL / Data Science Department
3. Past work (small sample):
   Milne and Strachey / OSI / Survey / TraCIM
4. Current work (very small sample):
   Joint Appointments / Model-Based Systems Engineering
5. Conclusions
Aims of the Presentation

• Present overview of some of NPL’s experiences with formal methods
• Stimulate some discussion: E.g. lessons from the past relevant today
• Set the scene for further talks
Aims of the Presentation

• Make clear NPL DS still interested in formal methods / functional programming / theoretical computer science
• But now via universities, e.g. joint appointments, research excellence grants, PhD students
• Engagement with BCS FACS is key (hence this presentation)
1. Aims

2. About NPL / Data Science Department

3. Past work (small sample):
   Milne and Strachey / OSI / Survey / TraCIM

4. Current work (very small sample):
   Joint Appointments / Model-Based Systems Engineering

5. Conclusions
This presentation is…

- An overview: Not going into depth
- A quick tour through some selected projects…
- …ending with some current work

This presentation is not…

- An introduction to formal methods
  - Will assume at least a basic knowledge
- A complete picture
About NPL

The UK’s National Metrology Institute (NMI)

- Approximately 900 staff
- Approximately 200 visiting researchers
- Main laboratory in Teddington, London
- Regional hubs at Cambridge, Glasgow, Huddersfield
Pilot ACE 1946

Packet-switching developed at NPL 1966
Aim: Confidence in the intelligent & effective use of data

- Mix of mathematicians, computer scientists, statisticians and physicists. Includes secondee staff from other NPL departments.
- Approximately 40 staff across three sites, including joint appointments with Cambridge, Surrey and Strathclyde.
- Approximately 12 students (PhDs, sandwich courses).
- Extensive collaboration: Can’t do data science without data.
  - **Internal:** work with most other departments at NPL.
  - **Fellow NMIs** worldwide.
  - **External companies:** collaborations and consultancy.
  - **Academia:** CDT engagement, grant-funded projects.
  - **Other** establishments & industry bodies: UK & worldwide.
• **Machine learning**: innovative work on robustness and uncertainty quantification, introductory guide to ML methods and associated training.

• **Reliable software & algorithms**: Trustworthiness

• **Large-scale inference**, uncertainty quantification and complex data processing chains

• **Image analysis**: Quantification, feature extraction & data fusion.

• **Time series analysis**: Tipping point & trend extraction.
NPL: DS: Informatics

• Development of data models to integrate measurement data with device calibration data
• Ontology-based information modelling for sustainable data storage
• Automated data annotation to implement FAIR principles (Findable, Accessible, Interoperable and Reproducible)
• Definition of minimum metadata standards initial focus on imaging in healthcare and life sciences
Contents

1. Aims
2. About NPL / Data Science Department
3. Past work (small sample):
   Milne and Strachey / OSI / Survey / TraCIM
4. Current work (very small sample):
   Joint Appointments / Model-Based Systems Engineering
5. Conclusions
A theory of programming language semantics
a single work in two parts

First published 1976
by Chapman and Hall Ltd.,
11 New Fetter Lane, London EC4P 4EE

© 1976 Robert Milne
Printed in Great Britain by
Whitstable Litho Ltd., Whitstable, Kent

ISBN 0 412 14260 0

I am very grateful to the people mentioned above and to the Science Research Council and the National Physical Laboratory, which gave me financial support. Naturally the defects of this book are my own responsibility; what I regard as some of them should occasionally become evident from the tone of my remarks.

