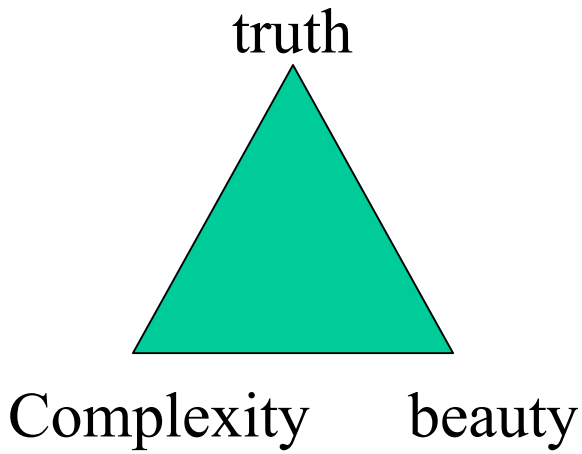
PROPYLON
Transforming Business Integration

Truth, Beauty & Complexity



XMLOpen 2004

Sean McGrath

Propylon

PROPYLON
Transforming Business Integration

Keats on beauty/truth/XML

"Beauty is truth and truth is beauty. That is all ye know on earth…so choose our XML application development tools wisely…"

John Keats

[1795–1821]

Urn

Extended from a base datatype known as "Ode to a Grecian Universal Resource Name"

http://www.bartleby.com/101/625.html

PROPYLON
Transforming Business Integration

Eugene Wigner on simplicity/complexity/XML

Eugene Wigner
Phyiscist/Mathematician

[1902-1995]

*"The **simplicities** of natural laws arise through the **complexities** of the **languages** we use to express them.*

Therefore, domain specific languages are a honking good idea for XML processing."

http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Wigner.html

PROPYLON
Transforming Business Integration

Complex symbols -> "simple" expressions

Leonard Euler
1707-1783

$$e^{i\pi} = -1$$

There is no permenant place in this world for ugly mathematics.

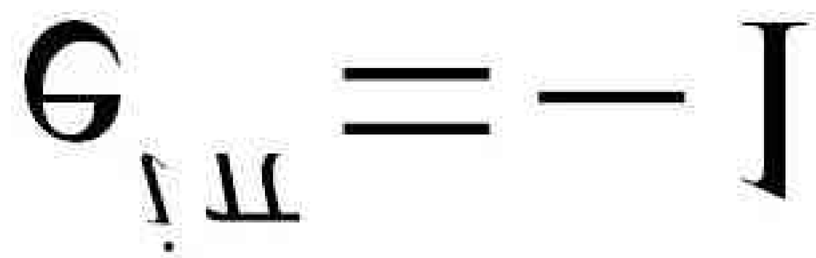
G H Hardy
[1877-1947]

"the most remarkable formula in mathematics!"

*"The **simplicities** of natural laws arise through the complexities of the **languages** we use to express them."*

Richard Feynman
[1918-1988]

So beautiful, it even looks good backwards and upside down



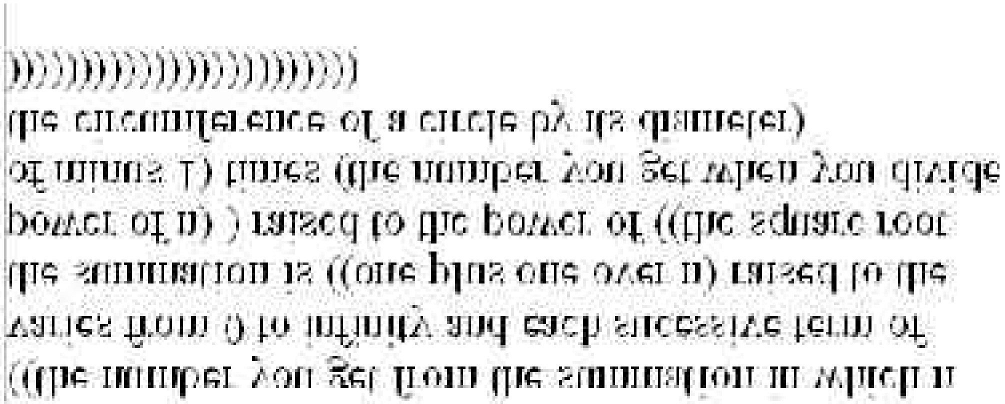
PROPYLON
Transforming Business Integration

A life without complex (= higher order) symbols

Eulers Identity#2:

((the number you get from the summation in which n varies from 0 to infinity and each sucessive term of the summation is ((one plus one over n) raised to the power of n)) raised to the power of ((the square root of minus 1) times (the number you get when you divide the circumference of a circle by its diameter))))))))))))))))))))))))

So beautiful, it works backwards and upside down – not!



