Formal Methods: The National Physical Laboratory's Experiences

Keith Lines

Data Science Department

National Physical Laboratory, UK

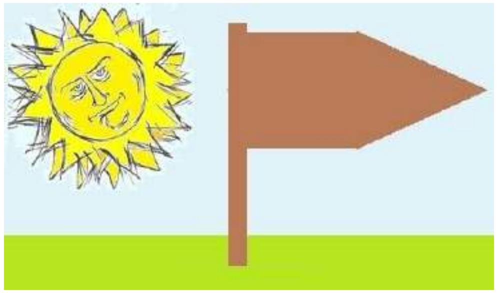
BCS, FACS meeting 06/04/2021 Draft v3 06/04/2021

Contents

Aims of the Presentation

National Physical Laboratory

- 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

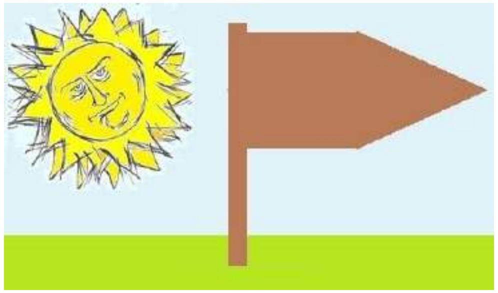


Aims of the Presentation

National Physical Laboratory

- 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)

Nothing new



Contents

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…

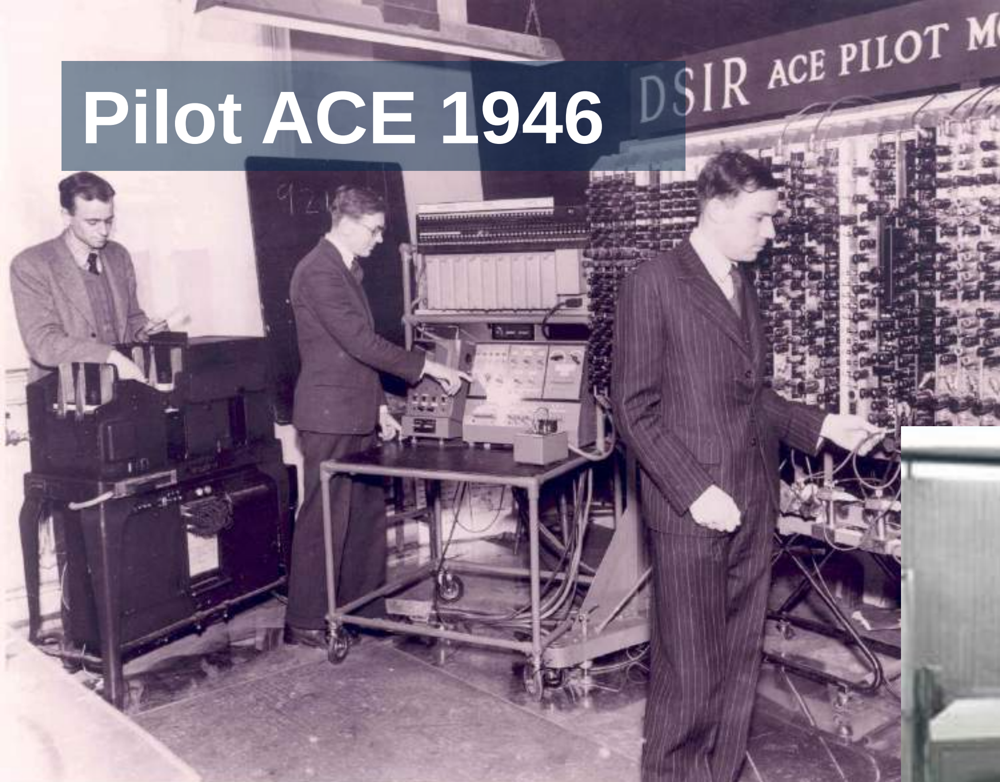
- An introduction to formal methods
  - Will assume at least a basic knowledge
- A complete picture

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

About NPL: Data Science Dept.

National Physical Laboratory

Aim: Confidence in the intelligent & effective use of data

- Mix of mathematicians, computer scientists, statisticians and physicists. includes secondees from other NPL departments
- ~40 staff across three sites, including joint appointments with Cambridge, Surrey and Strathclyde.
- ~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.

NPL: DS: Modelling & Analytics

National Physical Laboratory

- **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 (**F**indable, **A**ccessible, **I**nteroperable and **R**eproducible)
- Definition of minimum metadata standards initial focus on imaging in healthcare and life sciences

Contents

NPL National Physical Laboratory

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

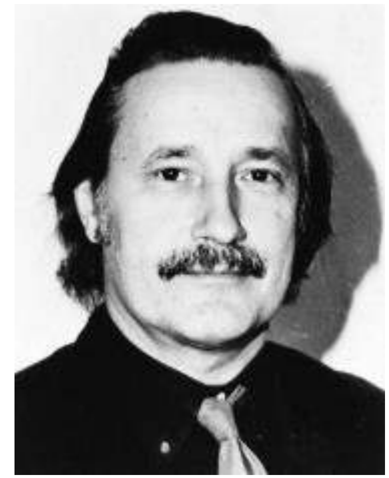
Milne and Strachey [2]

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

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

DITC – ISE -- THIS

National Physical Laboratory

"(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]

BCS working group

- 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]

Report by Brian Wichmann [7]

OSI: Distributed Transaction Processing

- 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"

OSI: Distributed Transaction Processing

- 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”

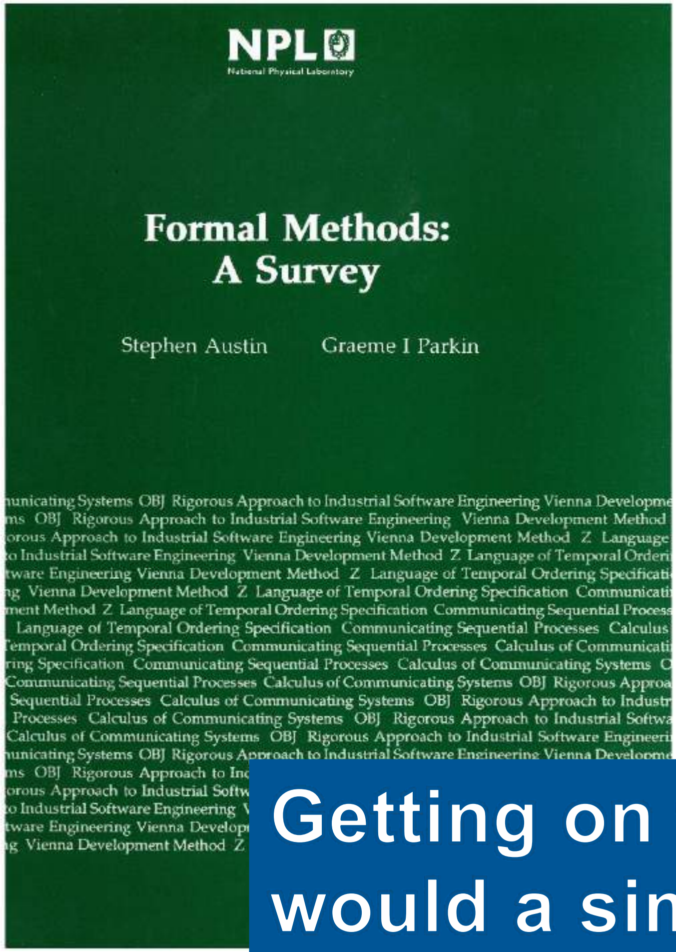
OSI: Distributed Transaction Processing

- 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.”