Robert Milne
“(1990) The Division (DITC) saw its chief role as giving technical support for the Department’s (DTI) policy of promoting quality of products and procedures in IT…” [1]

• Data Security, e.g.:
  • Message Authenticator Algorithm (MAA) [3]: Formal specifications in VDM, Z and LOTOS [4]
  • Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT): Formal specification in VDM-SL of reference implementation [5]

• Communications Protocols, e.g.:
  • OSI: Distributed Transaction Processing [6]
• Work summarized in an NPL report, published October 1995 [9]
• “…highlights many of the problems of describing a real-world protocol.”
• Team of three worked at NPL for three years on the LOTOS description
• Delegated to NPL by the ISO committee responsible after initial work elsewhere
• “As the work progressed… increasing numbers of problems were found with the English text”
• Report continued…

• “By far the most all-pervading problem with LOTOS was the awkwardness of the data typing language …every datatype has to be defined from scratch, there are no short cuts”

• C.F. From ACT-ONE to Miranda, a Translation Experiment Charles, Bowman, Thompson, University of Kent (1997) [16]

• “…benefits from producing a formal description of a standard as it was being produced …experts on-hand who could explain… the protocol …writers of the English text… could correct their specification when questions and comments from NPL revealed genuine problems”
• Report continued…

• “…diplomatic form of words. In many cases ambiguities can not be reflected in the formal description, this is, after all, one of the advantages of using formal methods.”

• **Tools:**

  • “…description was so large that it defeated all LOTOS tools that it was given to.”

  • “Most tools failed to read in the entire… description and none made any progress in trying to animate…”

  • “…only verification possible… by manual comparison with corresponding English text.”
Published 31 March 1993

In 1992 a literature search and survey of industry conducted to discover reasons for low acceptance (of formal methods)

Getting on for 30 years later (!!!!), would a similar survey produce very different results?
Consists of four sections:

- **Part I:** Overview: Survey of Formal Methods in Software Engineering (summary)
- **Part II:** Survey of Formal Methods in Software Engineering (details)
- **Part III:** Survey of Formal Methods in Higher Education
- **Part IV:** Benefits, Limitations and Barriers to Formal Methods
Formal Methods: A Survey

Part I / Part II: Formal Methods in Software Engineering

800 questionnaires sent out, 444 returned, 104 analysed
All questionnaires read

<table>
<thead>
<tr>
<th>Questionnaires</th>
<th>Received</th>
<th>Analysed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post (UK)</td>
<td>385</td>
<td>104</td>
</tr>
<tr>
<td>Electronic (UK)</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Post (non-UK)</td>
<td>48</td>
<td>3</td>
</tr>
<tr>
<td>Electronic (non-UK)</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Discarded</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>444</strong></td>
<td><strong>126</strong></td>
</tr>
</tbody>
</table>
Which formal methods have you considered using?

<table>
<thead>
<tr>
<th>Method</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>55 %</td>
</tr>
<tr>
<td>VDM</td>
<td>55 %</td>
</tr>
<tr>
<td>LOTOS, CSP, CCS</td>
<td>18 %</td>
</tr>
<tr>
<td>None</td>
<td>11 %</td>
</tr>
<tr>
<td>OBJ</td>
<td>7 %</td>
</tr>
<tr>
<td>RAISE</td>
<td>5 %</td>
</tr>
<tr>
<td>Temporal Logics</td>
<td>4 %</td>
</tr>
<tr>
<td>Others</td>
<td>24%</td>
</tr>
</tbody>
</table>
In what way would you consider using formal methods (specification, refinement, proving etc.)?

<table>
<thead>
<tr>
<th>Use</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification</td>
<td>89 %</td>
</tr>
<tr>
<td>Proofs</td>
<td>39 %</td>
</tr>
<tr>
<td>Refinement</td>
<td>17 %</td>
</tr>
<tr>
<td>Design</td>
<td>10 %</td>
</tr>
<tr>
<td>Any way</td>
<td>5 %</td>
</tr>
<tr>
<td>Verification</td>
<td>5 %</td>
</tr>
<tr>
<td>Requirements capture</td>
<td>5 %</td>
</tr>
<tr>
<td>Others</td>
<td>11 %</td>
</tr>
</tbody>
</table>

Not 100 %
What do you consider the benefits of using formal methods are?

- It clarifies requirements 48 %
- It removes ambiguities 40 %
- Removal of errors earlier on in project, savings costs 24 %
- Prove properties 19 %
- Easier to build software because you know about it.
- Prove relations between program and specification
- Basis of discussion with the client
What do you consider the limitations of using formal methods are?