PROPYLON
Transforming Business Integration

More beauty through higher order symbols

$$\neg(c \wedge p) \rightarrow \neg c \vee \neg p$$

Augustus De Morgan [1806-1871]

Ramanujan, 1887-1920

$$\sum_{k=0}^{\infty} \frac{(-1)^k(4k+1)[(2k-1)!!]^4}{[(2k)!!]^4} = \sum_{k=0}^{\infty} \frac{(-1)^k(4k+1)[\Gamma(k+\frac{1}{4})]^4}{x^{k/2}[\Gamma(k+1)]^4} = \frac{2}{x}$$

PROPYLON
Transforming Business Integration

De Morgan's Law

Original :

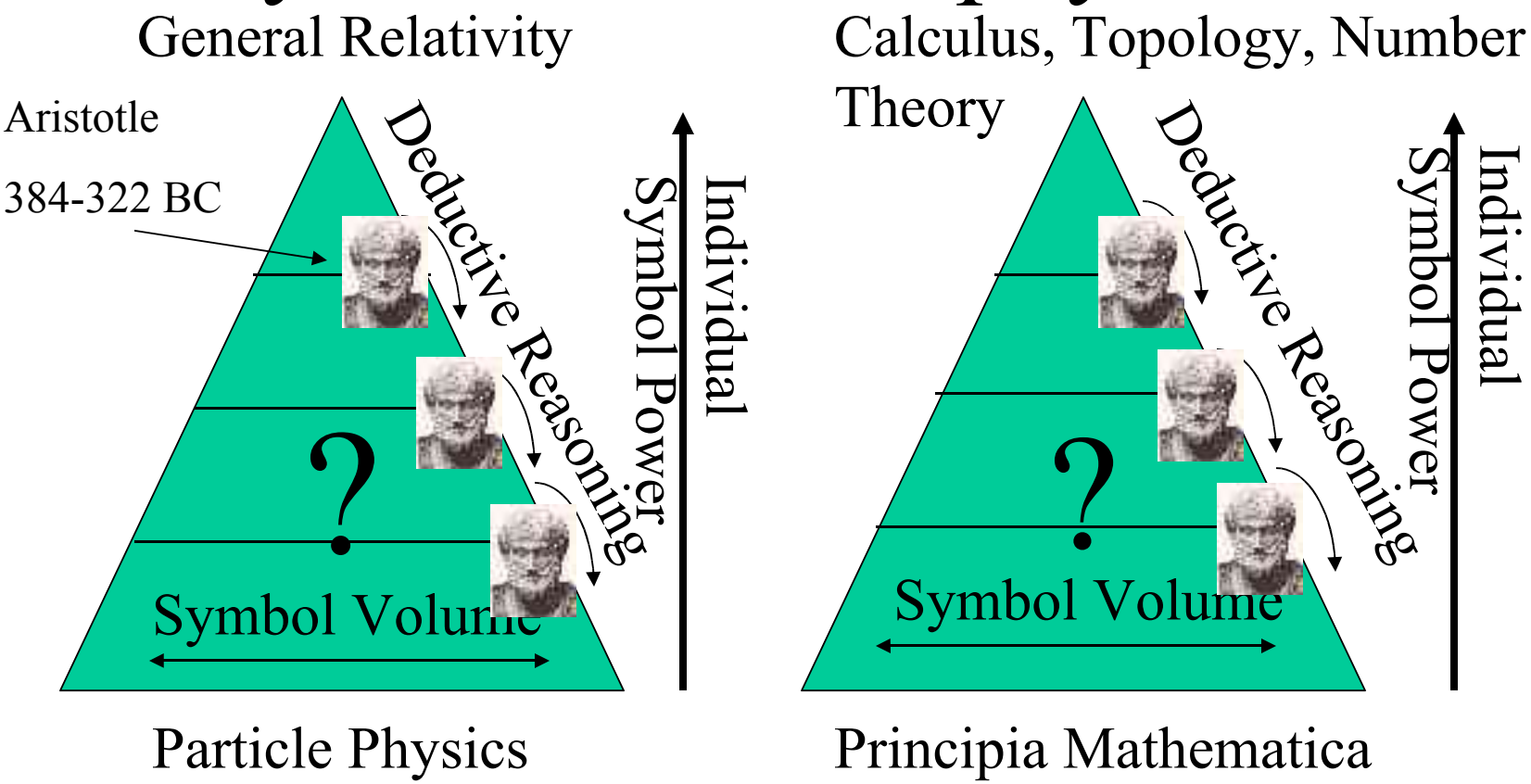
"Great fleas have little fleas upon their backs to bite 'em, And little fleas have lesser fleas, and so ad infinitum." (http://www.worldofquotes.com/topic/Fleas/1/)

Adapted :

"Complex symbols have more complex symbols *up* the layers to bite 'em, and complex symbols have less complex symbols, *down* the layers ad infinitum. (or at least until you hit 1's and 0's."