Goto Appendix II for further details

- **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.”

Formal Methods: A Survey [17]



- 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?

Formal Methods: A Survey

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

NPL National Physical Laboratory

Part I / Part II: Formal Methods in Software Engineering

800 questionaries sent out, 444 returned, 104 analysed

All questionaries read

Questionaires	Received	Analysed
Post (UK)	385	104
Electronic (UK)	2	1
Post (non-UK)	48	3
Electronic (non-UK)	9	3
Discarded		15
Total	444	126

Formal Methods: A Survey

Part I / Part II: Formal Methods in Software Engineering

Which formal methods have you considered using?

Method	Percent
Z	55 %
VDM	55 %
LOTOS, CSP, CCS	18 %
None	11 %
OBJ	7 %
RAISE	5 %
Temporal Logics	4 %
Others	24%

Formal Methods: A Survey

NPL National Physical Laboratory

Part I / Part II: Formal Methods in Software Engineering

In what way would you consider using formal methods (specification, refinement, proving etc.)?

Use	Percent
Specification	89 %
Proofs	39 %
Refinement	17 %
Design	10 %
Any way	5 %
Verification	5 %
Requirements capture	5 %
Others	11 %

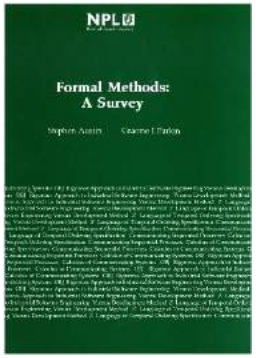
Not 100 %

Formal Methods: A Survey

Part I / Part II: Formal Methods in Software Engineering

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

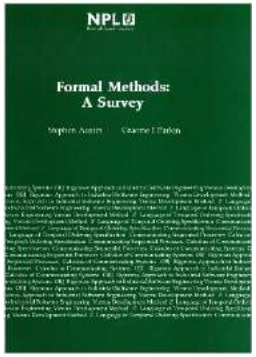


Formal Methods: A Survey

Part I / Part II: Formal Methods in Software Engineering

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 %



Formal Methods: A Survey

Part I / Part II: Formal Methods in Software Engineering

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 %

As LOTOS

Formal Methods: A Survey

Part I / Part II: Formal Methods in Software Engineering

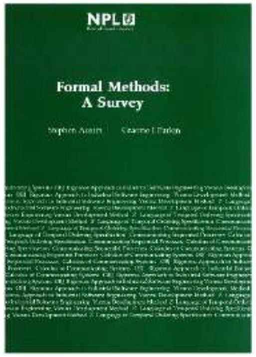
Do you have any suggestions on how to overcome these barriers?

- 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

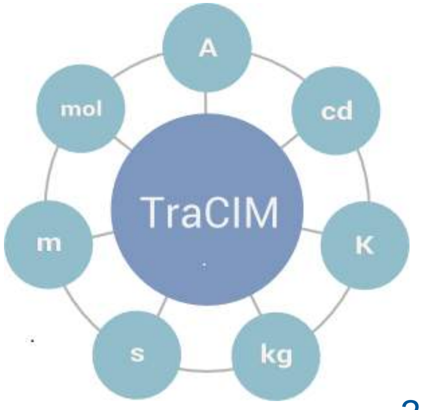
- 94 questionnaires sent out, 39 returned
- 26 email replies
- Of the 65 replies, 5 were duplicates
- Of the 60 replies, 57 were teaching formal methods and other 3 thinking about teaching formal methods
- Z and VDM most popular
- Main uses: Specification / Refinement / Proof



Are formal methods still taught?

TraCIM Project

- **T**raceability for **C**omputationally **I**ntensive **M**etrology
- 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



TraCIM Project: Verifying Software

1. **Computational Aim:** Clear, complete and unambiguous statement of the mathematics to be implemented. **<u>Does not state how to implement</u>**
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

TraCIM Project: Formal Specification

- 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

TraCIM Project: Formal Specification

- 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 standard13568 [18]

Recommendation of survey that Z becomes ISO standard

TraCIM Project: Acknowledgements

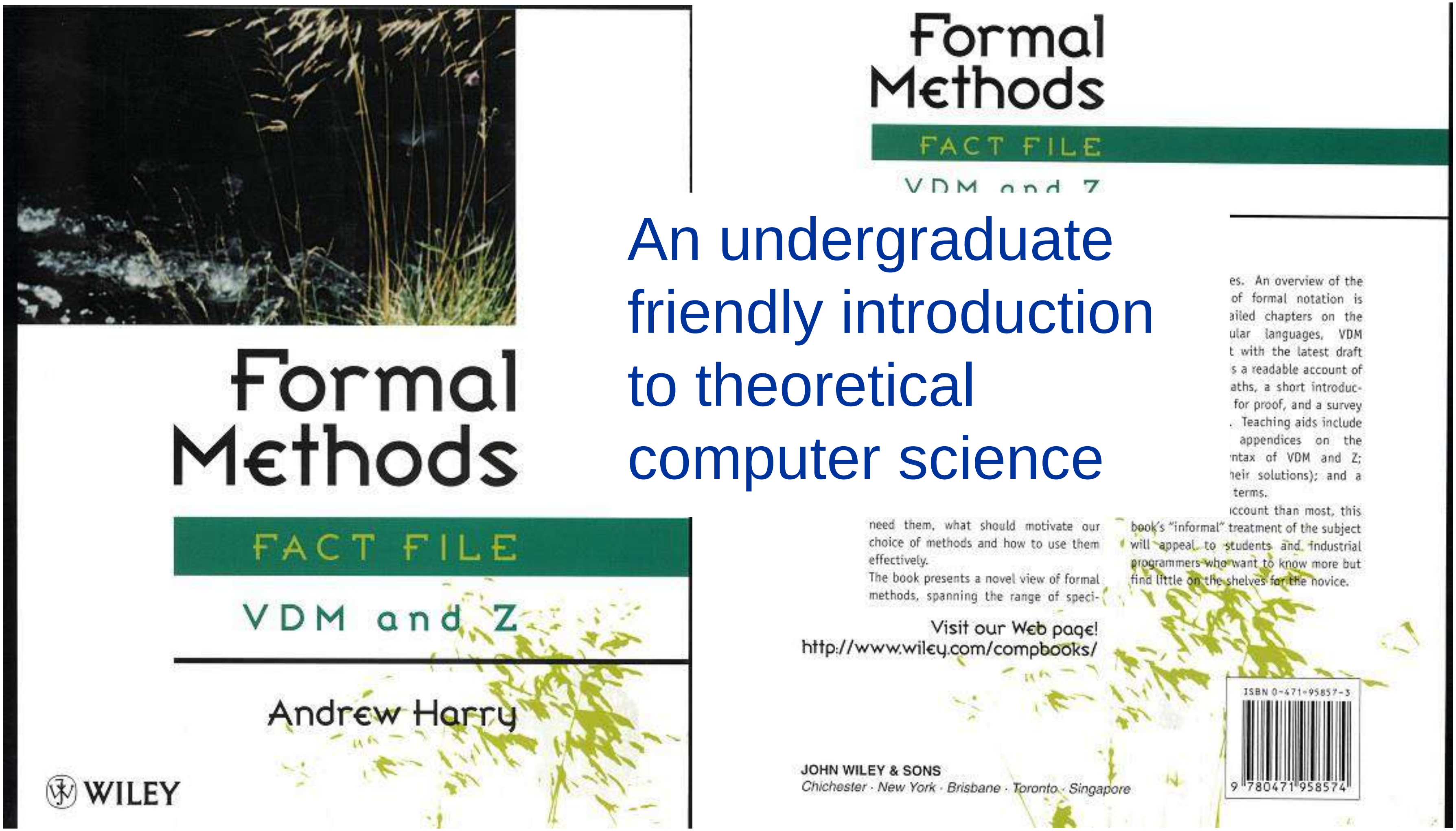
- **TraCIM:** This work has been carried out as part of the European Metrology Programme for Innovation and Research (EMPIR) project 15SIP06.