- Specification is not readable by the clients 23 %
- Some aspects of specification difficult to define in a mathematical model: e.g. timing constraints, HCI 21 %
- Specification will not model all aspects of the real world 19%
- Lack of experienced staff 18 %
- Development costs increased 15 %
- Mistakes can be made in the specification 14 %
What do you consider the barriers to the use of formal methods are?

- **Tools are not available 43 %**
- Increased costs 24 %
- Training needed which costly and take time 18 %
- Lack of trained staff 18 %
- Difficult to use 17 %
- No objective case for commercial benefits 15 %
- Formal methods are not mature enough 15 %
Formal Methods: A Survey

Part I / Part II: Formal Methods in Software Engineering

Do you have any suggestions on how to overcome these barriers?

• Education. Universities to teach formal methods  41%
• Case studies  29%
• Tools: more available, better ones to help automate processes (intelligent proof assistants)  26%
• Improve marketing  14%
• More research and development needed: mechanized proof assistance, tool support, real time systems, animation  9%
• Guidelines / Legislation / Accrediation to enforce use  8%
Formal Methods: A Survey

Part III: Survey of Formal Methods in Higher Education

• 94 questionnaires sent out, 39 returned
• 26 email replies
• Of the 65 replies, 5 were duplicates
• Of the 60 replies, 57 were teaching formal methods and other 3 thinking about teaching formal methods
• Z and VDM most popular
• Main uses: Specification / Refinement / Proof

Are formal methods still taught?
TraCIM Project

- Traceability for Computationally Intensive Metrology
- EU-funded
- Ran from June 2012 to May 2015
- Computationally intensive means significant use of mathematical software
- Will explain what traceability means in this context
1. **Computational Aim:** Clear, complete and unambiguous statement of the mathematics to be implemented. **Does not state how to implement**

2. **Reference Data Sets:** Reference input data and corresponding reference output data, *reference pair*

3. **Verification:** Software to be verified presented with a selection of reference input data as test data. Output generated by software compared corresponding reference output data

**Software should be traceable to Computational Aim via Reference Data Set**
Omissions and ambiguities may occur, even in computational aims expressed using mathematical notation.

Would formal methods allow them to be identified and addressed?

Could added discipline of formal methods allow better computational aims to be written?

Can be analysed using software tools.
University of York awarded one-year grant

First stage, select formal specification language

Z chosen; expressive style closest to mathematics used to write computational aims

ISO/IEC standard 13568 [18]

Recommendation of survey that Z becomes ISO standard
• **TraCIM**: This work has been carried out as part of the European Metrology Programme for Innovation and Research (EMPIR) project 15SIP06.

Thanks to Andy Galloway, Richard Paige and Jim Woodcock (University of York)
An undergraduate friendly introduction to theoretical computer science
1. Aims

2. About NPL / Data Science Department

3. Past work (small sample):
   - Milne and Strachey / OSI / Survey
   - TraCIM

4. Current work (very small sample):
   - Joint Appointments / Model-Based Systems Engineering

5. Conclusions

A couple of projects the presenter believes will be of particular interest to FACS

There’s a lot more going on
Joint Appointments

• Work in formal methods / formal aspects continues via joint appointments (JA) with universities

• Amongst others, current JAs with:
  • University of Strathclyde: Type systems for programs respecting dimensions
  • University of Edinburgh: Curated Databases

Data Science / NPL interests are in trustworthiness and managing complex systems
University of Strathclyde: Type systems for programs respecting dimensions

ISO 80000-1:2013 Quantities and units [21]

<table>
<thead>
<tr>
<th>Base quantity</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>L</td>
</tr>
<tr>
<td>mass</td>
<td>M</td>
</tr>
<tr>
<td>time</td>
<td>T</td>
</tr>
<tr>
<td>thermodynamic temperature</td>
<td>Θ</td>
</tr>
</tbody>
</table>

Explore the free abelian group of all base quantities

E.g. dimension of force denoted by \( \text{dim } F = LMT^{-2} \)
Model-Based Systems Engineering

• An initial investigation into using model-based systems engineering (MBSE)

• **AIMS:** Specify, design, implement, verify, validate and, above all, document complex cyber-physical systems at NPL

• Compare some of the software tools which are essential to MBSE

• Draw some conclusions.

