zühlke
empowering ideas

Nothing New Under the Sun

Techniques that still work no matter how hard we try to forget them

Keith Braithwaite

Context

zühlke
empowering ideas

"I come as an entertainer, not as a salesman. I want you to enjoy these ideas because I enjoy them"—

Alan Watts

Keith Braithwaite

If IT were a person…

zühlke
empowering ideas

It would be diagnosed with

- ADHD
- Retrograde amnesia
- OCD

Keith Braithwaite

If IT were a person…

zühlke
empowering ideas

It would be diagnosed with

- ADHD
  - We have difficulty retaining focus on the job at hand
  - We are very easily distracted by *ooh, shiny!*
- Retrograde amnesia
- OCD

Keith Braithwaite

If IT were a person…

zühlke
empowering ideas

It would be diagnosed with

- ADHD
  - We have difficulty retaining focus on the job at hand
  - We are very easily distracted by
- Retrograde amnesia
  - We don't recall our past
- OCD

Keith Braithwaite

If IT were a person…

zühlke
empowering ideas

It would be diagnosed with

- ADHD
  - We have difficulty retaining focus on the job at hand
  - We are very easily distracted by
- Retrograde amnesia
  - We don't recall our past
  - We don't recall our past
- OCD

Keith Braithwaite

If IT were a person…

zühlke
empowering ideas

It would be diagnosed with

- ADHD
  - We have difficulty retaining focus on the job at hand
  - We are very easily distracted by
- Retrograde amnesia
  - We don't recall our past
  - We don't recall our past
- OCD
  - We follow rituals independent of their effectiveness

Keith Braithwaite

If IT were a person…

zühlke
empowering ideas

It would be diagnosed with

- ADHD
  - We have difficulty retaining focus on the job at hand
  - We are very easily distracted by
- Retrograde amnesia
  - We don't recall our past
  - We don't recall our past
- OCD
  - We follow rituals independent of their effectiveness

Keith Braithwaite

Tony Hoare said…

zühlke
empowering ideas

"If we could only learn the right lessons from the successes of the past we would not need to learn from the failures "

Keith Braithwaite

Zombies

zühlke
empowering ideas

www.flickr.com/photos/ hryckowian/3540744713/

Nothing New Under the Sun
Slide 10
21 July 2009

Keith Braithwaite

Zombies

zühlke
empowering ideas

Given half a chance they *will* eat your brain

- Code the works “first time”
- Structured Programming

These, and others, we should forget

Keith Braithwaite

Code that works “first time”

City and Guilds COBOL

- 3 attempts to compile, run and test or fail

There was a time when this sort of thing made sense

Keith Braithwaite

Code that works “first time”

zühlke
empowering ideas

There was a time when this sort of thing made sense

Keith Braithwaite

Code that works “first time”

zühlke
empowering ideas

Jerry Weinberg tells of being told that

- The computer (singular) earns more than you do, so behave accordingly

Keith Braithwaite

Code that works “first time”

The computer learns more than you, behave accordingly

- cost(processor time) >> cost(developer time)
- Cycle time to get feedback—hours to days

Keith Braithwaite

Code that works “first time”

The computer learns more than you, behave accordingly

- cost(processor time) >> cost(developer time)
- Cycle time to get feedback—hours to days

In fact, you earn much more than the computer

Keith Braithwaite

Code that works “first time”

zühlke
empowering ideas

The computer learns more than you, behave accordingly

- cost(processor time) >> cost(developer time)
- Cycle time to get feedback—hours to days

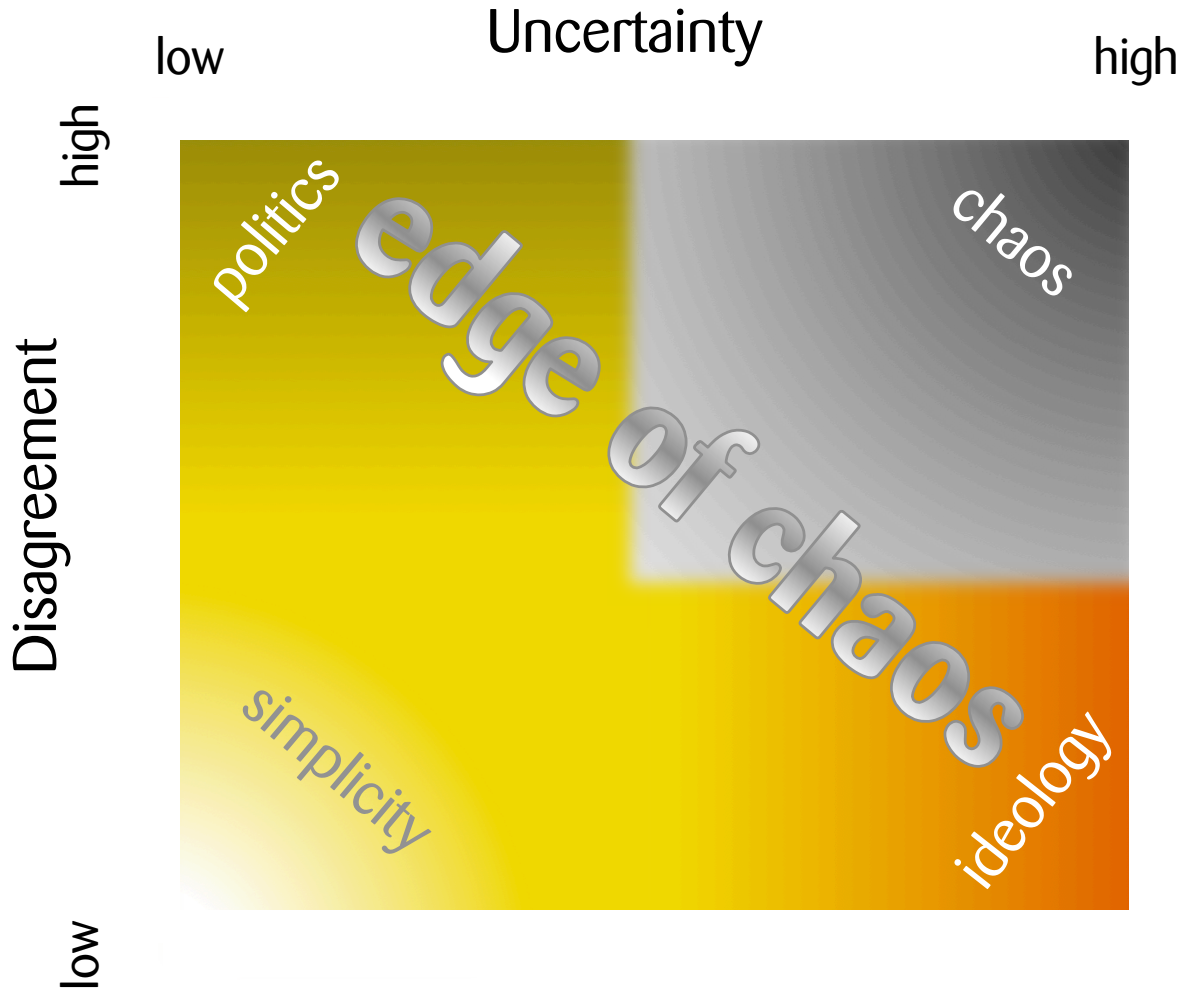
You earn much more than the computer, behave accordingly