EMPIR EURAMET

The EMPIR initiative is co-funded by the European Union's Horizon 2020 research and innovation programme and the EMPIR Participating States

Thanks to Andy Galloway , Richard Paige and Jim Woodcock (University of York)

Textbook by ex-NPL staff member [20]



An undergraduate friendly introduction to theoretical computer science

Contents

1. Aims
2. About NP
3. Past work

   Milne an

   TraCIM

4. Current work (very small sample):

   Joint Appointments / Model-Based Systems Engineering

5. Conclus

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]

Base quantity	Symbol
length	L
mass	M
time	T
thermodynamic temperature	Θ

Explore the <u>free abelian group</u> of all base quantities

E.g. dimension of force denoted by dim F = LMT^{-2}

Model-Based Systems Engineering

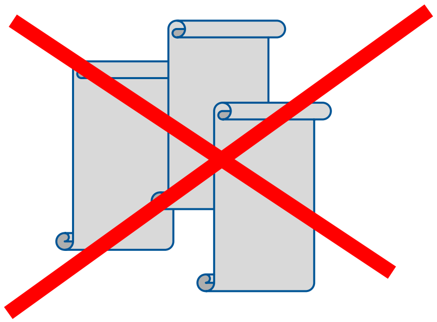
- An initial investigation into using model-based systems engineering (MBSE)
- **AIMS:** Specify, design, implement, verify, validate and, above all, document **complex cyber-physical systems** at NPL
- Compare some of the software tools which are essential to MBSE
- Draw some conclusions.
- Suggest some future case studies

Model-Based Systems Engineering

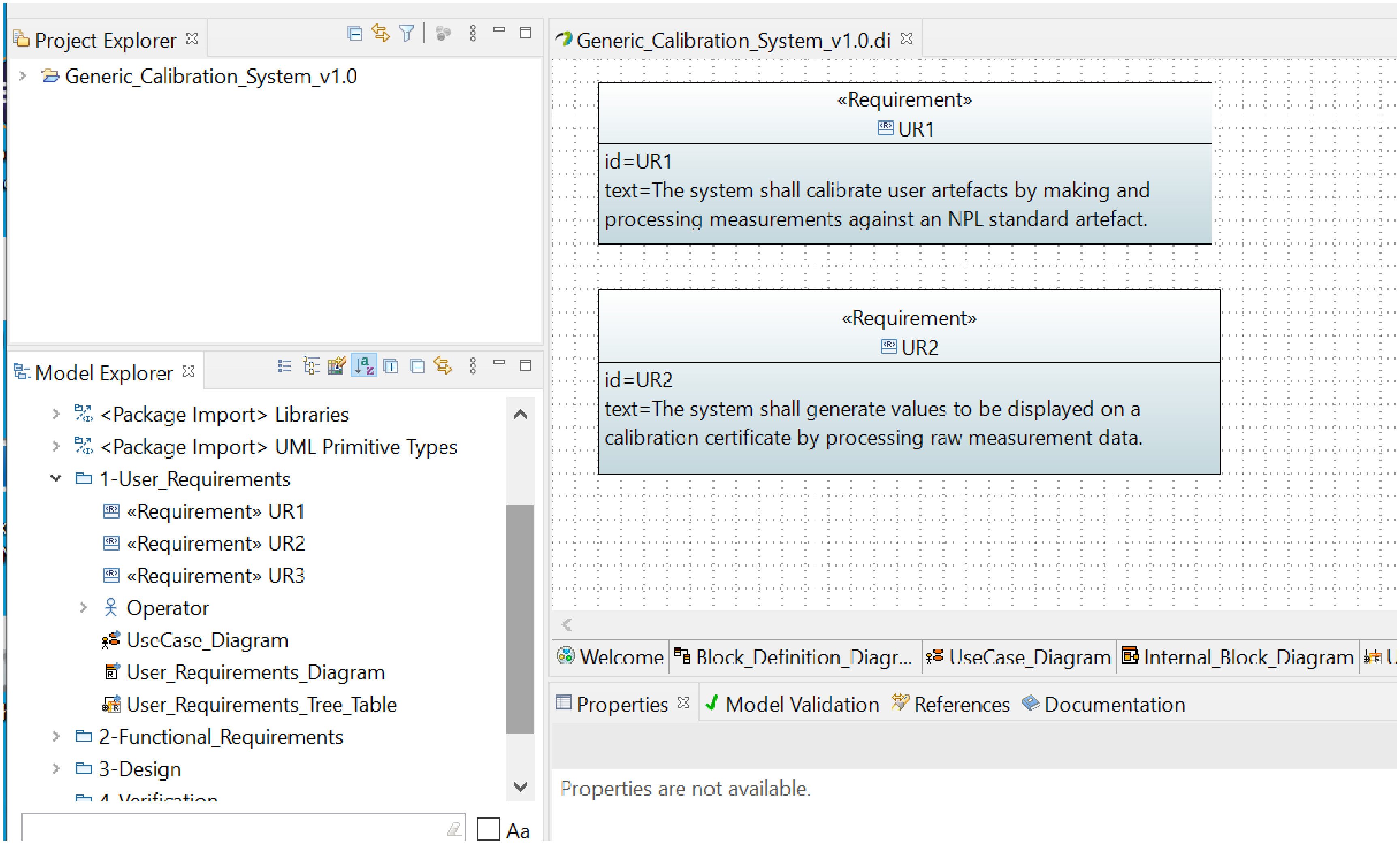
- What is MBSE?
- "…formalized application of modeling to support system requirements, design, analysis, verification and validation activities… throughout development and later life cycle phases".