• Suggest some future case studies
Model-Based Systems Engineering

• What is MBSE?
• “…formalized application of modeling to support system requirements, design, analysis, verification and validation activities… throughout development and later life cycle phases”.

  **International Council on Systems Engineering**

• Replace current document-based approach with model-based approach
**Model-Based Systems Engineering**

![Diagram of a software model showing requirements in UML.](image)

1. **Requirement** UR1:
   - **id**: UR1
   - **text**: The system shall calibrate user artefacts by making and processing measurements against an NPL standard artefact.

2. **Requirement** UR2:
   - **id**: UR2
   - **text**: The system shall generate values to be displayed on a calibration certificate by processing raw measurement data.

---

© NPL Management Ltd, 2021
Conclusions

• Techniques for development of fit-for-purpose software will always be of interest to NPL
  • Must now be strongly linked to metrology

• Formal methods / formal aspects / theoretical computer science will continue
  • Via universities, e.g. joint appointments, research excellence grants, PhD students

• …but can be difficult to find audience for the work
  • Metrologists don’t always “get” computer science, computer scientists don’t always “get” metrology

There’s work to be done...
Questions?

Department for Business, Energy & Industrial Strategy

Funded by BEIS

The National Physical Laboratory is operated by NPL Management Ltd, a wholly-owned company of the Department for Business, Energy and Industrial Strategy (BEIS).
Appendix I:
BCS working group on Formal Methods in Standards definition for formal methods
• **Definition** of formal methods for the BCS working group
  Formal Methods in Standards (as referenced in NPL Survey)

• **Formal Method**: A method making use of a *calculus* or *theory* to analyse or reason about software *specification* and/or *design*

• **Calculus**: A system of mathematical inference or computation in which results are obtained by the manipulation of formal symbols and expressions according to a finite set of precisely defined rules, e.g. the propositional calculus; the calculus of communicating systems

• **Specification**: The characterisation of all of the properties of an object relevant to some particular purpose (e.g. program design); the process of producing a specification.
Appendix II:
OSI: Distributed Transaction Processing
A little clearer... At Network layer and below, concerned with next node in network. At Transport layer and above concerned with endpoint.
• OSI TP is an Application layer protocol
• “…defines mechanisms which allow several distributed systems to be part of the same transaction, with the guarantee that resources (normally database entries) will only be changed as a result of the transaction if all systems agree” NPL report CISE 1 / 95 [9]

• Transaction: A set of related operations characterized by: atomicity, consistency, isolation and durability (ACID).
• A transaction that may span more than open system is called a distributed transaction
OSI: Distributed Transaction Processing

• ISO/IEC 10026-3: 1992 \[10\]

• Information technology – Open System Interconnection Distributed Transaction Processing – Part 3: Protocol specification

• Contains formal descriptions in:
  • Estelle (extended finite state machine model): Annex G
  • LOTOS (Language Of Temporal Ordering Specification): Annex H

NPL: Centre for information systems engineering
Work carried out 1990 – 1992
OSI: Distributed Transaction Processing

- **LOTOS**: Language Of Temporal Ordering Specification
- Consists of:
  - A language for **data description**: ACT-ONE
  - A **process calculus**:
    - Draws on Milner’s Calculus of Communicating Systems (CCS) [12]. Including **internal action**: $\tau$ (CCS), $i$ (LOTOS)
    - Also includes concepts from Hoare’s Communicating Sequential Processes (CSP) [13]
OSI: Distributed Transaction Processing

• LOTOS Example: After Logrippo / Faci / Haj-Hussein [14]
• A “lossy” channel (my term!)