- cost(processor time) << cost(developer time)
- Cycle time to get feedback—milliseconds to minutes
- A top-end dev workstation amortised over 3 years
  - £1 per day
  - 2 or 3 *orders of magnitude* cheaper than a programmer

behave accordingly

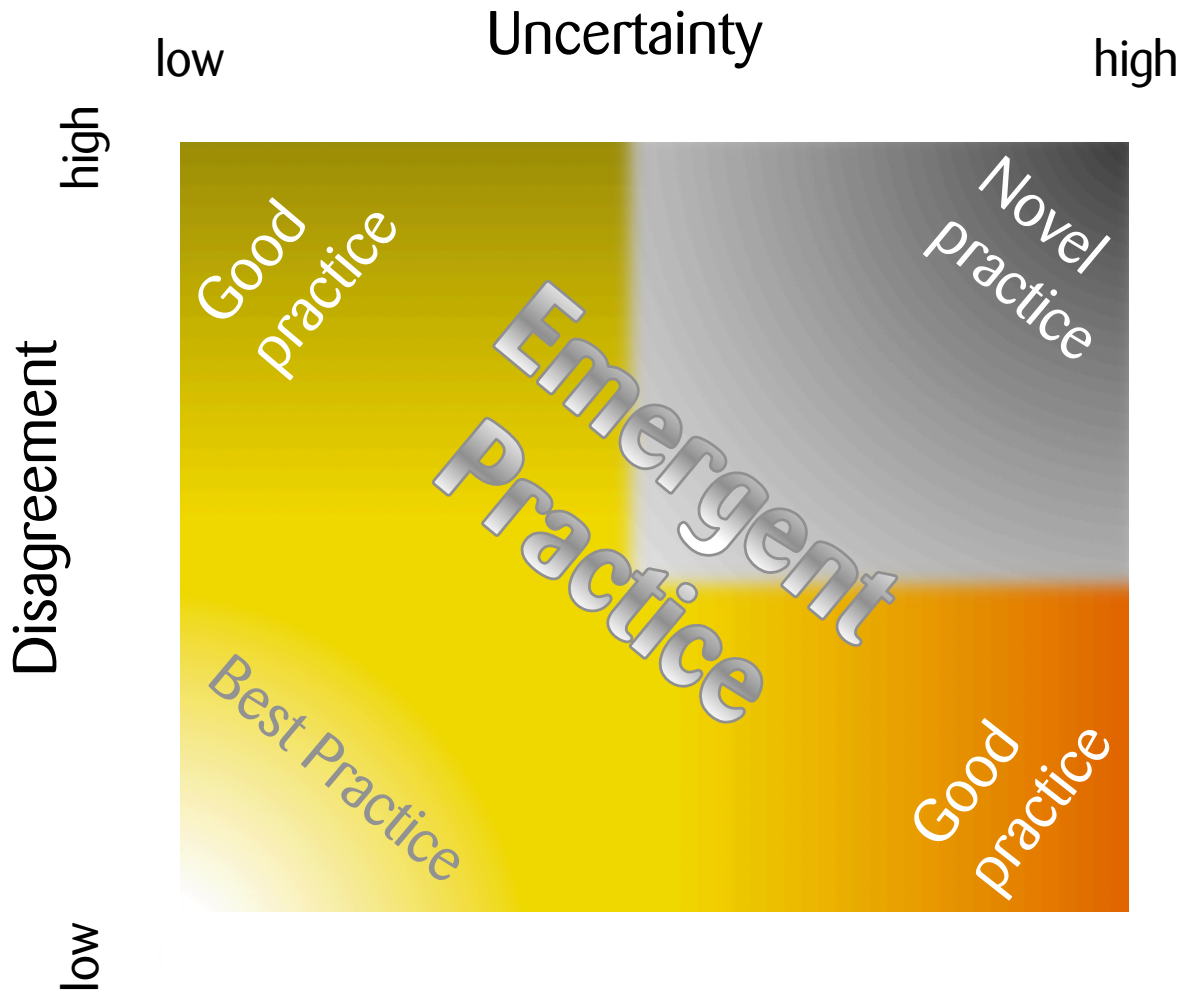
Keith Braithwaite

Code that works “first time”