International Council on Systems Engineering

- Replace current document-based approach with model-based approach



Model-Based Systems Engineering



Conclusions

Trustworthiness

- 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

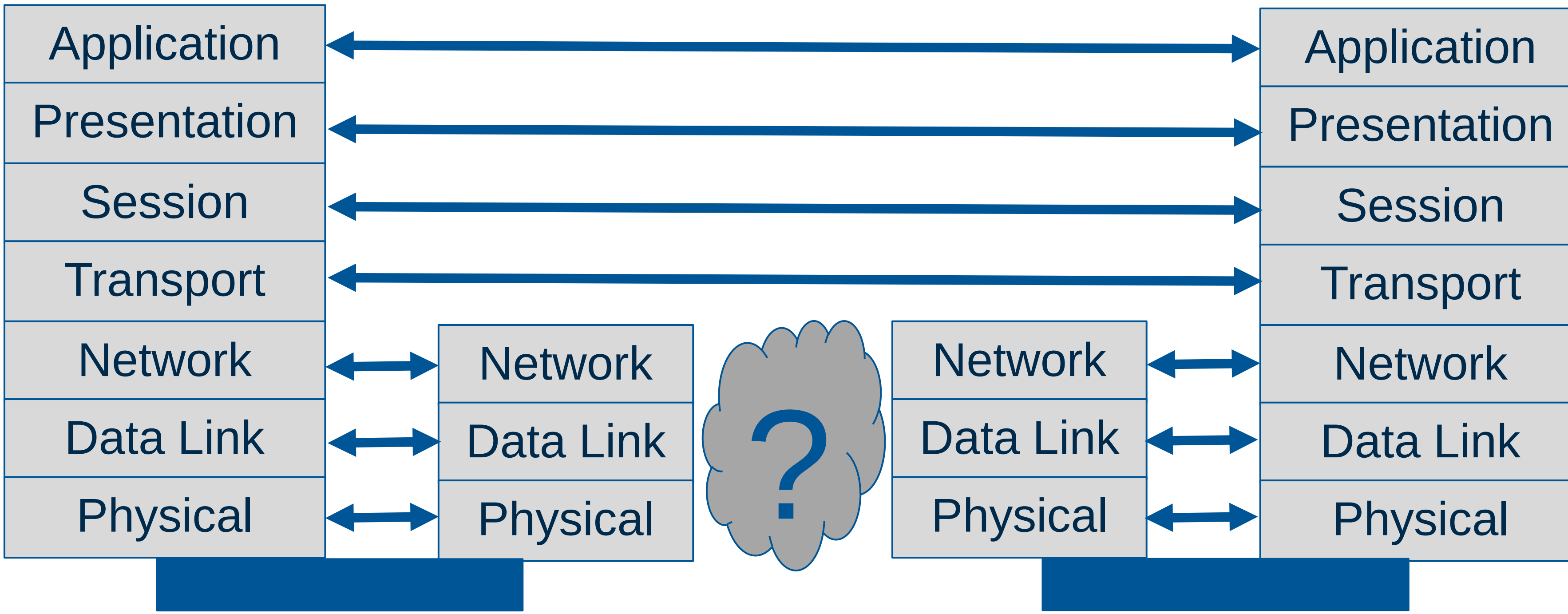
- **Definition** of formal methods for the BCS working group Formal Methods in Standards (as referenced in NPL Survey)
- **Formal Method:** A method making use of a **calculus** or **theory** to analyse or reason about software **specification** and/or **design**
- **Calculus:** A system of mathematical 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.

recursive

Appendix II:
OSI: Distributed Transaction Processing

OSI: Distributed Transaction Processing

Open Systems Interconnection model [8]



A little clearer... At Network layer and below, concerned with next node in network. At Transport layer and above concerned with endpoint

OSI: Distributed Transaction Processing

- 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]

ISO/IEC 10026-1:1992 [6]

- **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

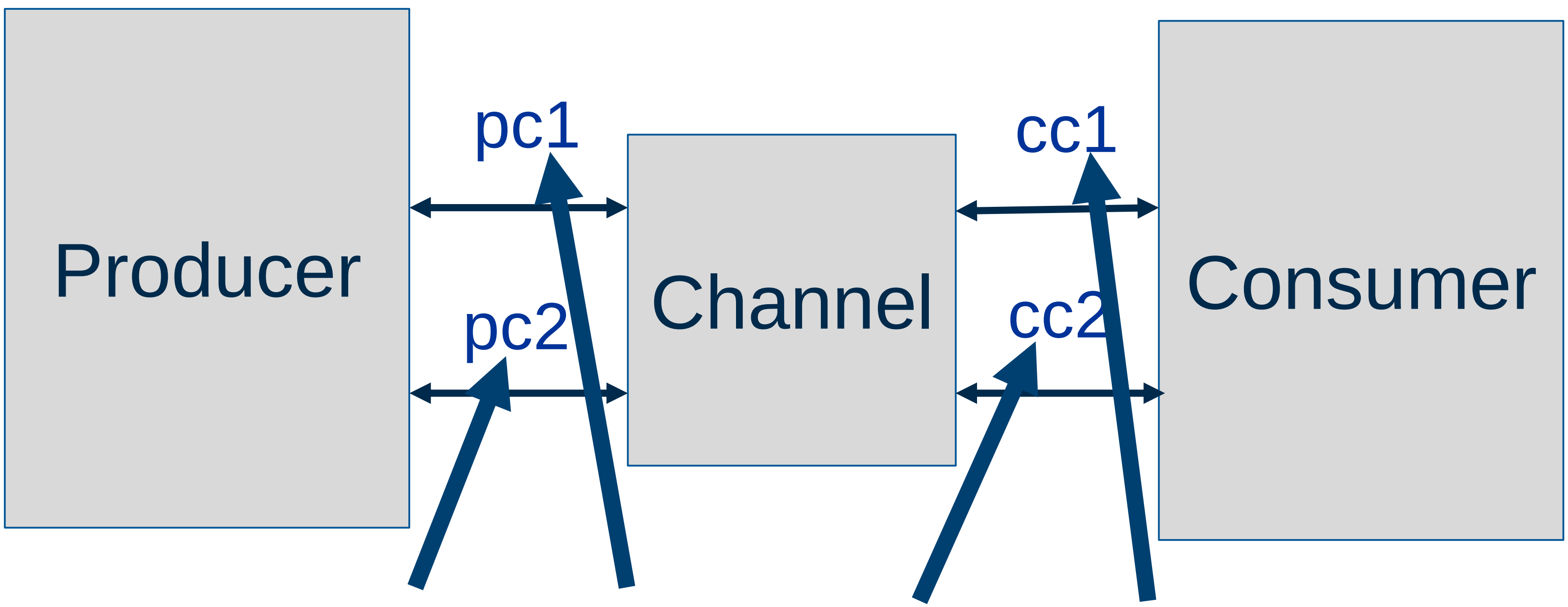
NPL: Centre for information systems engineering
Work carried out 1990 – 1992

OSI: Distributed Transaction Processing

- **LOTOS:** Language Of Temporal Ordering Specification
- ISO standards: ISO/IEC 15437:2001 / ISO 8807:1989 [11]
- 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**: τ (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”

OSI: Distributed Transaction Processing

- 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

OSI: Distributed Transaction Processing

Operators build up behaviour expressions

- **Action prefix** operator ";". E.g.:

  pc1; pc2; **exit**

  - Execute actions pc1 and pc2 and terminate
- **Choice** operator "[]", choice between alternative behaviours. E.g.:

  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

OSI: Distributed Transaction Processing

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**)

OSI: Distributed Transaction Processing

Operators build up behaviour expressions

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

But not today

- 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** |[a]| d; a; c; **exit**

  "||" for synchronize on all actions

  Equivalent to:

  d; a; (b; c; c; **exit** [] c; b; c; **exit**)

OSI: Distributed Transaction Processing

Operators build up behaviour expressions

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

E.g.:

hide b **in** a; b; **exit** |[b]| b; c; **exit**

Equivalent to:

