

1-1

Introduction

1-2

abstract *adj.* to do with or existing in thought rather than matter.
v. take out of, extract, remove; summarize.
abstraction *n.* the act or an instance of abstracting or taking away; an abstract or visionary idea.

1-3

Example

When the first map of the London Underground was published in 1908, it was faithful to the geography of the lines: all the twists and turns of the tracks and the relative distances between stations were recorded faithfully and to scale.

However, the purpose of the map was to show travellers the order of stations on each line, and the various interchanges between lines; the fidelity of the map made it difficult to extract this information.

1-4



1-5

In 1933, the map was replaced by a more abstract representation, called the Diagram, which showed only the connectivity of stations.

Abstracted were:

- surface detail
- distances between stations
- orientation of lines

1-6



1-7

The Diagram gives people a good conceptual model; it is how we see the London Underground network. It is a specification that allows people to make sense of a complex implementation. Furthermore, although revised regularly to reflect changes in the network, it is still the same diagram proposed in 1931 by engineering draughtsman Harry Beck.

1-8

- The success of the diagram is due to
- an appropriate choice of abstraction
 - an elegant presentation

1-9

Qualities

A good specification should be

- abstract and complete
- clear and unambiguous
- concise and comprehensible
- easy to maintain and cost-effective

Above all, it should be useful.

1-10

Software

Existing specifications are extensive, but

- relevant information is hard to find
- different developers may have different interpretations
- there is no objective test of contract fulfilment
- design flaws are difficult to detect

1-11

Formal methods

Techniques based upon mathematics can be used at every stage of software development.

Examples include: probability theory; relational calculus; context-free grammars.

1-12

Why mathematics?

- abstraction vs confusion
- precision vs ambiguity
- reason vs doubt

1-13

Why not mathematics?

- mathematics is seen as difficult
- abstraction is rarely taught

1-14

Example

The following problem, originally formulated by two psychologists—Daniel Kahneman and Amos Tversky—was quoted by Keith Devlin in the *Guardian*, 29th May 1997.

1-15

There has been a hit-and-run involving a taxi. There are two taxi companies in town: Blue Cabs, with 15 cars, and Black Cabs, with 85. All the taxis were on the streets: we have no other information about their whereabouts at the time of the accident. It was dark, and there was a single witness, who believes that the taxi in question was blue. In tests, the witness correctly identified the colour of four out of every five taxis: on the other occasions, they mistakenly thought that a blue taxi was black, or vice versa. What is the probability that a blue taxi was involved?

1-16

Hint

The answer is not '80%'.

1-17

There are two possibilities to consider, given that the witness believes that the taxi was blue:

- the taxi at the location was blue, and the witness correctly identified its colour ($0.15 * 0.8 = 0.12$)
- the taxi at the location was black, and the witness incorrectly identified its colour ($0.85 * 0.2 = 0.17$)

The probability that the taxi was blue is $0.12/0.29$. There is only a 41% chance of this, all other things being equal.

1-18

Aside

In conditional probability,

$$\begin{aligned}
 & P(\text{car is blue} \mid \text{witness says blue}) \\
 &= \frac{P(\text{car is blue} \cap \text{witness says blue})}{P(\text{witness says blue})} \\
 &= \frac{0.12}{0.29} \\
 &= 0.41
 \end{aligned}$$

1-19

Response

The following letter was published in the *Guardian*, 5th June, 1997, under the heading

Maths sucks

1-20

"MATHEMATICS tells us that even in the highly simplified circumstances of artificial examples, people are notoriously fallible", writes Keith Devlin. Fortunately, most of us don't live in the world of simplified circumstances and, in this world, evolution has given us a more reliable way of reasoning, which I'd trust over maths any day.

Barry Brown

barry@soc.surrey.ac.uk

1-21

Lesson

Abstract representations can form the basis of excellent specifications.

Some people are afraid of abstraction. Some people can't see the wood for the trees.

1-22

Expectations

Formal methods may be abstract, universal tools, but they are still only tools. They must be used properly if they are to be effective.

Furthermore, some knowledge and intuition about the system will be required.

1-23

Description

Formal specifications cannot replace knowledge. However, we can use an abstract representation as a basis for discussion. The lack of ambiguity makes consensus harder to achieve, but more valuable.

1-24

Reasoning

Logical argument cannot replace intuition. However, we can use logic to break a problem down—to factorise it—so that it can be solved in several small steps, rather than one giant leap.

1-25

The Z Notation

- a mathematical language of logic, sets, and relations;
- a schema language of patterns and objects;
- a theory of refinement between abstract data types.

1-26

Courses

- Software Engineering Mathematics
- Specification and Design
- Advanced Software Development

1-27

Description

We can use Z to:

- describe data structures;
- model system state;
- formalise properties.

1-28

Reasoning

We can use Z to:

- explain design intentions;
- verify development steps;
- compare descriptions at different levels of abstraction.

1-29

Case study

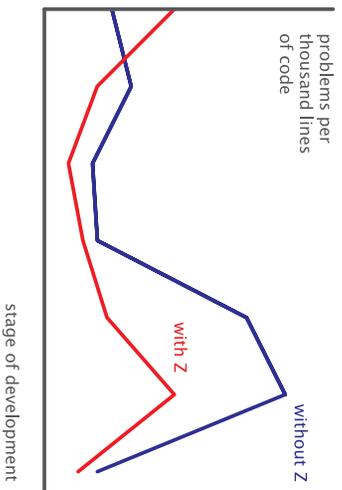
- collaboration between IBM UK Laboratories and the Programming Research Group, Oxford
- applying mathematical techniques to the development and design of new CICS modules
- providing education and consultancy—technology transfer from academia to industry
- providing an opportunity to evaluate methods in an industrial context

1-30

Subjective results

- initial cost in education
- more time spent on earlier stages; less time spent on coding
- increased precision in the use of natural language
- quality of work improved
- increased confidence in code

Qualitative results



1-31

Essential ingredients

- appropriate education
- employee motivation
- management support

1-32

Benefits

The production of a formal specification helps us to:

- understand requirements
- clarify intentions

The construction of a proof helps us to:

- identify assumptions
- explain correctness

1-33

A choice of methods

The Z notation is used to model systems in terms of state: we describe the state of the system, and explain the relationship between this and the state of various components.

Other notations, such as CSP, are used to model systems in terms of their communicating behaviour. The two notations can be used together in a description of the same system.

1-34

Formal description techniques

The formal description techniques—SDL, LOTOS, and ESTELLE—are more concrete than either Z or CSP. As such, they are more useful for description than for reasoning; nevertheless, they can prove extremely valuable in an industrial context.

All of the lessons learnt in Z or CSP can be applied in SDL, LOTOS, or ESTELLE.

1-35

Summary

- abstraction
- mathematics
- expectations
- the Z notation
- benefits

1-36