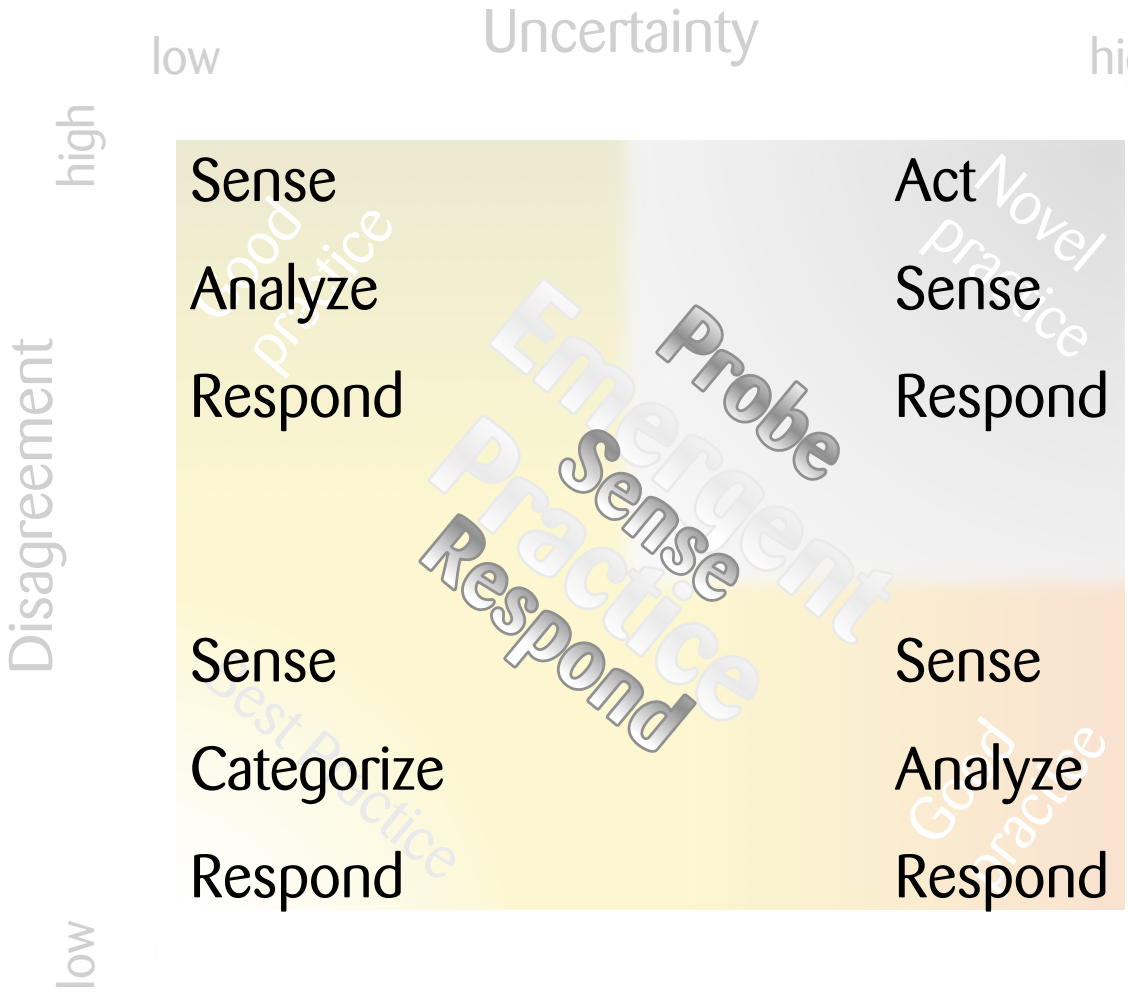
Keith Braithwaite

Code that works “first time”



Keith Braithwaite

Code that works “first time”