a; **i**; c; **exit**

And now bring it all together…

OSI: Distributed Transaction Processing

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…

OSI: Distributed Transaction Processing

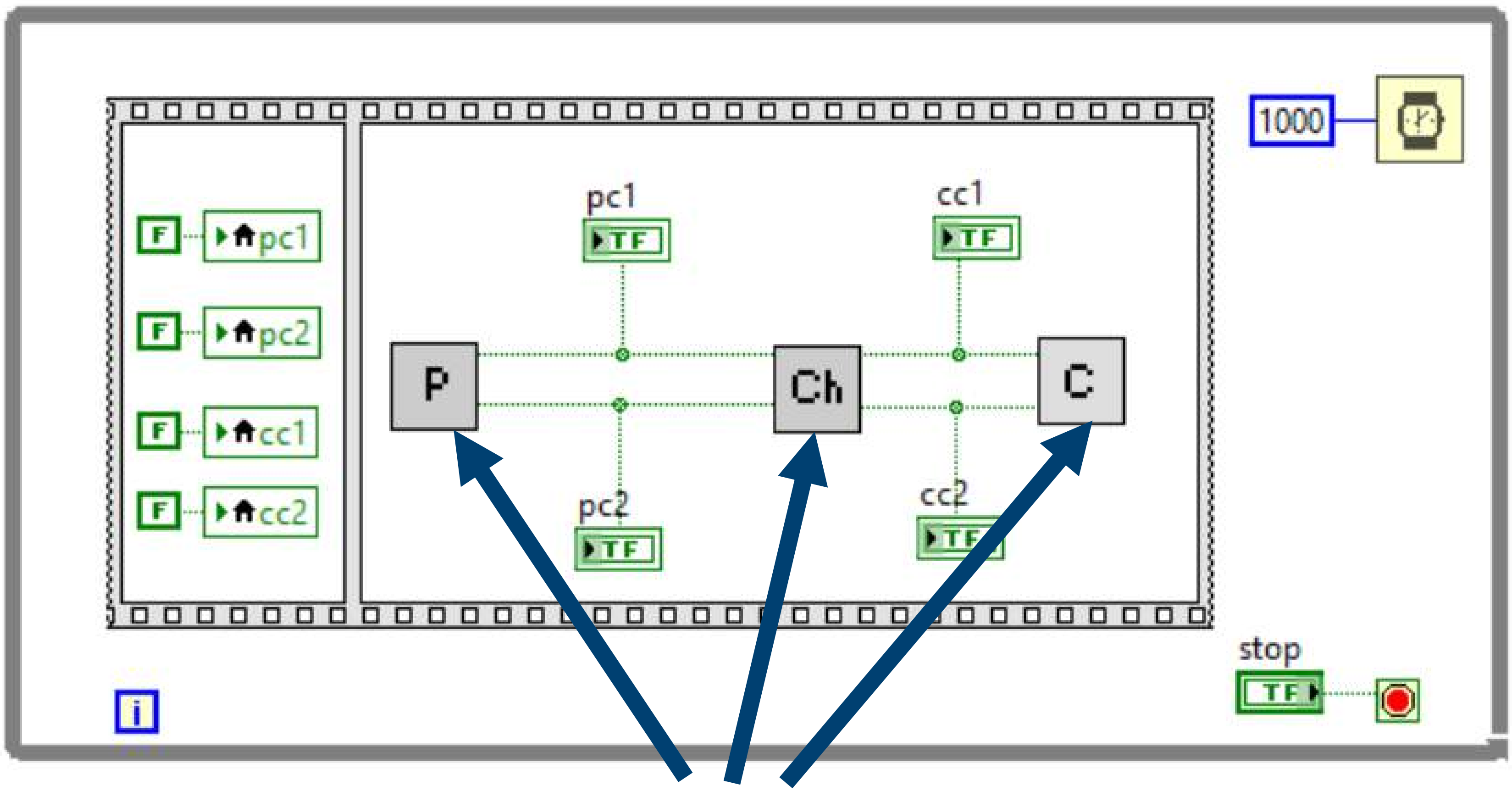
process Producer[pc1, pc2]: **exit** :=
pc1; pc2; **exit**
endproc

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

OSI: Distributed Transaction Processing

NI LabVIEW [15] simulation of lossy channel

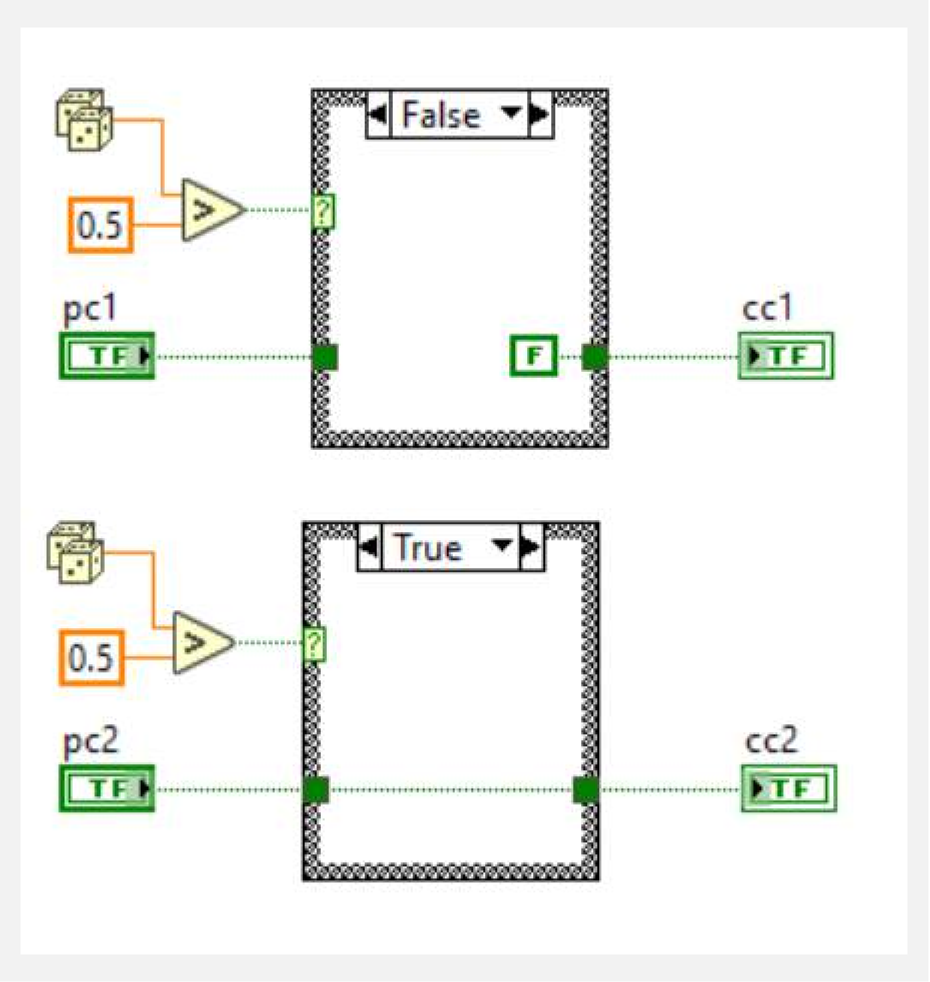


subVIs, processes running in parallel communicating via “wires”

