

ALVARO VIDELA

LECTOR IN CODIGO

**EXPLORE THE RELATION BETWEEN THE
PROCESS OF WRITING COMPUTER
PROGRAMS WITH THAT OF WRITING
LITERARY WORKS OF FICTION.**

UMBERTO ECO

LECTOR IN FABULUA

SIX WALKS IN THE FICTIONAL WOODS

**WHAT CAN WE LEARN FROM
THESE THEORIES TO BECOME
BETTER* PROGRAMMERS**

**WHAT CAN WE LEARN FROM
THESE THEORIES TO BECOME
BETTER* PROGRAMMERS?**

WHAT A PROGRAMMER DOES

It has been believed that a programmer occasionally writes code and gets it running on a computer, and that this is what he is paid for. In spite of his obvious inefficiency, no one else seems to do this work more effectively. However, his activity is still observed principally as loafing—a kind of ritual (like the British and teatime) which must be put up with.

Another view of what a programmer does addresses more constructively all that "wasted" time and

cludes more than the running code, more than the symbolic code, or even the operator's guide, the maintenance guide, or the design guide. For in fact, in response to any serious breach of the program's integrity, a programmer will become involved, as part of the integral organization built by the original programmer. If one now looks closely, he can begin to recognize the intent of those steps in the ritual of programming.

WHAT A PROGRAMMER DOES

It has been believed that a programmer occasionally writes code and gets

cludes more than the running code, more than the symbolic code, or even

BEST UNKNOWN PAPER

company as learning a kind of ritual (like the British and teatime) which must be put up with.

Another view of what a programmer does addresses more constructively all that "wasted" time and

part of the integral organization built by the original programmer. If one now looks closely, he can begin to recognize the intent of those steps in the ritual of programming.

**“A PROGRAMMER DOES NOT
PRIMARYLY WRITE CODE; RATHER,
HE PRIMARYLY WRITES TO ANOTHER
PROGRAMMER ABOUT HIS
PROBLEM SOLUTION”**

**“PROGRAMS MUST BE
WRITTEN FOR PEOPLE TO
READ, AND ONLY INCIDENTALY
FOR MACHINES TO EXECUTE”**

THE USE OF SUB-ROUTINES IN PROGRAMMES

D. J. Wheeler

Cambridge & Illinois Universities

THE USE OF SUB-ROUTINES IN PROGRAMMES

D. J. Wheeler

Cambridge & Illinois Universities

The above remarks may be summarized by saying sub-routines are very useful-although not absolutely necessary-and that the prime objectives to be born in mind when constructing them are simplicity of use, correctness of codes and accuracy of description. All complexities should-if possible-be buried out of sight.

LITERATURE AND PROGRAMMING

LITERATE PROGRAMMING

Donald Knuth

“INSTEAD OF IMAGINING THAT OUR MAIN TASK IS TO INSTRUCT A COMPUTER WHAT TO DO, LET US CONCENTRATE RATHER ON EXPLAINING TO HUMAN BEINGS WHAT WE WANT A COMPUTER TO DO”

LITERATE PROGRAMMING

- ▶ Introduces the WEB system
- ▶ Write documentation along with the code
- ▶ Partially adopted by tools like JavaDocs and the like

**EXPLAINS HOW WEB WORKS,
BUT NOT HOW TO WRITE CODE
THAT'S EASIER TO UNDERSTAND**

CYBERTEXT: PERSPECTIVES ON ERGODIC LITERATURE

Aarseth, Espen J

**“[...] A SEARCH FOR LITERARY VALUE
IN TEXTS THAT ARE NEITHER INTENDED
NOR STRUCTURED AS LITERATURE
WILL ONLY OBSCURE THE UNIQUE
ASPECTS OF THESE TEXTS AND
TRANSFORM A FORMAL INVESTIGATION
INTO AN APOLOGETIC CRUSADE.”**

“PROGRAMS ARE NORMALLY WRITTEN WITH TWO KINDS OF RECEIVERS IN MIND: THE MACHINES AND OTHER PROGRAMMERS. THIS GIVES RISE TO A DOUBLE STANDARD OF AESTHETICS, OFTEN IN CONFLICT: EFFICIENCY AND CLARITY”

“A DIFFERENCE BETWEEN WRITING AND PROGRAMMING, [IS THAT] IN PROGRAMMING, THE PROGRAMMER GETS FEEDBACK VERY EARLY ON WHETHER THE PROGRAM TEXT IS EXECUTABLE, DURING COMPILING. FURTHERMORE, THEY GET FEEDBACK ON WHETHER THE PROGRAM IS WORKING AS INTENDED”