Nothing New Under the Sun
Slide 20
21 July 2009

Keith Braithwaite

Old School: things we got right

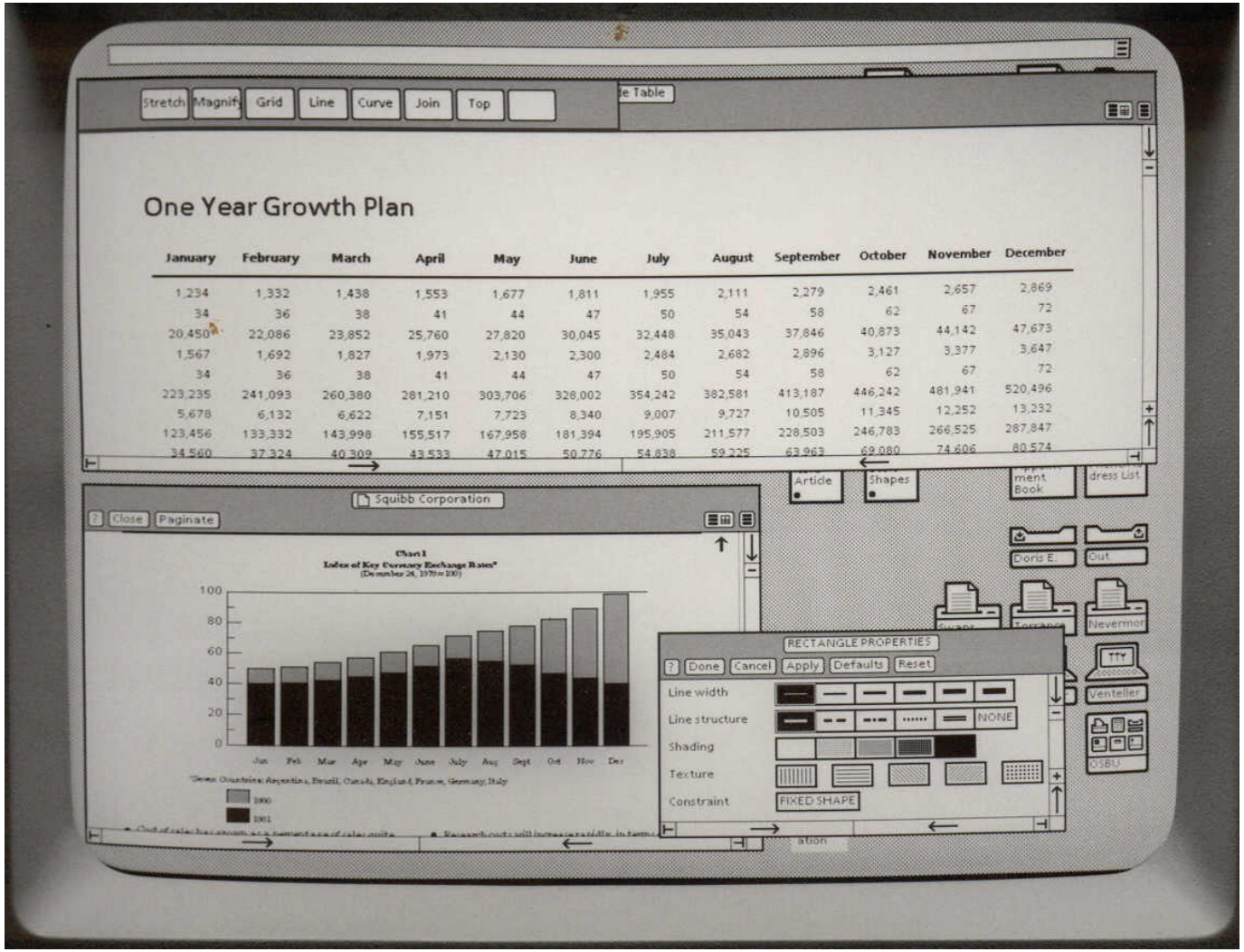
zühlke
empowering ideas

http://commons.wikimedia.org/wiki/File:Vincent_Series_C_Black_Shadow_1950.jpg

Keith Braithwaite

Old School: things we got right

zühlke
empowering ideas



http://www.digibarn.com/collections/screenshots/xerox-star-8010/xerox-star-8010-02.jpg

Keith Braithwaite

Old School: things we got right

zühlke
empowering ideas

Analysis

Architecture

Modelling

http://commons.wikimedia.org/wiki/File:Vincent_Series_C_Black_Shadow_1950.jpg

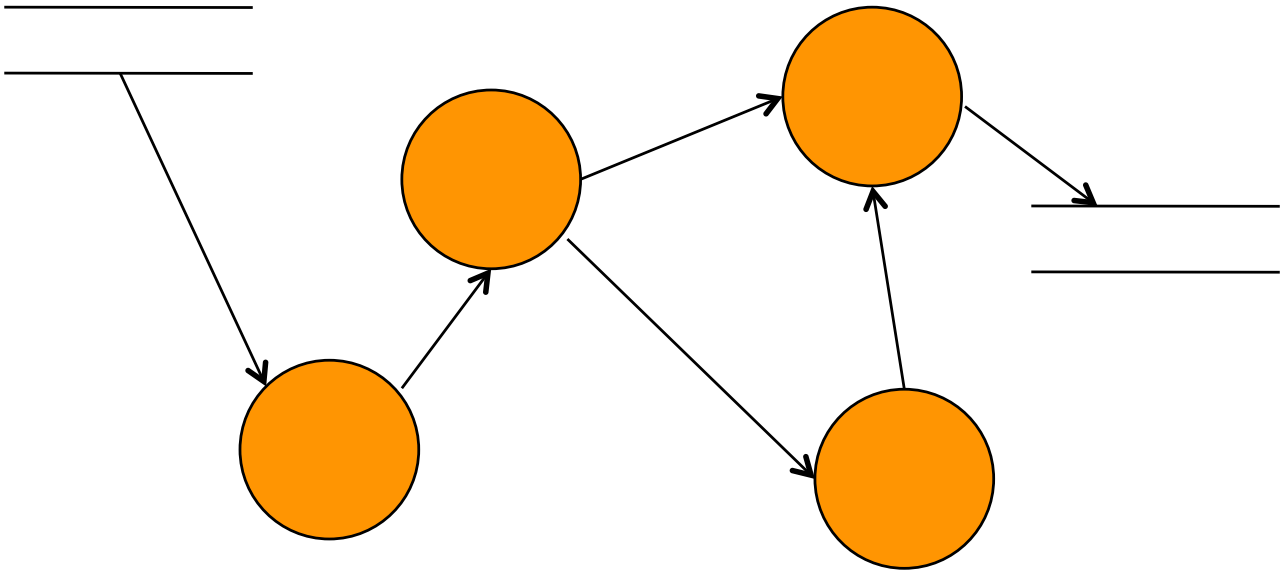
Keith Braithwaite

Analysis

zühlke
empowering ideas

There used to be this thing called Systems Analysis

- It used to be a core skill
- But it got a bad name



Keith Braithwaite

zühlke
empowering ideas

So, we stopped doing it

- Agile gave some of us an excuse

Keith Braithwaite

Analysis

zühlke
empowering ideas

We had to re-invent *understanding*

- Behaviour Driven Development
  - (AKA TDD the way you were always supposed to do it)
- Domain Driven Design
  - “Until I started working in "enterprise IT" I didn't realize that people *didn't* do this. I suppose that this is an important book, but it's depressing that this is so”—Nat Pryce