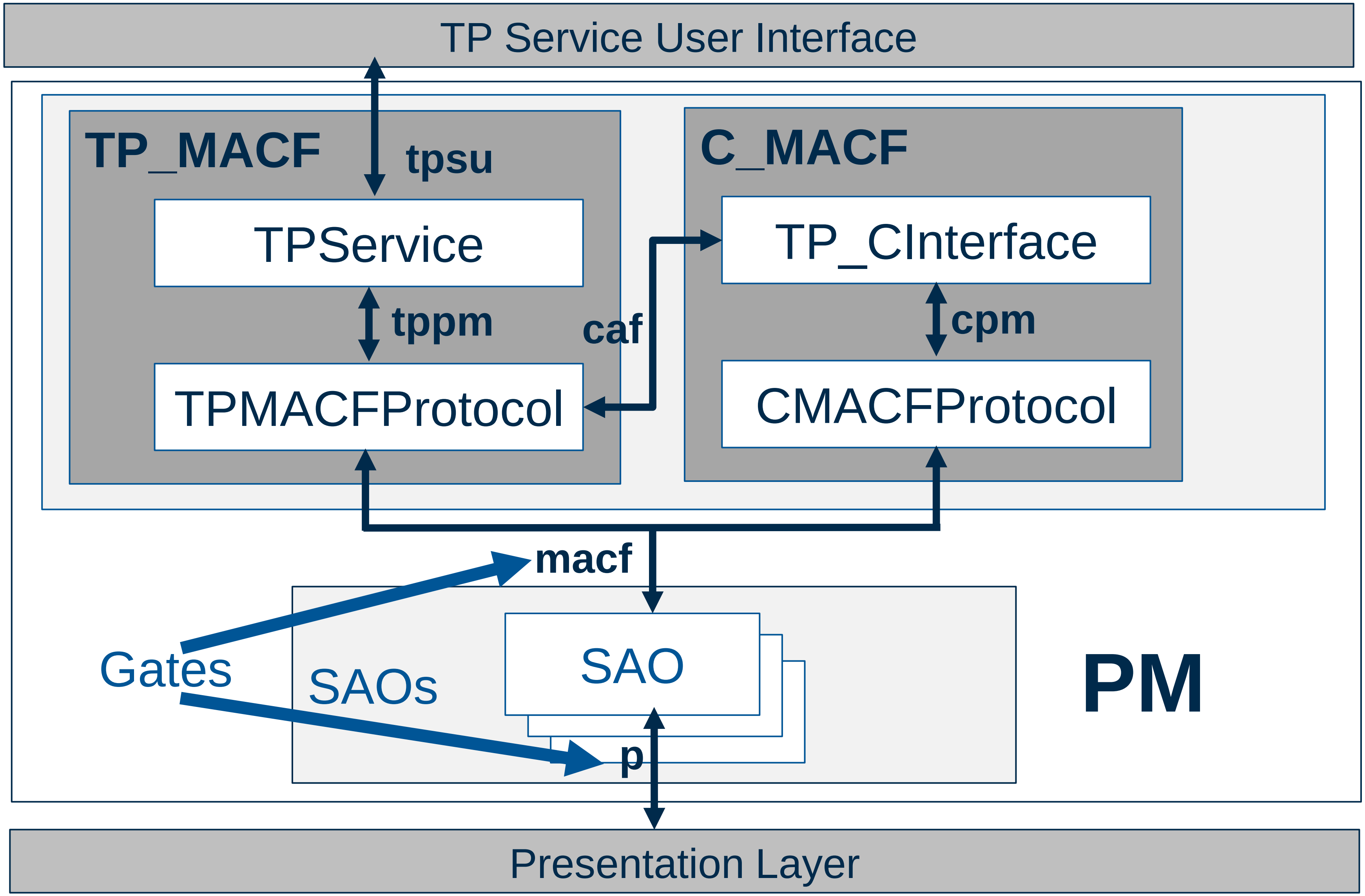
OSI: Distributed Transaction Processing

In channel subVI, use random number generator to simulate data loss

Internal action **i** not controlled by the environment



OSI: Distributed Transaction Processing



OSI: Distributed Transaction Processing

Terminology ISO/IEC 10026-1:1992 [6]

Term	Description
MACF	Multiple Association Control Function
PM	Protocol machine
SAO	Single association object
TP	Transaction Processing
TPPM	Transaction Processing Protocol Machine

OSI: Distributed Transaction Processing

Brief look at the specification (a thing of beauty)

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

behaviour

AEI[tpsu, p]

where

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

PM[tpsu, p] | [tpsu, p] |
conformance_requirements[tpsu, p]

endproc

OSI: Distributed Transaction Processing

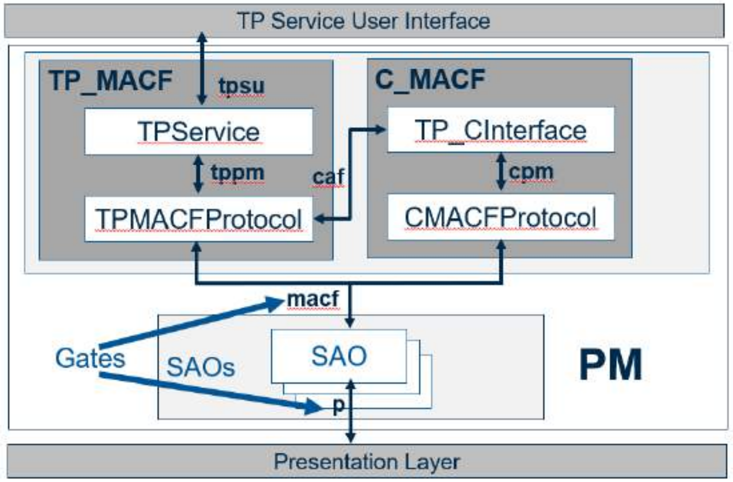
Continued…

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

hide macf, aei **in**

MACFs[tpsu, macf, aei] | [macf, aei] | SAOs[macf, p, aei]

endproc



OSI: Distributed Transaction Processing

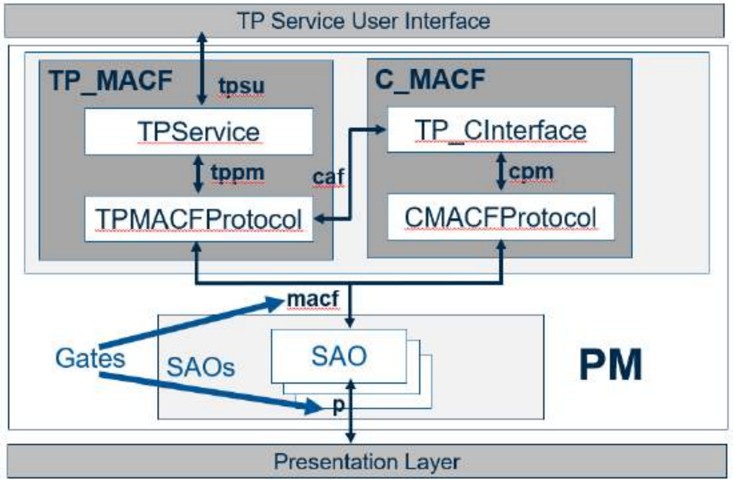
Continued…

process MACFs[tpsu, macf, aei]: **noexit** :=

hide caf **in**

TP_MACF[tpsu, caf, macf, aei] | [caf] | C_MACF[caf, macf, aei]

endproc



OSI: Distributed Transaction Processing

Continued…

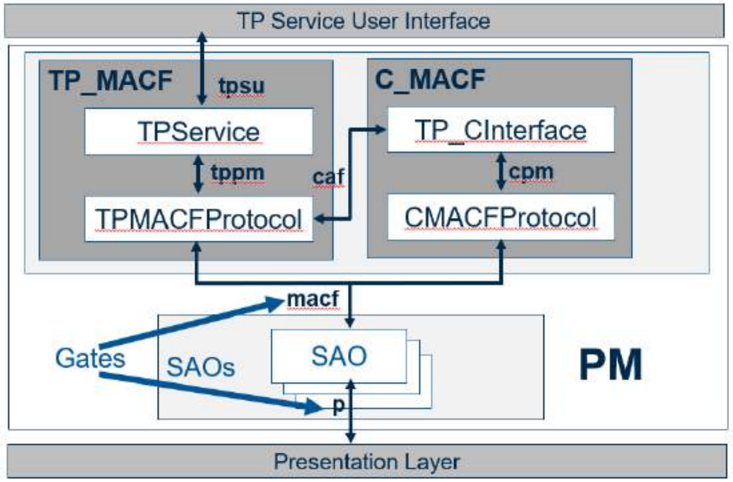
process SAOs[macf, p, aei]: **noexit** :=

hide macf, aei **in**

(SAO [macf, p, aei] >> **stop**) |||

i; SAOs[macf, p, aei]

endproc



OSI: Distributed Transaction Processing

- 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"

OSI: Distributed Transaction Processing

- 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”

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."