PROPYLON
Transforming Business Integration

Hierarchies of complex symbols/facts in physics/math

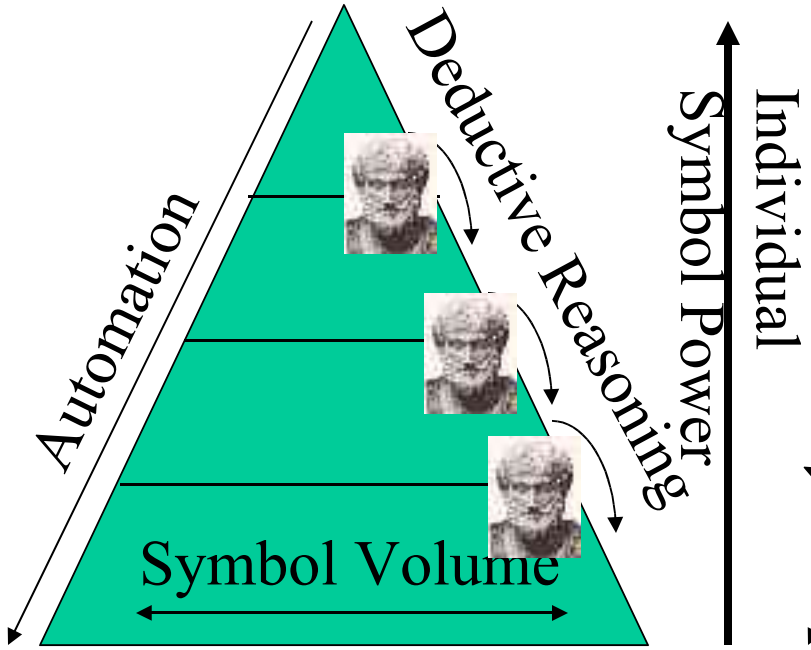


A hierarchy of layers, complex symbols building on less complex symbols…how elegant!