Keith Braithwaite

zühlke
empowering ideas

Domain Driven Design

"Leading software designers have recognized domain modeling and design as critical topics for at least 20 years, yet surprisingly little has been written about what needs to be done or how to do it."—Evans

Keith Braithwaite

Syntropy

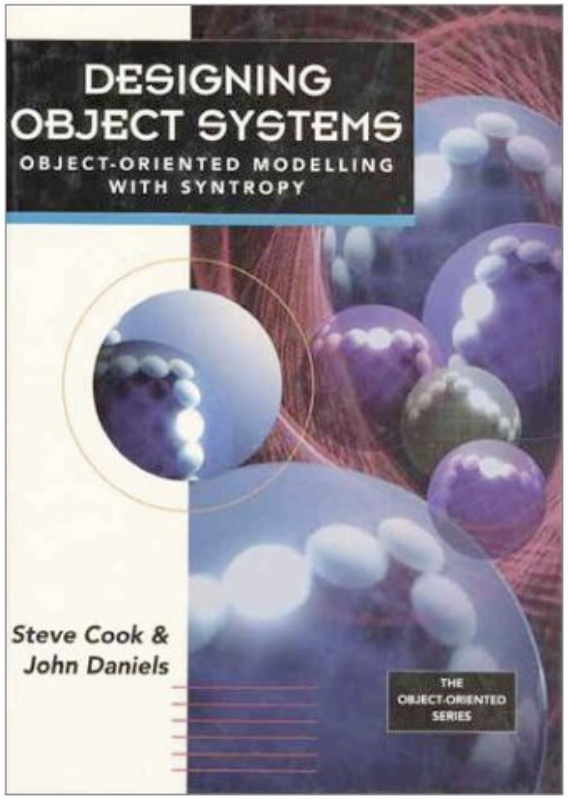
zühlke
empowering ideas

Three Perspectives:

- Essential
- Specification
- Implementation

Three Kinds of Domain

- Concept
- Interaction
- Infrastructure



http://www.syntropy.co.uk/syntropy/designing-object-systems.pdf

Keith Braithwaite

Syntropy: Essential Perspective

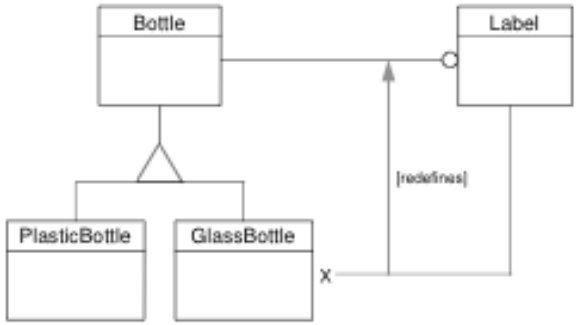
zühlke
empowering ideas

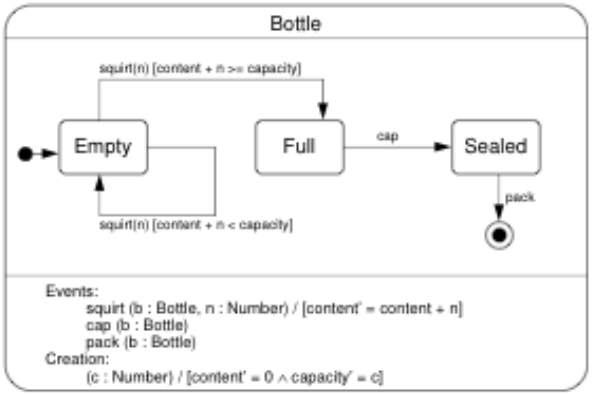
Essential models talk about

- (someone's understanding of)
- (a part of)
- (a situation in)

The World

- Objects with observable state, identity
- Globally broadcast, instantaneous events
- Objects change state in response to events
  - Not "message sending"





Keith Braithwaite

Syntropy: Specification Perspective

zühlke
empowering ideas

Specification models talk about a (proposed) system

"At some point during a development (which may or may not be at the beginning), the interface at the boundary between the software and its *environment* must be specified, and the specification model provides a way to specify this interface precisely."—Cook and Daniels §6.1 emphasis in original

- Objects with observable state, identity
- Globally broadcast, instantaneous events
- Objects change state in response to events and generate new events

Keith Braithwaite

Syntropy: Implementation Perspective

zühlke
empowering ideas

Implementation models talk about design

- Objects with observable state, identity, and responsibilities
- Point-to-point messages
- Objects change state in response to messages and send new messages