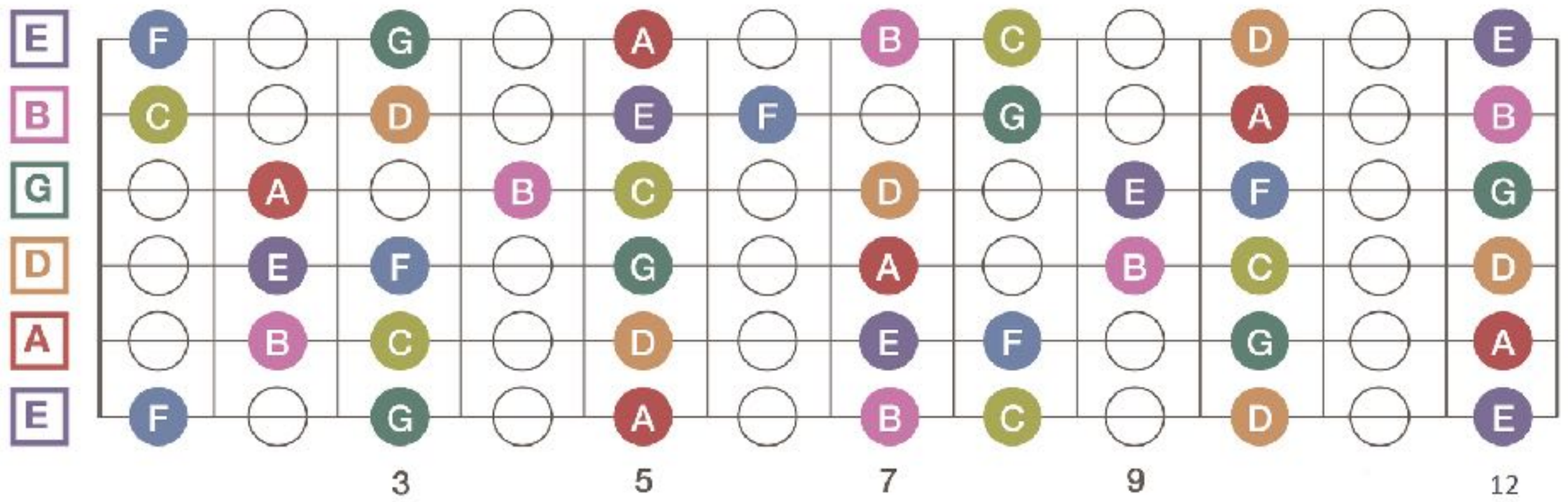
Hermans, Felienne, and Marlies Aldewereld

ABOUT EARLY FEEDBACK

- ▶ What does the program means?
- ▶ What process of the real world is trying to represent?
- ▶ How the problem was solved?

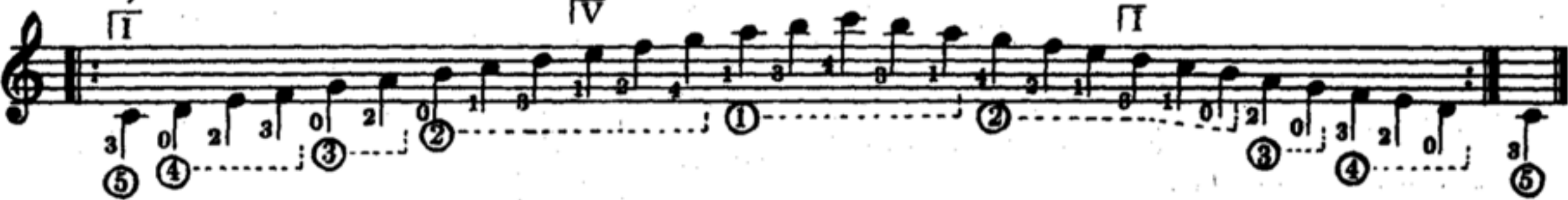
**COMPARE THIS WITH
MUSIC INTERPRETATION**

NOTES ON THE GUITAR

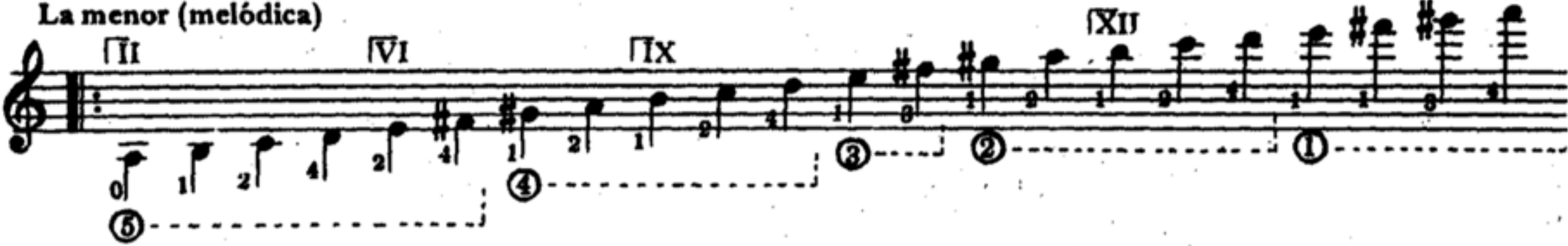


ABEL CARLEVARO

Do mayor



La menor (melódica)



The image displays two musical staves. The top staff is titled "Do mayor" and shows a scale with notes C, D, E, F, G, A, B, C. It includes Roman numerals I, IV, and VII above the staff. The bottom staff is titled "La menor (melódica)" and shows a scale with notes G, A, B, C, D, E, F, G. It includes Roman numerals II, VI, IX, XII, and XIV above the staff. Both staves feature fingerings (0-5) and circled numbers (1-5) indicating fret positions. The notation includes treble clefs, a key signature of one sharp (F#), and a common time signature (C).