Processes communicate via “gates”
Focus on the process calculus, not ACT-ONE

A specification is a hierarchy of process definitions

Processes run concurrently and communicate via gates

Behaviour expressions define process behaviour. Predefined expressions:

- **stop** Unsuccessful termination of process
- **exit** Successful termination of process
Operators build up behaviour expressions

- **Action prefix** operator “;”. E.g.:
  
  \[
  \text{pc1; pc2; exit}
  \]

  - Execute actions pc1 and pc2 and terminate

- **Choice** operator “[ ]”, choice between alternative behaviours. E.g.:
  
  \[
  \text{pc1; ( pc2; cc1; exit [ ] cc1; pc2; exit) }
  \]

  - Synchronize with Producer on gate pc1 then either with Producer again on gate pc2 or Consumer on gate cc1
Operators build up behaviour expressions

• **Enable** operator “>>”, sequential composition of two behaviour expressions. E.g.:

```
pc1; ( pc2; cc1; exit [] cc1; pc2; exit)
>> cc2; exit
```

• Use with i for “lossy” behaviour:

```
pc1; ( pc2; cc1; exit [] cc1; pc2; exit [] i; pc2; exit)
>> (cc2; exit [] i; exit)
```
Operators build up behaviour expressions

- **Interleaving** operator “|||” to express parallelism where no synchronisation required

- Interleaving is **NOT** parallelism, but a good enough approximation in this context? Debate…

- **Selective parallel** operator where processes must synchronize on common actions “| [<list of gates>] |”

  E.g. a; b; c; **exit** ![||](https://example.com/latex/image) d; a; c; **exit**

  Equivalent to:

  d; a; (b; c; c; **exit** ![||](https://example.com/latex/image) c; b; c; **exit**)
Operators build up behaviour expressions

- **hide** operator hides actions internal to a process.

  E.g.:

  \[
  \text{hide } b \text{ in } a; b; \text{ exit } |[b]| b; c; \text{ exit}
  \]

  Equivalent to:

  \[
  a; i; c; \text{ exit}
  \]
specification Lossy_Channel [pc1, pc2, cc1, cc2] : exit

behaviour
(
    Producer [pc1, pc2] 
    | [pc1, pc2] | 
    Channel [pc1, pc2, cc1, cc2] 
    | [cc1, cc2] | 
    Consumer [cc1, cc2]
)

where…
process Producer[pc1, pc2]: exit := pc1; pc2; exit endproc

process Channel[pc1, pc2, cc1, cc2]: exit := pc1; ( pc2; cc1; exit [] cc1; pc2; exit [] i; pc2; exit) >> (cc2; exit [] i; exit) endproc

process Consumer[cc1, cc2]: exit := cc1; (cc2; exit [] exit) [] cc2; exit [] exit endproc
NI LabVIEW [15] simulation of lossy channel

subVIs, processes running in parallel communicating via “wires”
In channel subVI, use random number generator to simulate data loss.

Internal action is not controlled by the environment.
OSI: Distributed Transaction Processing

TP Service User Interface

TP_MACF
- TPService
- TPMACFProtocol

C_MACF
- TP_CInterface
- CMACFProtocol

Gates
- SAOs
- SAO

PM

Presentation Layer

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACF</td>
<td>Multiple Association Control Function</td>
</tr>
<tr>
<td>PM</td>
<td>Protocol machine</td>
</tr>
<tr>
<td>SAO</td>
<td>Single association object</td>
</tr>
<tr>
<td>TP</td>
<td>Transaction Processing</td>
</tr>
<tr>
<td>TPPM</td>
<td>Transaction Processing Protocol Machine</td>
</tr>
</tbody>
</table>
OSI: Distributed Transaction Processing

**Brief** look at the specification (a thing of beauty)

**specification** OSITP[tpsu, p]: **noexit**

**behaviour**

\[
\text{AEI}[tpsu, p]
\]

**where**

**process** AEI[tpsu, p]: **noexit** :=

\[
\text{PM}[tpsu, p] | [tpsu, p] | \text{conformance}_\text{requirements}[tpsu, p]
\]