Keith Braithwaite

Syntropy: Domains

zühlke
empowering ideas

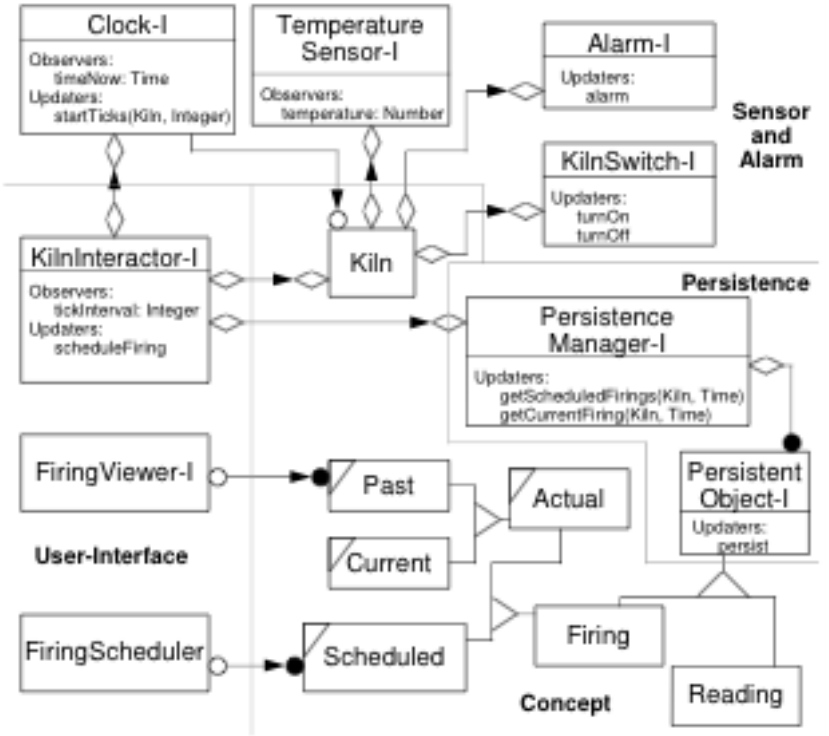
Three kinds of domain

- Interaction: at the system boundary
- Concept: what the system is about
- Infrastructure: How it works

Essential Perspective models concentrate on concept domains

Specification Perspective models have Interaction domains

Implementation Perspective models have all three



Kiln control system, Cook and Daniels fig 11.2

Keith Braithwaite

Syntropy: Relationship Between Models

zühlke
empowering ideas

"…the main correspondence between all three [perspectives] is in the ***type views for the concept domains.***" –Cook and Daniels §10.3 emphasis in original

"State diagrams do not, in general, correspond between essential and specification models." –***ibid***

Keith Braithwaite

Syntropy: The System Boundary

zühlke
empowering ideas

"If the software boundary is implicit in the situation itself, essential modelling may not be very helpful, because the essential model would express the same behaviour as the specification model without specifying which events are software-generated. However, when the software boundary is to be designed, an essential model provides a systematic way of designing it." —Cook and Daniels §10.5

Keith Braithwaite

Aside: Diagnostic Instrument

zühlke
empowering ideas

If everyone agrees that it's obviously correct:

- That the system has layers
- The one at the "top" is a website
- The one at the bottom is a database

Then:

- You probably don't have much of a domain
- Note that adding an ORM layer does not magically create a domain

Keith Braithwaite

Syntropy: Implications

zühlke
empowering ideas

There is no “seamlessness”

- From world to spec to system requires *translation*

You can’t sensibly use exactly one modelling language:

- To understand the world
- To specify a system
- To design an implementation

Keith Braithwaite

Problem Frames

zühlke
empowering ideas

"The problem is not at the computer interface—it is deeper into the world, further away form the computer"—Jackson *Problem Frames* §1.4

"Many descriptions that aught to be about the world […] are really about the computer and the software"—*ibid* §1.7

"Not everything worth thinking about has to result in a piece of program text or a designed domain. And conversely, the fact that there will be no model of [some things] doesn't mean that you don't have to describe them in your problem analysis" —*ibid* §2.

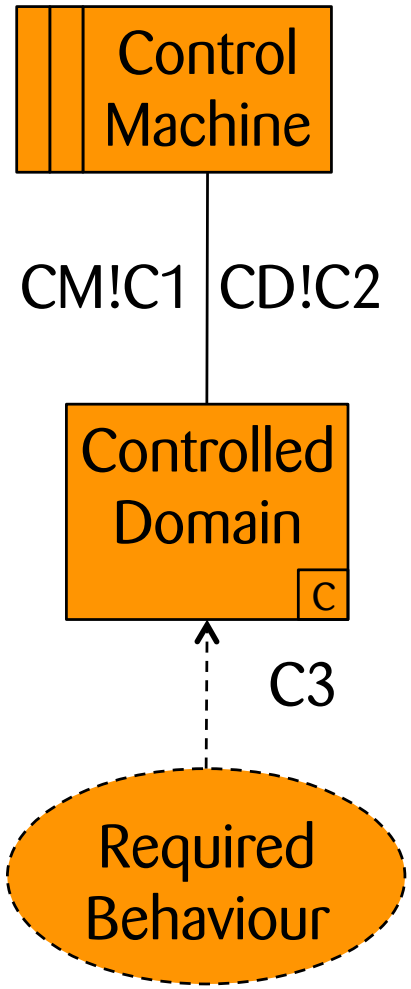
Keith Braithwaite

Problem Frame: Required Behaviour

zühlke
empowering ideas

- The Control Machine, which is to be built, directs Causal Phenomena (C1) at the Controlled Domain
- The Controlled Domain may respond with Causal Phenomena (C2) to provide feedback
- The Controlled Domain is a a Causal Domain, so we can predict how C2 depends on C1
- The Required Behaviour specifies some Causal Phenomena (C3) which must be satisfied if the Control Machine is to be correct