**“CORRECT GUITAR PLAYING
IS UNCONCEIVABLE WITHOUT
CORRECT FINGERING”**

Abel Carlevaro

ABEL CARLEVARO

Do mayor

La menor (melódica)

The image displays two musical staves for guitar. The top staff is for the 'Do mayor' (C major) scale, starting on a C4 octave and ending on a C5 octave. It includes chord diagrams for I, IV, and I. The bottom staff is for the 'La menor (melódica)' (A minor melodic) scale, starting on an A4 octave and ending on an A5 octave. It includes chord diagrams for II, VI, IX, XII, and XIV. Both staves show the notes of the scale on a treble clef staff and the corresponding fingerings (0-4) and fret numbers (1-5) for each note. Dashed lines connect the fret numbers to the notes they apply to.

ABOUT EARLY FEEDBACK

- ▶ Knuth: Is 2 a random number?
- ▶ Is a square function that returns a hardcoded 25 a correct implementation?
- ▶ As long as we provide [5, -5] as arguments, it is correct.
- ▶ TDD advocates this kind of program building

**“PROGRAM TESTING CAN BE USED
TO SHOW THE PRESENCE OF BUGS,
BUT NEVER TO SHOW THEIR
ABSENCE!”**

Edsger Dijkstra

ABOUT EARLY FEEDBACK

- ▶ Knuth: Is 2 a random number?
- ▶ Is a square function that returns a hardcoded 25 a correct implementation?
- ▶ As long as we provide [5, -5] as arguments, it is correct
- ▶ TDD advocates this kind of program building
- ▶ QuickCheck tries to alleviate this problem

**HOW CAN WE SHARE
KNOWLEDGE BETWEEN
PROGRAMMERS?**

**“THE CODE SPEAKS
FOR ITSELF”**

**WE ARE NOT
ADVERSARIES**

**IMAGINE IF EVERY TIME WE
TRIED TO READ A BOOK, WE
HAD TO PLAY CODE BREAKERS?**

**UNLESS WE WERE
READING
FINNEGANS WAKE...**

PROGRAMMING AS THEORY BUILDING

Peter Naur

**“[...] A PERSON WHO HAS OR POSSESSES
A THEORY IN THIS SENSE KNOWS HOW TO
DO CERTAIN THINGS AND IN ADDITION CAN
SUPPORT THE ACTUAL DOING WITH
EXPLANATIONS, JUSTIFICATIONS, AND
ANSWERS TO QUERIES, ABOUT THE
ACTIVITY OF CONCERN”**

**“[...] WHAT HAS TO BE BUILT BY
THE PROGRAMMER IS A THEORY
OF HOW CERTAIN AFFAIRS OF THE
WORLD WILL BE HANDLED BY, OR
SUPPORTED BY, A COMPUTER
PROGRAM”**

**THIS THEORY IS VERY HARD
TO SHARE, IT WON'T BE
REFLECTED IN
DOCUMENTATION OR THE
PROGRAM TEXT**

**HOW CAN WE SHARE
THIS THEORY?**

THE

ENCYCLOPEDIA

THE ENCYCLOPEDIA

- ▶ There's the Encyclopedia
- ▶ And there's the "encyclopedia"
- ▶ All the world's knowledge vs. my knowledge

**“THE COMPETENCE OF THE
DESTINATARY IS NOT NECESSARILY
THAT OF THE SENDER”**

**ABSENCE OF
DETAILS**

**WE FILL IN DETAILS FROM
OUR OWN ENCYCLOPEDIA**

THE MODEL READER

MODEL READER

- ▶ Not the empirical reader
- ▶ Lives in the mind of the author (the empirical one)
- ▶ It's built as the author writes the story
- ▶ Helps the author decide how much detail to include in the story

NOTICE



**Dogs must
be carried
on escalator**

DOGS MUST BE CARRIED ON ESCALATOR

- ▶ Does it mean that you must carry a dog in the escalator?
- ▶ Are you going to be banned from the escalator unless you find a stray dog to carry?
- ▶ “Carried” is to be taken metaphorically and help dogs get through life?