PROPYLON
Transforming Business Integration

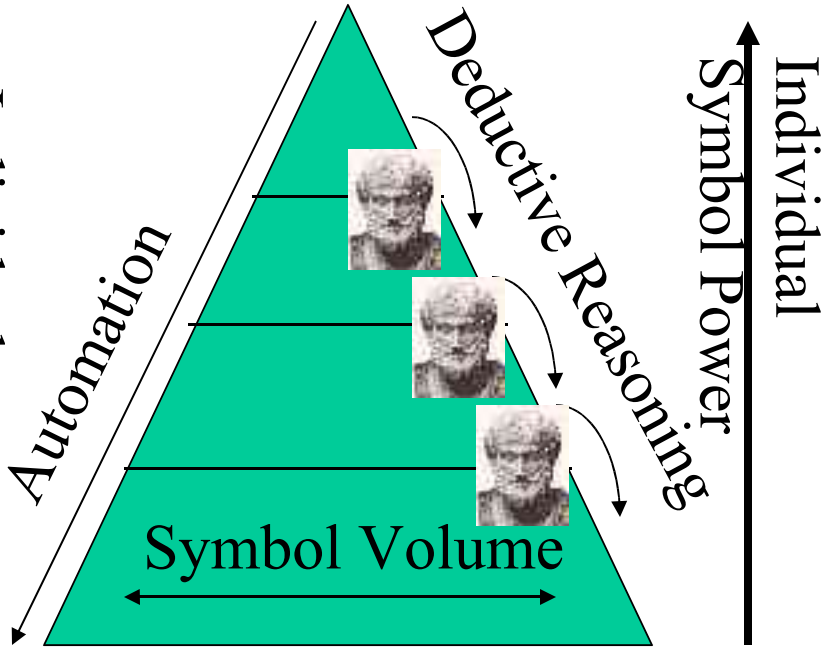
Hierarchies of symbols/facts in XML#1

Document Schema



Document Validation Machine

XSLT

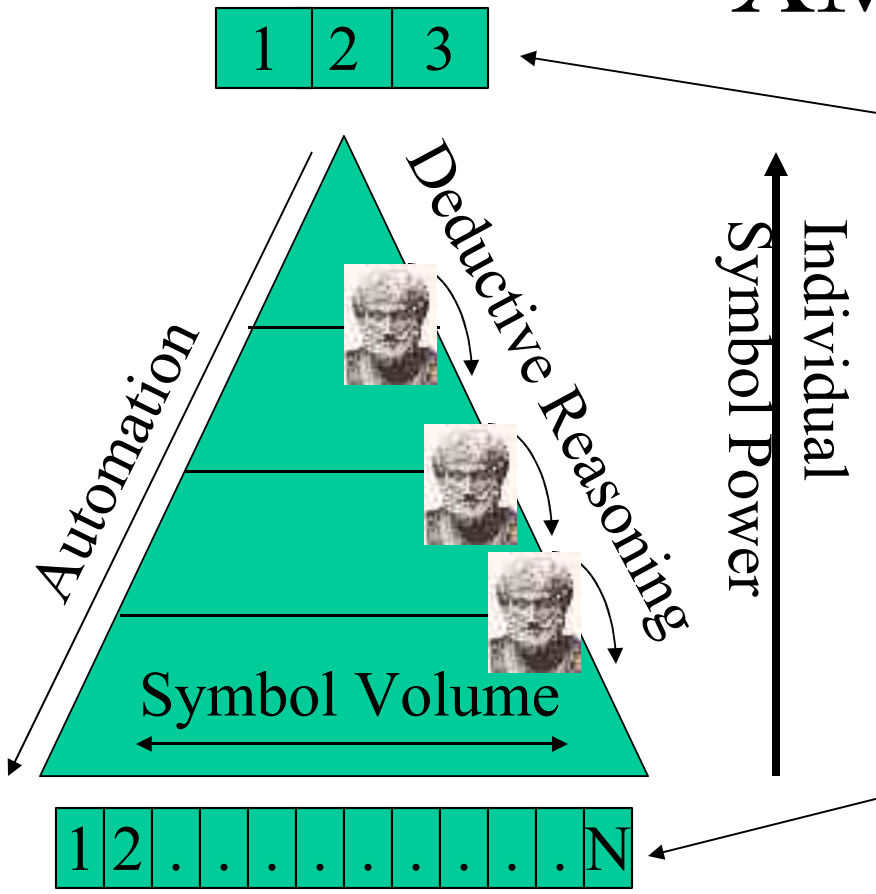


Document Formatting Machine

A hierarchy of layers, complex symbols building on less complex symbols…how elegant!