**endproc**
Continued…

```plaintext
process PM[tpsu, p]: noexit :=
    hide macf, aei in
    MACFs[tpsu, macf, aei] | [macf, aei] | SAOs[macf, p, aei]
endproc
```
Continued…

```
process MACFs[tpsu, macf, aei]: noexit :=
  hide caf in
  TP_MACF[tpsu, caf, macf, aei] | [caf] |
  C_MACF[caf, macf, aei]
endproc
```
Continued…

```plaintext
process SAOs[macf, p, aei]: noexit :=
  hide macf, aei in
  (SAO [macf, p, aei] >> stop) |||
i; SAOs[macf, p, aei]
endproc
```
OSI: Distributed Transaction Processing

- “…highlights many of the problems of describing a real-world protocol.”
- Team of three worked at NPL for three years on the LOTOS description
- Delegated to NPL by the ISO committee responsible after initial work elsewhere
- “As the work progressed… increasing numbers of problems were found with the English text”
• Report continued…

• “By far the most all-pervading problem with LOTOS was the awkwardness of the data typing language …every datatype has to be defined from scratch, there are no short cuts”

• C.F. From ACT-ONE to Miranda, a Translation Experiment Charles, Bowman, Thompson, University of Kent (1997) [16]

• “…benefits from producing a formal description of a standard as it was being produced …experts on-hand who could explain… the protocol …writers of the English text… could correct their specification when questions and comments from NPL revealed genuine problems”
Report continued…

“…diplomatic form of words. In many cases ambiguities can not be reflected in the formal description, this is, after all, one of the advantages of using formal methods.”

Tools:

“…description was so large that it defeated all LOTOS tools that it was given to.”

“Most tools failed to read in the entire… description and none made any progress in trying to animate…”

“…only verification possible… by manual comparison with corresponding English text.”
OSI: Distributed Transaction Processing

• Report continued…

• “…diplomatic form of words. In many cases ambiguities can not be reflected in the formal description, this is, after all, one of the advantages of using formal methods.”

• Tools:
  
  • “…description was so large that it defeated all LOTOS tools that it was given to.”
  
  • “Most tools failed to read in the entire… description and none made any progress in trying to animate…”
  
  • “…only verification possible… by manual comparison with corresponding English text.”
Appendix III: Computation Aim: Z-Spec
Explore use of formal methods for specification and analysis of computational aims

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Title</th>
<th>Description</th>
<th>Specification</th>
<th>Full Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>en/-/000008</td>
<td>Gaussian area filter</td>
<td>Gaussian filter for calculating surface texture area parameters.</td>
<td>SurfaceTexture_GaussianArealFilter.pdf [127 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/-/000009</td>
<td>Gaussian profile filter</td>
<td>Gaussian filter for calculating surface texture profile parameters.</td>
<td>SurfaceTexture_GaussianProfileFilter.pdf [110 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/-/000010</td>
<td>Arithmetical mean deviation ( \text{Pa} ) of assessed profile</td>
<td>Amplitude surface texture parameter for primary profile.</td>
<td>SurfaceTexture_Pa.pdf [105 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/-/000011</td>
<td>Best fit of a Gaussian to 3D point coordinates in coordinate metrology</td>
<td>That the sum of the squared distances from a point ((x_i, y_i, z_i)) to the line is minimized.</td>
<td>Gaussian 3D line pdf [304 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/-/000012</td>
<td>Best fit of a Gaussian sphere to 3D point coordinates in coordinate metrology</td>
<td>Determine a sphere to data points such that the sum of the squared distances from a point ((x_i, y_i, z_i)) to the sphere is minimized.</td>
<td>Gaussian Sphere.pdf [303 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/-/000023</td>
<td>Exponential decay</td>
<td>Fit a baseline and an exponential decay function through Cavity Ring Down Spectroscopy data</td>
<td>CompAimExpDecay_v4.pdf [109 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/-/000024</td>
<td>Calculate Error Vector Magnitude of a digital signal</td>
<td></td>
<td>CompAim_EVM_v2.pdf [107 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/-/000025</td>
<td>Peak fitting</td>
<td></td>
<td>CompAimPeaks_v6.pdf [137 KB]</td>
<td>Click for further details</td>
</tr>
<tr>
<td>en/D/000016</td>
<td>Principal Components Analysis (PCA)</td>
<td>Evaluate principal components and principal component loadings from mean-centred data.</td>
<td>PrincipalComponentsAnalysis_revised.pdf [151 KB]</td>
<td>Click for further details</td>
</tr>
</tbody>
</table>

http://www.tracim-cadb.npl.co.uk/
Minimum Circumscribed Circle (MCC):