DOGS MUST BE CARRIED ON ESCALATOR

- ▶ How do I know this is not a decoration?
- ▶ I need to understand that the sign has been placed there by some authority
- ▶ Conventions: I understand that "escalator" means *this* escalator and not some escalator in Paraguay
- ▶ "Must be" means must be *now*

TEXTUAL

COOPERATION

**A TEXT IS A LAZY (OR ECONOMIC)
MECHANISM THAT LIVES ON THE
SURPLUS VALUE OF MEANING
INTRODUCED BY THE RECIPIENT**

**“A TEXT WANTS SOMEONE
TO HELP IT WORK”**

**READING IS ESSENTIALLY A WORK
OF COOPERATION BETWEEN THE
AUTHOR AND THE READER**

**A STRATEGIC GAME
BETWEEN AUTHOR AND
READER**

NAPOLÉON VS WELLINGTON

DEVICES TO HELP PROGRAMMERS

- ▶ Type declarations
- ▶ Documentation
- ▶ Paratexts

PARATEXTS

"THE "PARATEXT" CONSISTS OF THE WHOLE SERIES OF MESSAGES THAT ACCOMPANY AND HELP EXPLAIN A GIVEN TEXT— MESSAGES SUCH AS ADVERTISEMENTS, JACKET COPY, TITLE, SUBTITLES, INTRODUCTION, REVIEWS, AND SO ON."

Eco quoting Genette

PARATEXTS IN CODE

- ▶ Documentation
- ▶ package names
- ▶ folder structure
- ▶ pragmas (as in Haskell)
- ▶ imports (hiding things from the Prelude or overloading it)
- ▶ compiler flags
- ▶ running mode (test, production, benchmarks)

**A PRIVILEGED PLACE OF A PRAGMATICS
AND A STRATEGY, OF AN INFLUENCE ON
THE PUBLIC, AN INFLUENCE THAT –
WHETHER WELL OR POORLY UNDERSTOOD
AND ACHIEVED – IS AT THE SERVICE OF A
BETTER RECEPTION FOR THE TEXT AND A
MORE PERTINENT READING OF IT**

Gérard Genette

**KEEPING PARATEXTS
RELEVANT**

**HOW TO KEEP
COMMENTS UP TO DATE?**

**NOT EVEN CERVANTES
ESCAPED THIS FATE**

**IN DON QUIXOTE, THE ORIGINAL
DESCRIPTION FOR CHAPTER X
DOESN'T MATCH THE CONTENTS OF
THE CHAPTER!**

**CONSIDER THIS
CODE**

```
class User {
    String username;
    String password;
    String role;

    User(String username, String password, String role) {
        this.username = username;
        this.password = password;
        this.role      = role;
    }

    public String getUsername() {return username;}
    public String getPassword() {return password;}
    public String getRole()     {return role;}
}
```

```
User user = new User('alice', 'secret', 'admin');
assertEquals(user.getUsername(), 'alice');
assertEquals(user.getPassword(), 'secret');
assertEquals(user.getRole(), 'admin');
```

**THE PREVIOUS TEST CAN GIVE US
FEEDBACK ABOUT THE CODE WORKING AS
EXPECTED, BUT WE ARE STILL IN THE DARK
ABOUT WHAT IS THIS CLASS PURPOSE, THAT
IS, WHAT CONCEPT OF THE REAL WORLD
THIS CLASS IS TRYING TO REPRESENT.**

```
class User {
    String username;
    String password;
    String role;

    User(String username, String password, String role) {
        this.username = username;
        this.password = password;
        this.role      = role;
    }

    public String getUsername() {return username;}
    public String getPassword() {return password;}
    public String getRole()     {return role;}
}
```

```
package database;
```

```
class User {  
    String username;  
    String password;  
    String role;
```

```
    User(String username, String password, String role) {  
        this.username = username;  
        this.password = password;  
        this.role      = role;  
    }
```

```
    public String getUsername() {return username;}  
    public String getPassword() {return password;}  
    public String getRole()     {return role;}  
}
```



```
package model;
```

```
class User {  
    String username;  
    String password;  
    String role;
```

```
    User(String username, String password, String role) {  
        this.username = username;  
        this.password = password;  
        this.role      = role;  
    }
```

```
    public String getUsername() {return username;}  
    public String getPassword() {return password;}  
    public String getRole()     {return role;}  
}
```

“TO INDICATE WHAT IS AT STAKE, WE CAN ASK ONE SIMPLE QUESTION AS AN EXAMPLE: LIMITED TO THE TEXT ALONE AND WITHOUT A GUIDING SET OF DIRECTIONS, HOW WOULD WE READ JOYCE'S ULYSSES IF IT WERE NOT ENTITLED ULYSSES?”