PROPYLON
Transforming Business Integration

Hierarchies of symbols/facts in XML#2



Individual Symbol Power

Automation

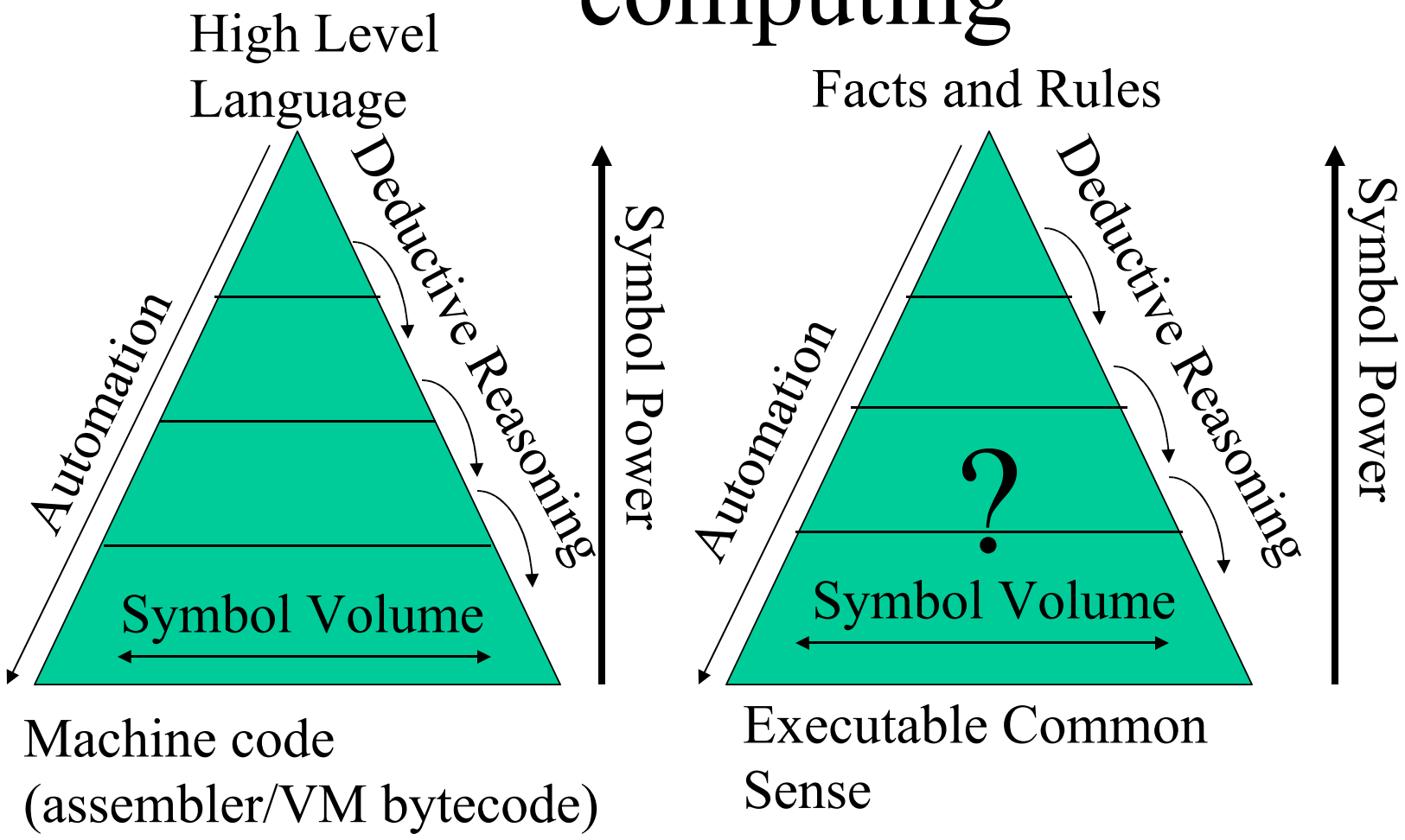
XML Document Transformation Expressed in business symbols.

XML Document Transformation Expressed in XML symbols (trees. nodes, text).

A hierarchy of layers, higher ones building on lower ones…how elegant!

PROPYLON
Transforming Business Integration

Hierarchies of symbols/facts in computing



A hierarchy of layers, higher ones building on lower ones…how elegant!

PROPYLON
Transforming Business Integration

The relationship between hierarchy and complexity

Herbert A. Simon

1916-2001

"Hierarchy … is one of the central structural schemes that the architect of complexity uses."

http://www-scf.usc.edu/~mehta/readings/sciences.html

"Complex" as in "carbohydrates" not "complex" as in "Post Schema Validation Infoset"