Example: timed traffic lights

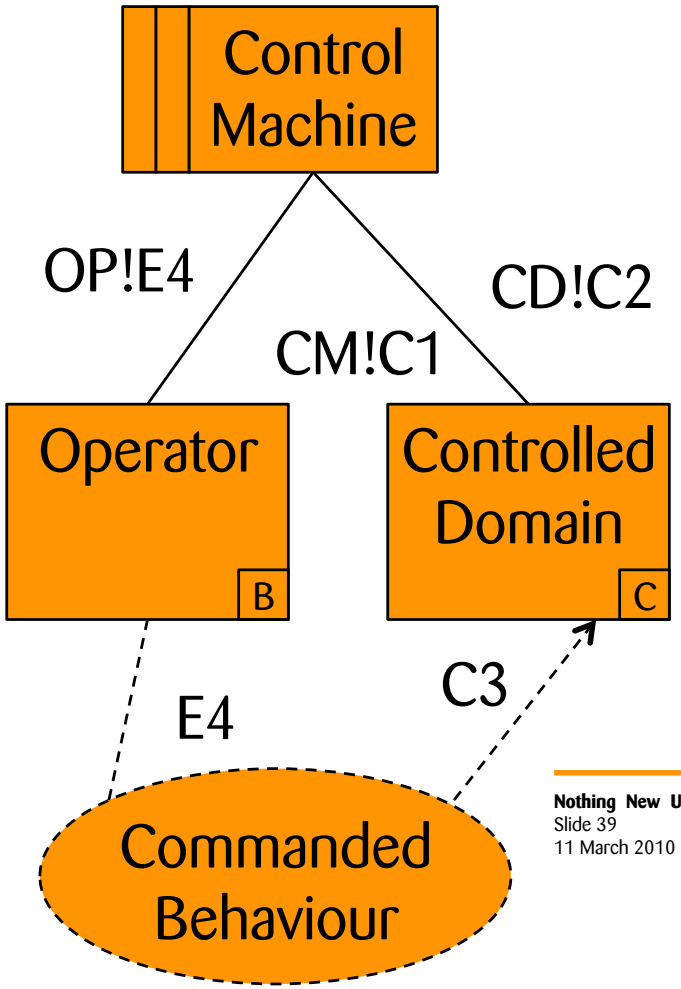


Keith Braithwaite

Problem Frame: Commanded Behaviour

zühlke
empowering ideas

- The Control Machine and Controlled Domain interact as in Required Behaviour
- Additionally an Operator observes events in the context of the Commanded Behaviour (such as E4) and sends commands to the control machine
- The Operator is a merely "biddable" domain, we cannot predict to a certainty how E4 and C3 are related
- **Example:** opening or closing a sluicegate under manual control

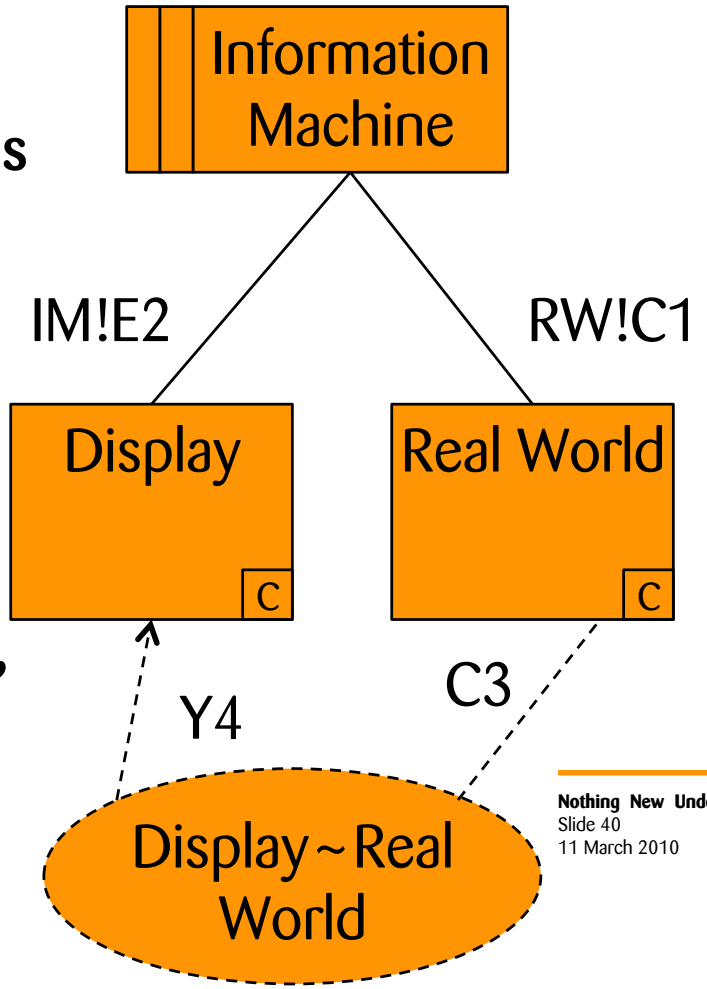


Keith Braithwaite

Problem Frame: Information Display

zühlke
empowering ideas

- The Real World is a Causal Domain, but an automomous, active one. It does it's own thing. It shares causal phenomena (such as C1) with the Information Machine.
- The Information Machine sends events (E2) to the Display (also a Causal Domain)
- The Display updates shared symbolic values (such as Y4) which must reflect phenomena C shared with the Real World, but which can only be inferred from C1
- **Example:** vehicle odometer