Gérard Genette

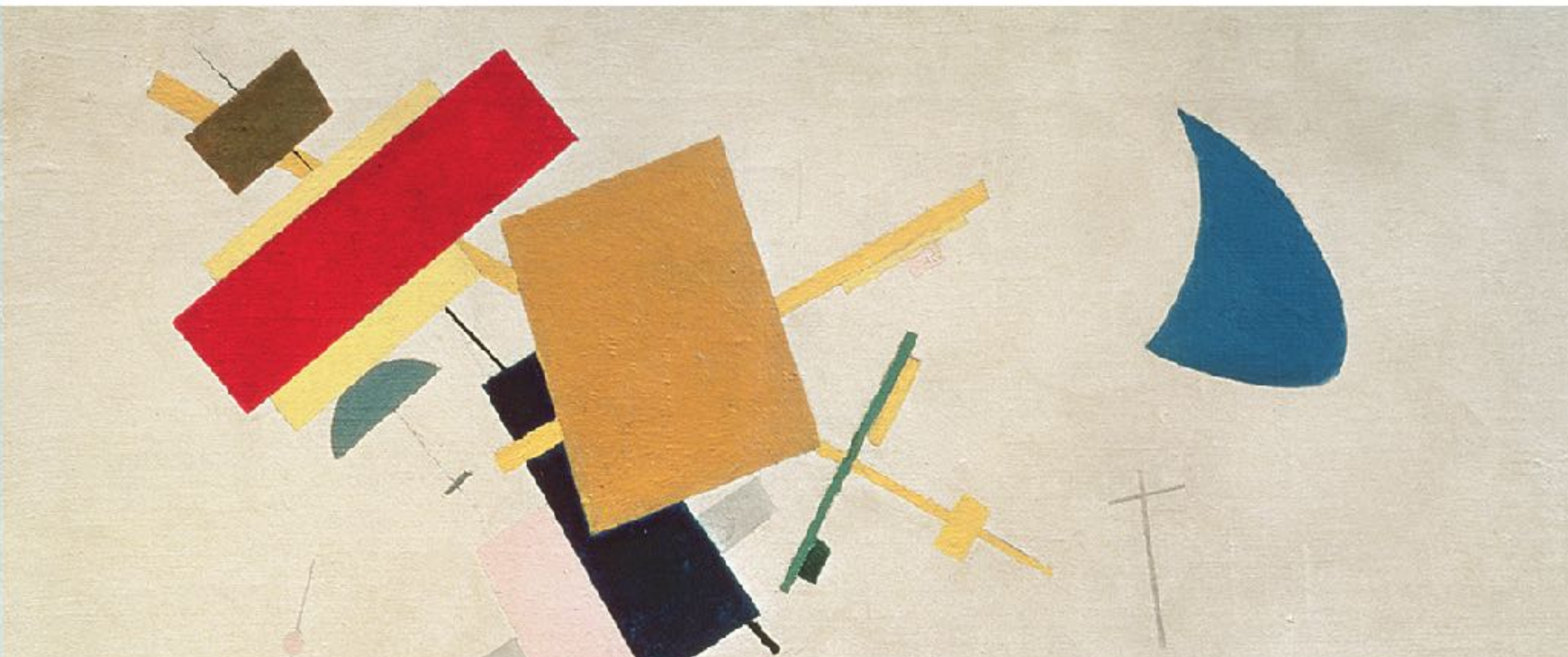
**HOW TO BUILD THE MODEL
READER FOR OUR CODE?**