PROPYLON
Transforming Business Integration

Chaitin on Elegance

Gregory Chaitin
Mathematician
IBM,

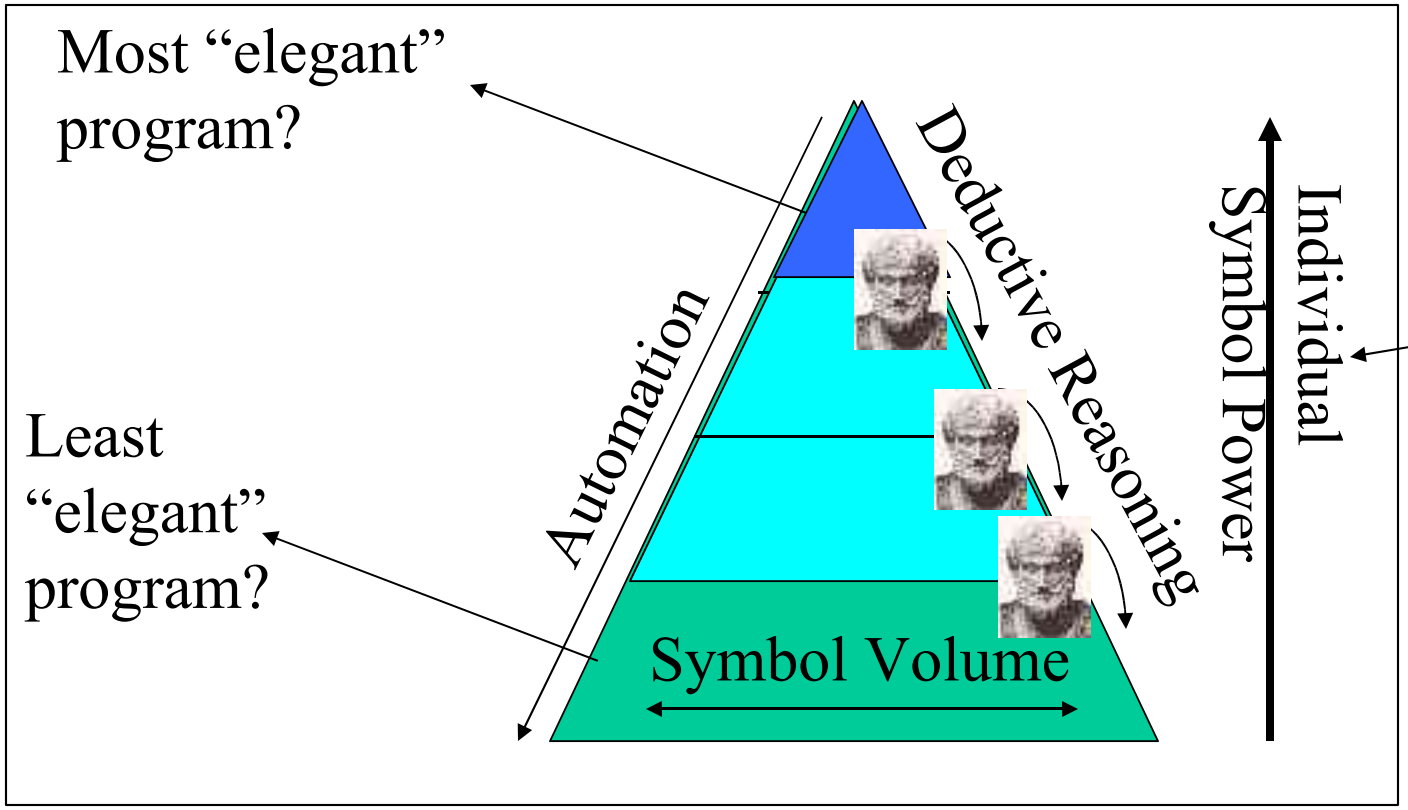
http://www.umcs.maine.edu/~chaitin/

"A computer program written in a given language is *elegant* if no smaller program written in the same language has the same output."

It is impossible to prove that a given program (above a certain very low level of complexity) is elegant."

PROPYLON
Transforming Business Integration

Beauty = Hierarchical symbol power? Elegance?



PROPYLON
Transforming Business Integration

The three edged sword of powerful symbols

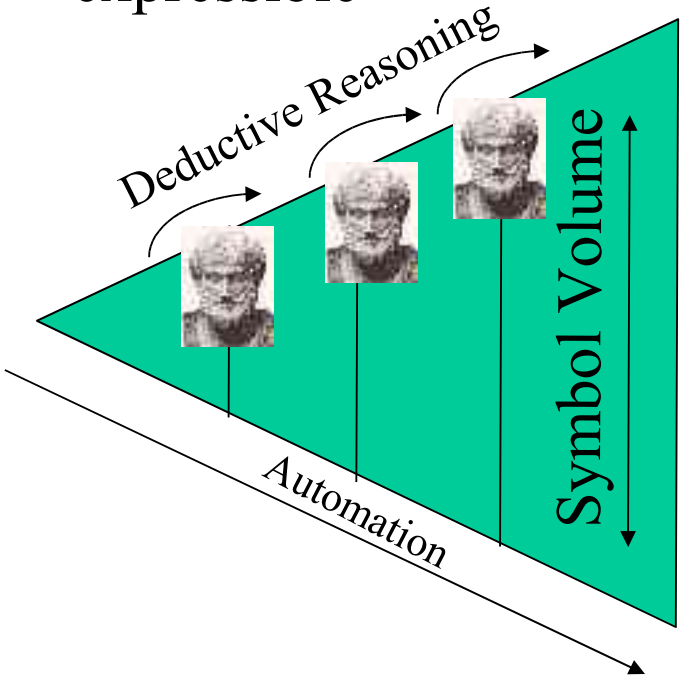
Increasing ease of expressing the expressible

Relax NG

XQuery

SQL

Schematron
etc.