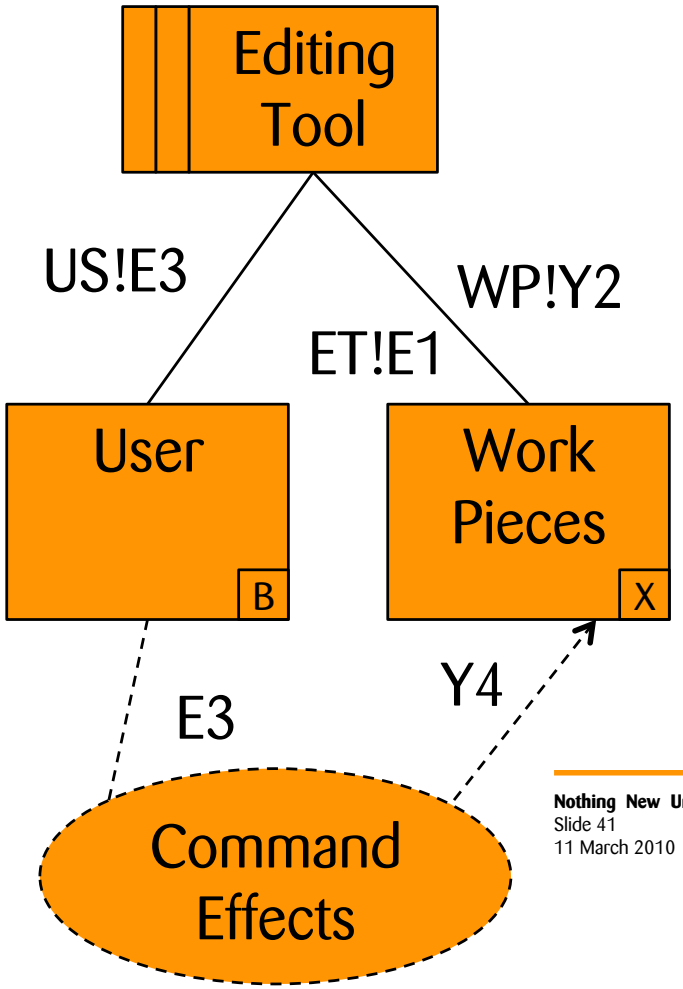
Keith Braithwaite

Problem Frame: Simple Workpieces

zühlke
empowering ideas

- The Editing Tool can send commands (in the form of events such as E1) to the work pieces.
- The workpieces are are leXical domain and may present symbolic phenomena (Y2) to the Editing Tool.
- The User may send events (E3) reflecting their desires for the workpieces.

Example: a party planning tool



Keith Braithwaite

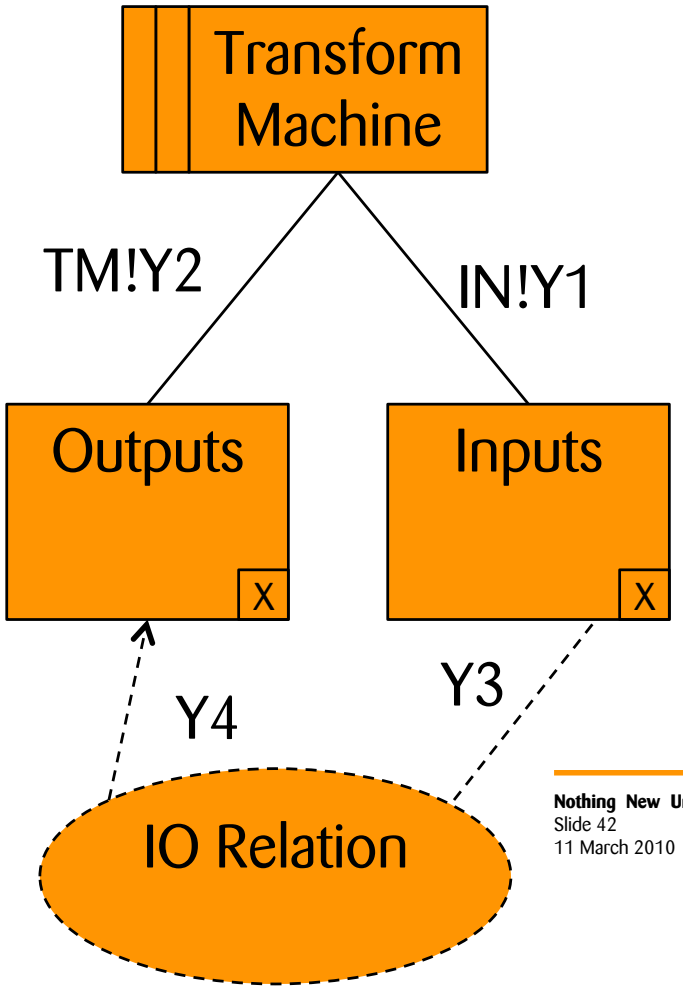
Problem Frame: Transformation

zühlke
empowering ideas

The Transform Machine notes symbolic phenomena (such as Y1) as input and emits symbolic phenomena (such as Y2) as output.

Input and Output are both lexical domains.

Example: email statistics analyser



Keith Braithwaite

Problem Frames: Three Descriptions

zühlke
empowering ideas

Three different things, three different descriptions

- The requirement is in the world
- The specification is at the system interface
- The domains join them up

Real problems have multiple, overlapping frames

- The book contains much guidance on managing this

Keith Braithwaite

Problem Frames: The aspiration

zühlke
empowering ideas

"Ideally, each problem frame would be associate with a systematic method that is known to be effective for analyzing and solving any problem that fits the frame"

Keith Braithwaite

Re-invention

zühlke
empowering ideas

About 20 years ago a bunch of really smart people thought really hard about domains and their relationship to design.

They came up with a bunch of really good ideas, and about 15 years ago, wrote them down.

We had to re-invent the idea of understanding domains as a guide to building systems

Are we going to have to re-discover those good ideas, too?

Keith Braithwaite

Not Doing Your Homework

zühlke
empowering ideas

Distillation

$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

—Oliver Heaviside, 1884

These four equations, along with the definitions of their terms and the body of mathematics they rest on, express the entirety of classical nineteenth-century electromagnetism.

Keith Braithwaite

Not Doing Your Homework

zühlke
empowering ideas

Domain Driven Design

"Leading software designers have recognized domain modeling and design as critical topics for at least 20 years, yet surprisingly little has been written about what needs to be done or how to do it."—Evans

But what *has* been written contains valuable lessons

Maybe it would be worth looking…

Keith Braithwaite