METAPHORS

THE GEOMETRY OF MEANING

SEMANTICS BASED ON CONCEPTUAL SPACES

PETER GÄRDENFORS



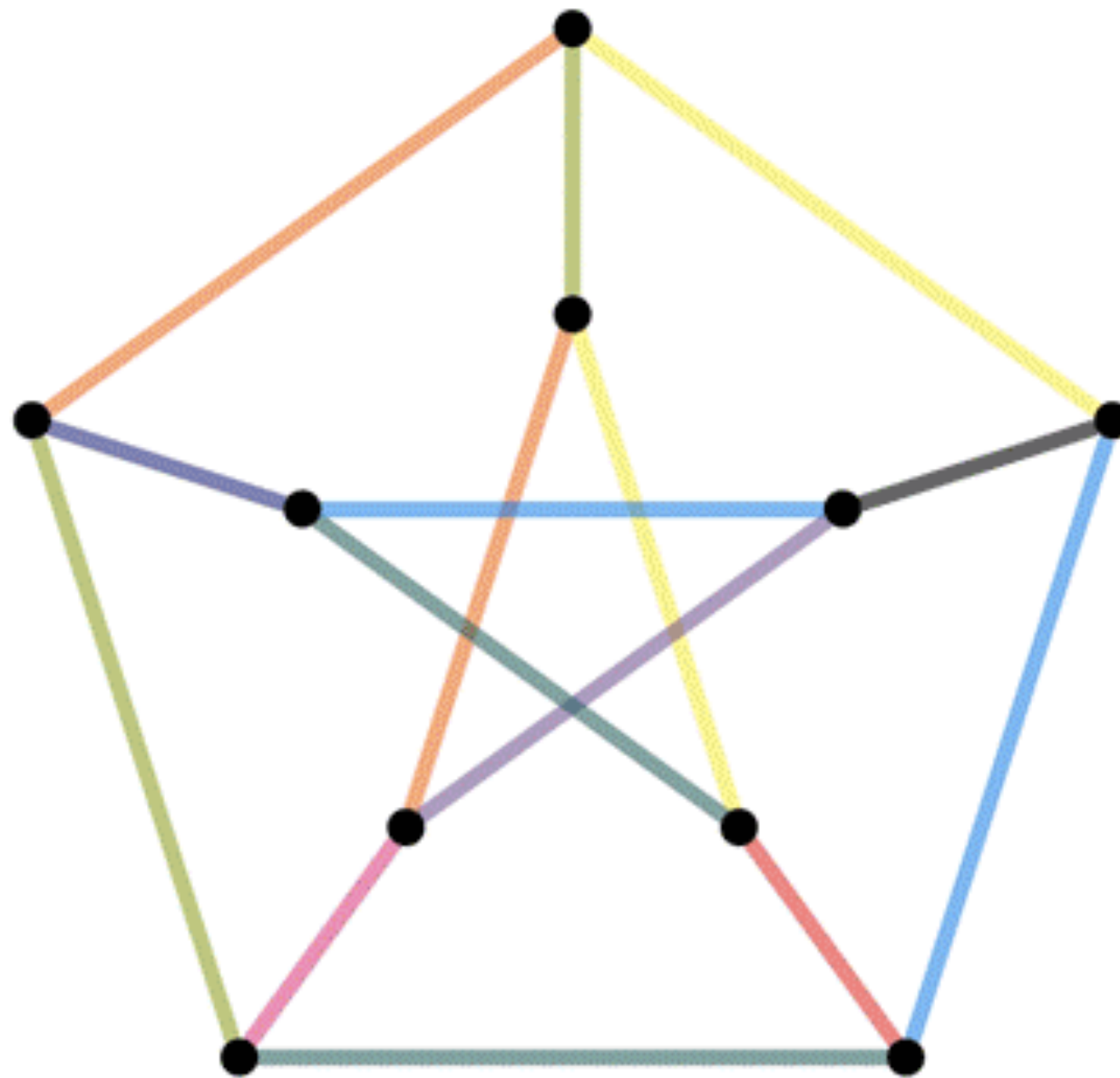
**METAPHORICAL
MAPPINGS PRESERVE THE
THE COGNITIVE TOPOLOGY
OF THE SOURCE DOMAIN**

**IN A WAY CONSISTENT
WITH THE INHERENT
STRUCTURE OF THE
TARGET DOMAIN**

**METAPHORS TRANSFER
INFORMATION FROM
ONE CONCEPTUAL
DOMAIN TO ANOTHER**

**WHAT IS TRANSFERRED
IS A PATTERN RATHER
THAN DOMAIN
SPECIFIC INFORMATION**

**A METAPHOR CAN THUS BE
USED TO IDENTIFY A
STRUCTURE IN A DOMAIN
THAT WOULD NOT HAVE BEEN
DISCOVERED OTHERWISE**



GRAPH ISOMORPHISM

THE SOCIAL CONSTRUCTION OF REALITY: A TREATISE IN THE SOCIOLOGY OF KNOWLEDGE

Berger, Peter L., and Thomas Luckmann

Microservices

a definition of this new architectural term

MICROSERVICES

25 March 2014



James Lewis

James Lewis is a Principal Consultant at ThoughtWorks and member of the Technology Advisory Board. James'

interest in building applications out of small collaborating services stems from a background in integrating enterprise systems at scale. He's built a number of

Contents

Characteristics of a Microservice Architecture

- Componentization via Services
- Organized around Business Capabilities
- Products not Projects
- Smart endpoints and dumb pipes
- Decentralized Governance
- Decentralized Data Management
- Infrastructure Automation
- Design for failure
- Evolutionary Design