JVM
Bytecode

CLR

C

Turing
Machines

Increasing expressability

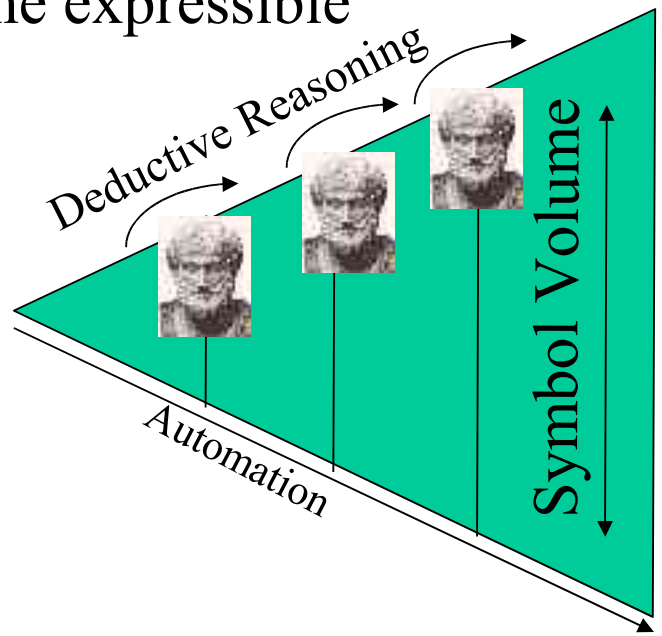
PROPYLON
Transforming Business Integration

The three edged sword of powerful symbols

Relax NG
XSLT
SQL
Sendmail
Configuration
File etc.

Some complex things ar easy. But many simple things are **impossible**.

Increasing ease of expressing the expressible



Expressability

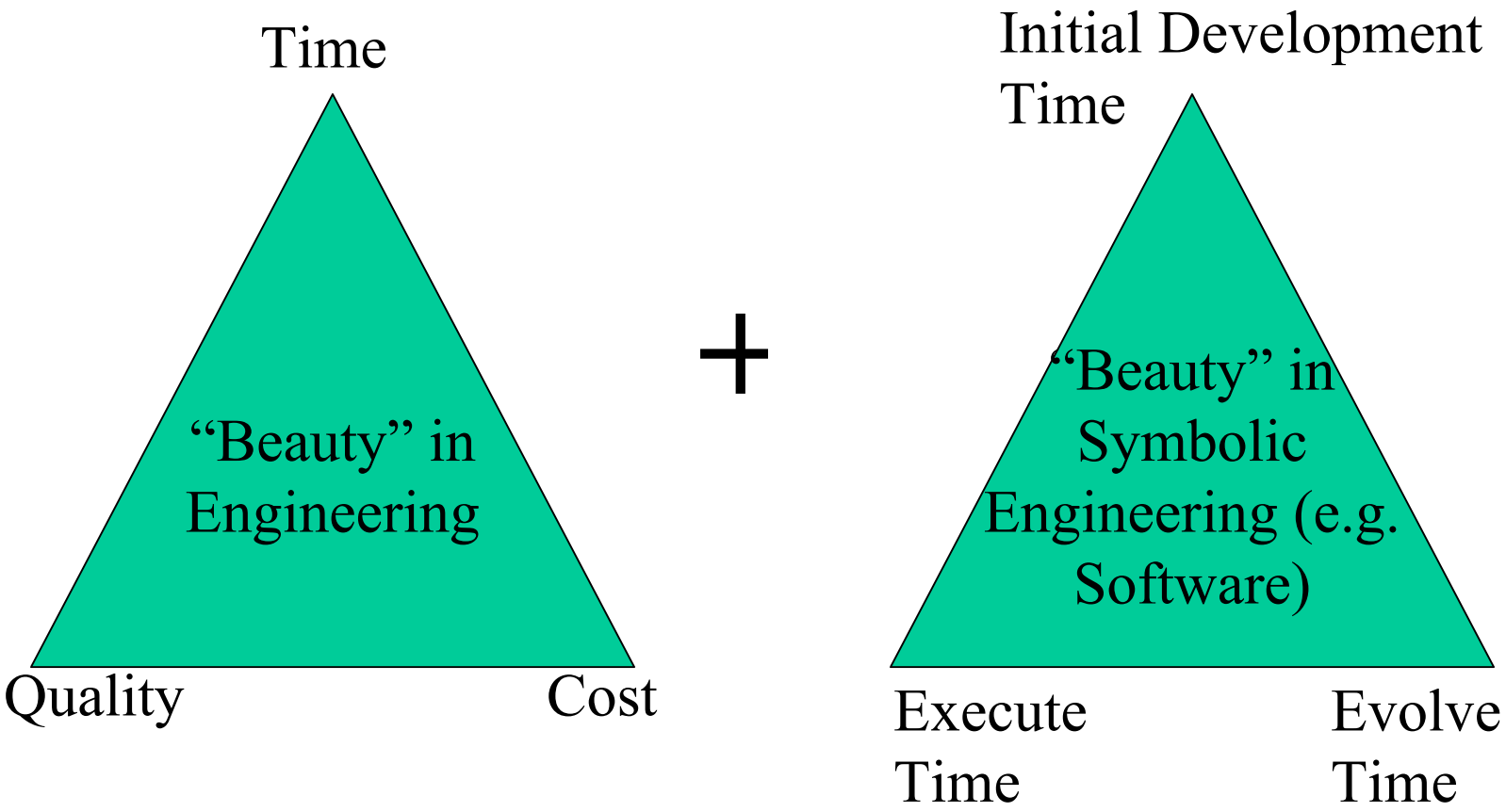
JVM Bytecode
CLR
C
Turing Machines
Yada, yada.