Determine centre coordinates and radius of the circle of minimum radius that circumscribes a given set of points in the $xy$-plane.
TraCIM Project: Z Specification

• Z specifications structured using *schemas*:

```
MCCInputs
```
```
m : N_1
X : M
```
```
X ∈ realmatrix(m, 2)
```

• Upper section contains variable declarations

• Lower section defines relationship between values of the variables and constraints on these values
\[ \langle X_0(1), X_0(2), r \rangle = \text{safemin}_v( \{ x_0, y_0, r : \mathbb{R} \mid (\forall i : 1 \ldots m \bullet (X(i)(1) - x_0)^2 + (X(i)(2) - y_0)^2 - r^2 \leq 0) \bullet (\langle x_0, y_0, r \rangle, r) \} ) \]
TraCIM: Z Specification [19]

- Confidence in validity of formal specification can be increased using software tools
- E.g. MCC for input data set containing two (distinct) data points is circle with diameter defined by those points
- Characterised in Z as:

\[
\begin{align*}
\forall \ MMCCComputation \ | \ m = 2 \bullet Circle((X_0(1), X_0(2)), r) &= \text{DiamToCircle(Line}(X((1)(1), X(1)(2)), (X(2)(1), X(2)(2))))
\end{align*}
\]

- Characterised in Mathematica as:

\[
\text{PropertyDiag}[x1\_, \ y1\_, \ x2\_, \ y2\_] \ := \ \\
\text{TwoPointMCCCircle}[x1, \ y1, \ x2, \ y2] \ == \ \\
\text{DiameterLinetoCircle[Line[\{\{x1, \ y1\}, \ \\
\{x2, \ y2\}\}]]}
\]
Appendix IV:
Joint Appointment: University of Edinburgh
University of Edinburgh JA: Curated databases

Buneman, Chapman, Cheney 2006 (SIGMOD)
University of Edinburgh JA: Programming foundations for trusted data science

- Data curators need:
  - **Web interfaces** to scientific databases
  - **Transparency** about data sources
  - Support for **synchronizing** data
  - Understanding of how data change over time

- We provide:
  - Single programming language for Web + database applications
  - Language-integrated **provenance** for queries
  - Language-integrated update via **Relational lenses**
  - Language-integrated **temporal queries** (versioning, time travel)
Why?

Safety: avoid SQL injection attacks
Convenience: catch type errors early
Productivity: use general programming features in queries

How?

query { for (x <- employees) where (x.salary > 50000) [(name = x.name)] } 

select name from employees e where e.salary > 50000

[“Bert”, “Drew”, “Erik”, “Gina”]
University of Edinburgh JA: Language-integrated query

- Why?
  - Safety: avoid SQL injection attacks
  - Convenience: catch type errors early
  - Productivity: use general programming features in queries

- How?

```plaintext
query { for (x <-- employees) where (x.salary > 50000) [(name = x.name)] } 

name
Bert
Drew
Erik
Gina

select name from employees e where e.salary > 50000
```

RDBMS

[“Bert”, “Drew”, “Erik”, “Gina”]


References


© NPL Management Ltd, 2021
References


17. Austin, S, Parkin, G, Formal Methods: A Survey. 1993. Contact keith.lines@npl.co.uk

References


Webinar: Dimensionally correct by construction: Type systems for programs

Dimensionally correct by construction: Type systems for programs respecting dimensions.

Speakers
- Conor McBride
- Fredrik Nordvall Forsberg

Agenda

Tuesday 15 June 5:15 pm – 8:00 pm