MICROSERVICES

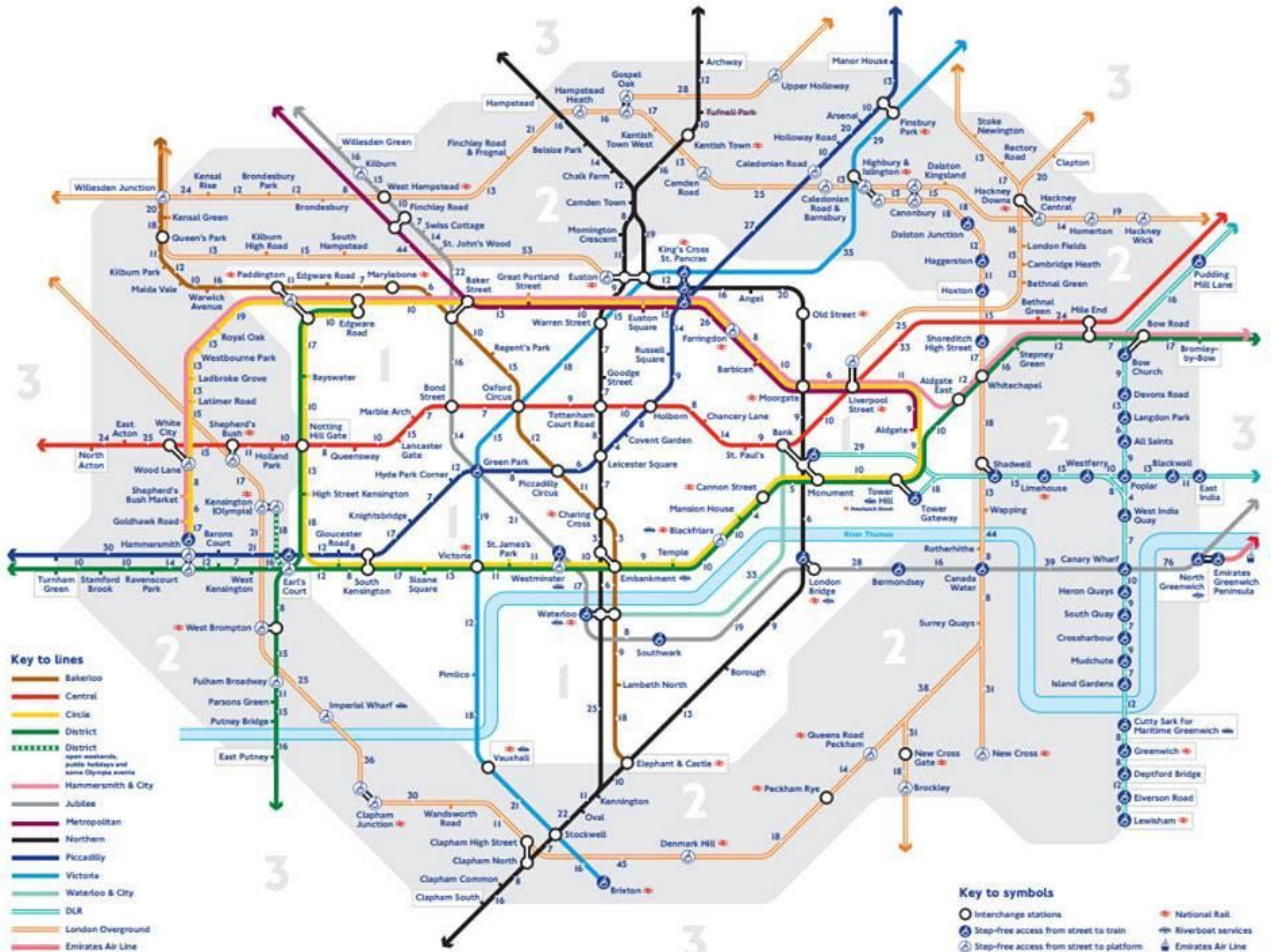
- ▶ Decentralised Governance
- ▶ Monolith vs. Microservice
- ▶ Isolation
- ▶ Collaboration
- ▶ Small is better - Big is cumbersome
- ▶ David vs. Goliath

ERLANG ANYONE?

“IN ANOTHER DIRECTION, ONE COULD ARGUE THAT MICROSERVICES ARE THE SAME THING AS THE ERLANG PROGRAMMING MODEL, BUT APPLIED TO AN ENTERPRISE APPLICATION CONTEXT”

**WHAT'S ERLANG'S
ELEVATOR PITCH?**

MAPS



ON BEAUTY

Noah Iliinsky

“[...] THAT FREED THE MAP OF ANY ATTACHMENT TO ACCURATE REPRESENTATION OF GEOGRAPHY AND LED TO AN ABSTRACTED VISUAL STYLE THAT MORE SIMPLY REFLECTED THE REALITIES OF SUBWAY TRAVEL: ONCE YOU’RE IN THE SYSTEM, WHAT MATTERS MOST IS YOUR LOGICAL RELATIONSHIP TO THE REST OF THE SUBWAY SYSTEM”

“THE FIRST AREA TO CONSIDER IS WHAT KNOWLEDGE YOU’RE TRYING TO CONVEY, WHAT QUESTION YOU’RE TRYING TO ANSWER, OR WHAT STORY YOU’RE TRYING TO TELL”

**“[...] THE NEXT
CONSIDERATION IS HOW THE
VISUALIZATION IS GOING TO BE
USED. THE READERS AND THEIR
NEEDS, JARGON, AND BIASES
MUST ALL BE CONSIDERED”**

**“THE READERS’ SPECIFIC
KNOWLEDGE NEEDS MAY NOT BE
WELL UNDERSTOOD INITIALLY,
BUT THIS IS STILL A CRITICAL
FACTOR TO BEAR IN MIND
DURING THE DESIGN PROCESS”**

**"IF YOU CANNOT, EVENTUALLY,
EXPRESS YOUR GOAL CONCISELY
IN TERMS OF YOUR READERS
AND THEIR NEEDS, YOU DON'T
HAVE A TARGET TO AIM FOR AND
HAVE NO WAY TO GAUGE YOUR
SUCCESS"**

**“OUR GOAL IS TO PROVIDE A
VIEW OF THE LONDON
SUBWAY SYSTEM THAT
ALLOWS RIDERS TO EASILY
DETERMINE ROUTES
BETWEEN STATIONS”**