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

TraCIM Project: Computational Aims Database

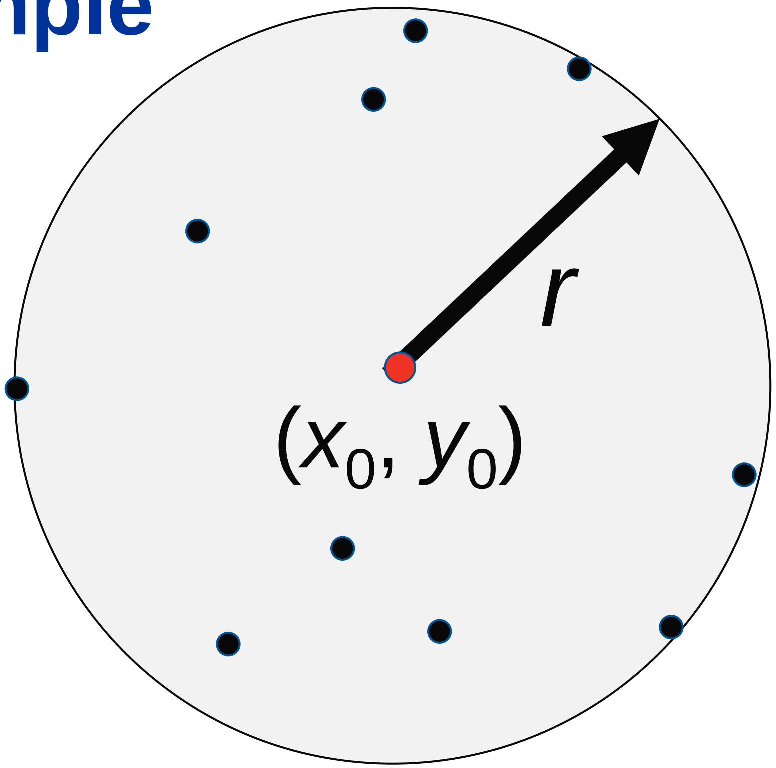
NPL National Physical Laboratory

Identifier	Title	Description	Specification	Full Details
en/-/0/000008	Gaussian areal filter	Gaussian filter for calculating surface texture areal parameters.	SurfaceTexture_GaussianArealFilter.pdf [127 KB]	Click for further details
en/-/0/000009	Gaussian profile filter	Gaussian filter for calculating surface texture profile parameters.	SurfaceTexture_GaussianProfileFilter.pdf [110 KB]	Click for further details
en/-/0/000010	Arithmetical mean deviation Pa of assessed profile	Amplitude surface texture parameter for primary profile.	SurfaceTexture_Pa.pdf [105 KB]	Click for further details
en/-/0/00001				Click for further details
en/-/0/00001				Click for further details
en/-/0/000013	Best fit of a Gaussian line to 3D point coordinates in coordinate metrology	that the sum of the squared distances from a point (xi, yi, zi) to the line is minimized	Gaussian 3D line.pdf [304 KB]	Click for further details
en/-/0/000014	Best fit of a Gaussian sphere to 3D point coordinates in coordinate metrology	Determine a sphere to data points such, that the sum of the squared distances from a point (xi, yi, zi) to the sphere is minimized	Gaussian Sphere.pdf [303 KB]	Click for further details
en/-/0/000023	Exponential decay	Fit a baseline and an exponential decay function through Cavity Ring Down Spectroscopy data	CompAimExpDecay_v4.pdf [109 KB]	Click for further details
en/-/0/000024	Calculate Error Vector Magnitude of a digital signal		CompAim_EVM_v2.pdf [107 KB]	Click for further details
en/-/0/000025	Peak fitting	Fit one or more line profiles to spectroscopic measurement data	CompAimPeaks_v6.pdf [137 KB]	Click for further details
en/D/0/000016	Principal Components Analysis (PCA)	Evaluate principal components and principal component loadings from mean-centred data	PrincipalComponentsAnalysis_revised.pdf [151 KB]	Click for further details

Explore use of formal methods for specification and analysis of computational aims

http://www.tracim-cadb.npl.co.uk/

TraCIM Project: Computational Aims: Example



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 [17]

- Z specifications structured using *schemas*:

MCCInputs

$m : \mathbb{N}_1$

$X : \mathbb{M}$

$X \in \mathbf{realmatrix}(m, 2)$

- Upper section contains variable declarations

- Lower section defines relationship between values of the variables and constraints on these values

TraCIM Project: Z Specification

MCCInputs

$m : \mathbb{N}_1$

$X : \mathbb{M}$

$X \in \mathbf{realmatrix}(m, 2)$

MCCOutputs

$X_0 : \mathbf{realvector}(2)$

$r : \mathbb{R}$

$r \geqslant 0$

TraCIM Project: Z Specification

MCCComputation

MCCInputs

MCCOutputs

$$\langle X_0(1), X_0(2), r\rangle = \mathbf{safemin}_v\,($$
$$\{x_0, y_0, r : \mathbb{R} \mid$$
$$(\forall\, i : 1\,..\,m \bullet (X(i)(1) - x_0)^2 + (X(i)(2) - y_0)^2 - r^2 \leqslant 0)$$
$$\bullet\ (\langle x_0, y_0, r\rangle, r)\ \})$$

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:

$$\vdash ?\ \forall\ MMCComputation \mid m = 2 \bullet Circle((X_0(1), X_0(2)), r) = DiamToCircle(Line(X((1)(1), X(1)(2)), (X(2)(1), X(2)(2))))$$