“UNDERSTANDING THE GOALS OF THE VISUALIZATION WILL ALLOW YOU TO EFFECTIVELY SELECT WHICH FACETS OF THE DATA TO INCLUDE AND WHICH ARE NOT USEFUL OR, WORSE, ARE DISTRACTING”

“[...] PARADIGMS SUCH AS OBJECT ORIENTATION [INSPIRE] PRACTICAL PHILOSOPHIES AND PROVIDES HERMENEUTIC MODELS FOR ORGANIZING AND UNDERSTANDING THE WORLD, BOTH DIRECTLY (THROUGH PROGRAMED SYSTEMS) AND INDIRECTLY (THROUGH THE WORLDVIEWS OF COMPUTER ENGINEERS)”

Aarseth, Espen J

DATA AND REALITY: A TIMELESS PERSPECTIVE ON PERCEIVING AND MANAGING INFORMATION IN OUR IMPRECISE WORLD

Kent, William

**“AFTER A WHILE IT DAWNED
ON ME THAT THESE ARE ALL
JUST MAPS, BEING POOR
ARTIFICIAL APPROXIMATIONS
OF SOME REAL UNDERLYING
TERRAIN”**

William Kent

**THE MAP IS NOT
THE TERRITORY**

**“WHAT IS THE TERRITORY
REALLY LIKE? HOW CAN I
DESCRIBE IT TO YOU? ANY
DESCRIPTION I GIVE YOU
IS JUST ANOTHER MAP”**

William Kent


```
class Person {
    String name;
    String age;

    User(String name, String age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {return name;}
    public String getAge() {return age;}
}
```

```
// This is not a person
class Person {
    String name;
    String age;

    User(String name, String age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {return name;}
    public String getAge() {return age;}
}
```

THANK YOU

REFERENCES

- ▶ Aarseth, Espen J. *Cybertext: Perspectives on Ergodic Literature*. Johns Hopkins University Press, 1997.
- ▶ Beck, Kent. *Test-Driven Development: by Example*. Addison-Wesley, 2006.
- ▶ Berger, Peter L., and Thomas Luckmann. *The Social Construction of Reality: a Treatise in the Sociology of Knowledge*. Penguin, 1991.
- ▶ Borges, Jorge Luis, and Andrew Hurley. *Collected Fictions*. Penguin Books, 1999.

REFERENCES

- ▶ Carlevaro, Abel. *Serie Didactica: Para Guitarra*. Barry, 1966.
- ▶ Eagleton, Terry. *Literary Theory: an Introduction*. Blackwell Publishing, 2015.
- ▶ Eco, Umberto, and Anthony Oldcorn. *From the Tree to the Labyrinth: Historical Studies on the Sign and Interpretation*. Harvard University Press, 2014.
- ▶ Eco, Umberto. *Lector in Fabula: La Cooperazione Interpretativa Nei Testi Narrativi*. Bompiani, 2016.

REFERENCES

- ▶ Eco, Umberto. *Six Walks in the Fictional Woods*. Harvard Univ. Press, 2004.
- ▶ Genette, Gérard. *Paratexts: Thresholds of Interpretation*. Cambridge Univ. Press, 2001.
- ▶ Gärdenfors, Peter. *Geometry of Meaning: Semantics Based on Conceptual Spaces*. The MIT Press, 2017.
- ▶ Hermans, Felienne, and Marlies Aldewereld. "Programming Is Writing Is Programming." *Proceedings of the International Conference on the Art, Science, and Engineering of Programming - Programming '17*, 2017, doi:10.1145/3079368.3079413.

REFERENCES

- ▶ Kent, William, and Steve Hoberman. *Data and Reality: a Timeless Perspective on Perceiving and Managing Information in Our Imprecise World*. Technics Publications, 2012.
- ▶ Lewis, James, and Martin Fowler. "Microservices." *Martinfowler.com*, 25 Mar. 2014, martinfowler.com/articles/microservices.html.
- ▶ Moore. "What a Programmer Does." *Datamation*, Apr. 1967, pp. 177-178., archive.computerhistory.org/resources/text/Knuth_Don_X4100/PDF_index/k-9-pdf/k-9-u2769-1-Baker-What-Programmer-Does.pdf.
- ▶ Naur, Peter. "Programming as Theory Building." *Microprocessing and Microprogramming*, vol. 15, no. 5, 1985, pp. 253-261., doi: 10.1016/0165-6074(85)90032-8.

REFERENCES

- ▶ "Random Numbers." *The Art of Computer Programming*, by Donald Ervin Knuth, vol. 2, Addison-Wesley, 2011.
- ▶ Steele, Julie, and Noah P. N. Iliinsky. *Beautiful Visualization*. O'Reilly, 2010.
- ▶ Videla, Alvaro. "Metaphors We Compute By." *Communications of the ACM*, vol. 60, no. 10, 2017, pp. 42-45., doi:10.1145/3106625.