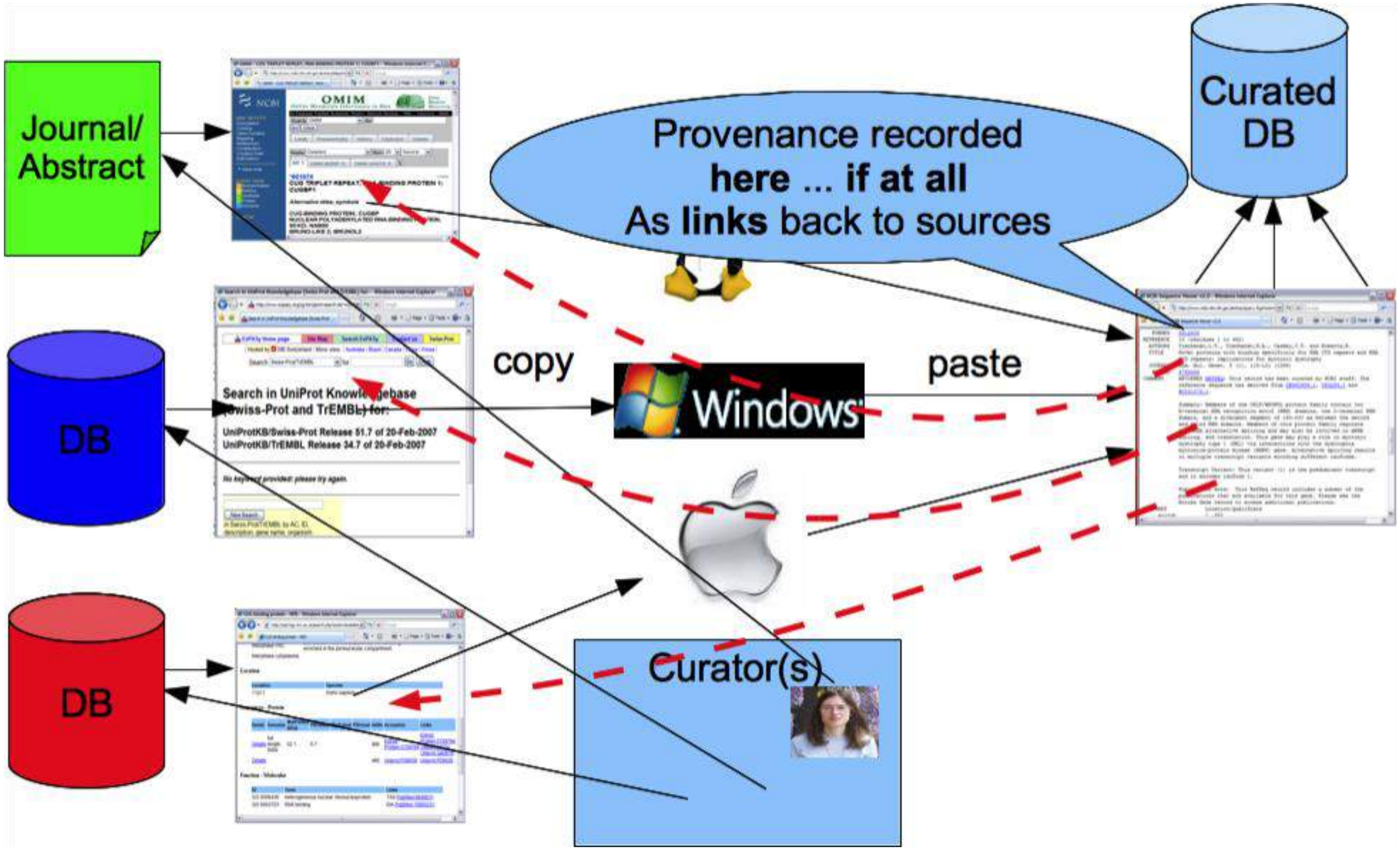
- Characterised in Mathematica as:

```
PropertyDiag[x1_, y1_, x2_, y2_] :=
  TwoPointMCCCircle[x1, y1, x2, y2] ==
    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 programing language for Web + database applications
  - Language-integrated **provenance** for queries
  - Language-integrated update via **Relational lenses**
  - Language-integrated **temporal queries** (versioning, time travel)

University of Edinburgh JA: Language-integrated query

National Physical Laboratory

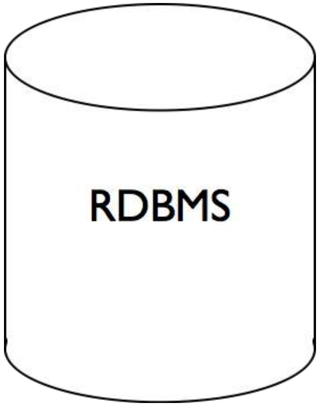
- Why?
  - Safety: avoid SQL injection attacks
  - Convenience: catch type errors early
  - Productivity: use general programming features in queries
- How?

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

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

```
["Bert",
"Drew",
"Erik",
"Gina"]
```

name
Bert
Drew
Erik
Gina



University of Edinburgh JA: Language-integrated query

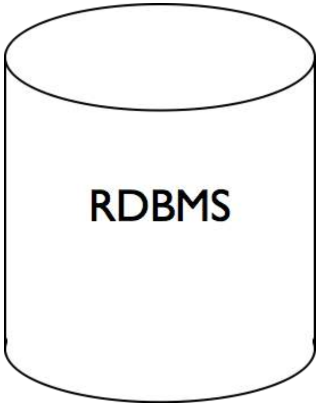
- Why?
  - Safety: avoid SQL injection attacks
  - Convenience: catch type errors early
  - Productivity: use general programming features in queries
- How?

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

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

```
[“Bert”,
“Drew”,
“Erik”,
“Gina”]
```

name
Bert
Drew
Erik
Gina



References

5. Kelly, G, **Formal specification in VDM-SL of the secure EDIFACT reference implementation. NPL Report. CISE 12/97.** 1997. Retrieved 3rd April 2021 from NPL https://eprintspublications.npl.co.uk/607/
6. ISO/IEC 10026-1:1998 **Information technology — Open Systems Interconnection — Distributed Transaction Processing — Part 1: OSI TP Model.** 1992 Retrieved 3rd April 2021 from ISO: https://www.iso.org/standard/27614.html
7. Wichmann, B, **A personal view of Formal Methods.** March 2000. Retrieved 3rd April 2021 from NPL http://resource.npl.co.uk/docs/science_technology/scientific_computing/ssfm/documents/baw_fm.pdf
8. Wikipedia. Retrieved 3rd April 2021 https://en.wikipedia.org/wiki/OSI_model

References

9. Barker, R, Brady, F, **Experiences in the use of formal methods in the standardisation of a complex OSI protocol. NPL Report. CISE 1/95.** 1995. Retrieved 3rd April 2021 from NPL https://eprintspublications.npl.co.uk/414/
10. ISO/IEC 10026-3:1992 **Information technology — Open Systems Interconnection — Distributed Transaction Processing — Part 3: Protocol specification**. ISO/IEC 10026-3:1992. 1992. Retrieved 3rd April 2021 from ISO: https://www.iso.org/standard/17981.html
11. ISO/IEC 15437:2001 **Information technology — Enhancements to LOTOS (E-LOTOS). 1992**. Retrieved 3rd April 2021 from ISO: https://www.iso.org/standard/27680.html
12. Milner, R, **Communication and Concurrency.** Prentice Hall 1989, ISBN 0 13 114984 9
13. Hoare, C. A. R, **Communicating Sequential Processes.** Prentice Hall 1985, ISBN 0 13 153271 5

References

14. Logrippo, L, Faci, M, Haj-Hussein, M, **An introduction to LOTOS: learning by examples.** Computer Networks and ISDN Systems Volume 23, Issue 5, February 1992, Pages 325-342
15. NI LabVIEW. Retrieved 3rd April 2021 from NI https://www.ni.com/en-gb.html
16. Charles, N, Bowman, H, Thompson, S, **From ACT-ONE to Miranda, a Translation Experiment.** Computer Standards and Interfaces, 19 (1). pp. 31-49. ISSN 0920-5489. 1997. Retrieved from University of Kent 3rd April 2021 https://kar.kent.ac.uk/21506/
17. Austin, S, Parkin, G, **Formal Methods: A Survey.** 1993. Contact keith.lines@npl.co.uk
18. ISO/IEC 13568:2002 **Information technology — Z formal specification notation — Syntax, type system and semantics.** Retrieved 3rd April from ISO https://www.iso.org/standard/21573.html

References

19. Lines, K, Smith, I, **Determining the quality of mathematical software using reference data sets.** BCS SQM Conference XXV. 2017. Retrieved 3rd April 2021 from Southampton Solent University https://ssudl.solent.ac.uk/id/eprint/3572/1/SQM%201-2%20Lines%20Smith.pdf
20. Harry, A, **Formal Methods Fact File VDM and Z.** John Wiley 1996 ISBN 0 471 94006 2
21. ISO 80000-1:2009 **Quantities and units — Part 1: General.** 2009. Retrieved 3rd April 2021 from ISO: https://www.iso.org/standard/30669.html
22. Delligatti, L, **SysML Distilled: A Brief Guide to the Systems Modeling Language.** Pearson 2014, ISBN 0 321 92786 9

To be continued…

NPL National Physical Laboratory

bcs The Chartered Institute for IT

Membership Get qualified Events Policy & influence Develop your people Deliver & teach qualifications More

Events > Event

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

Date and time

15 June, 5:15pm - 8:00pm

Location

Webinar

Price

Free

Tuesday 15 June 5:15 pm – 8:00 pm