Everything is complex (from a programming Point of view). But **nothing** (that can be computed) is impossible.

Sean McGrath, http://seanmcgrath.blogspot.com

PROPYLON
Transforming Business Integration

The three edged swords of powerful symbols



PROPYLON
Transforming Business Integration

The three edged sword of symbols

Low developer focus. Takes longer with simple symbols (but theoretically, you can always get a result).

Initial Development Time

"Beauty" in Symbolic Engineering (e.g. Software)

High developer focus.

Favours simple symbols.

Execute Time

Evolve Time

Mostly non-existent developer focus. More complex with simple symbols (but theoretically, you can always get a result.)

PROPYLON
Transforming Business Integration

Routes towards the incenter of time, cost, quality

Initial Development Time

Option 1:

Create **new**, higher level symbols.

Do more with less.

Data Type Libraries, RelaxNG, XQuery

Option 2:

More tools for shovelling **existing** symbols around.

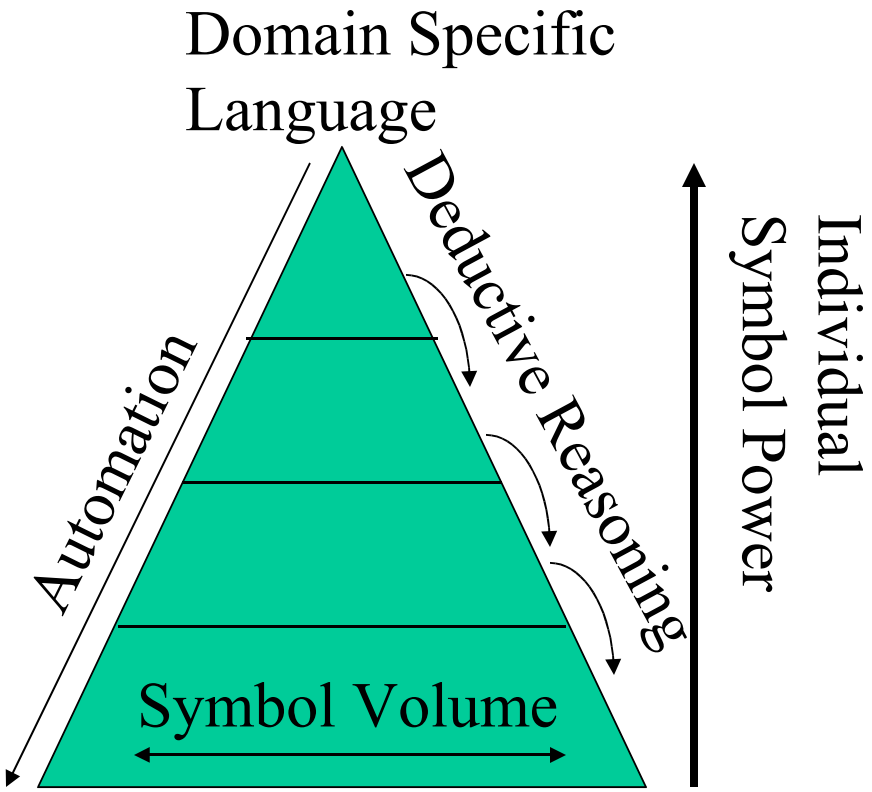
GUIs, Wizards, RAD tools.

Execute Time

Evolve Time

PROPYLON
Transforming Business Integration

Option 1 – higher level symbols



PROPYLON
Transforming Business Integration

Option 1 - Example

Michael Sperberg McQueen

xsd:date

High level symbol

Low level symbols

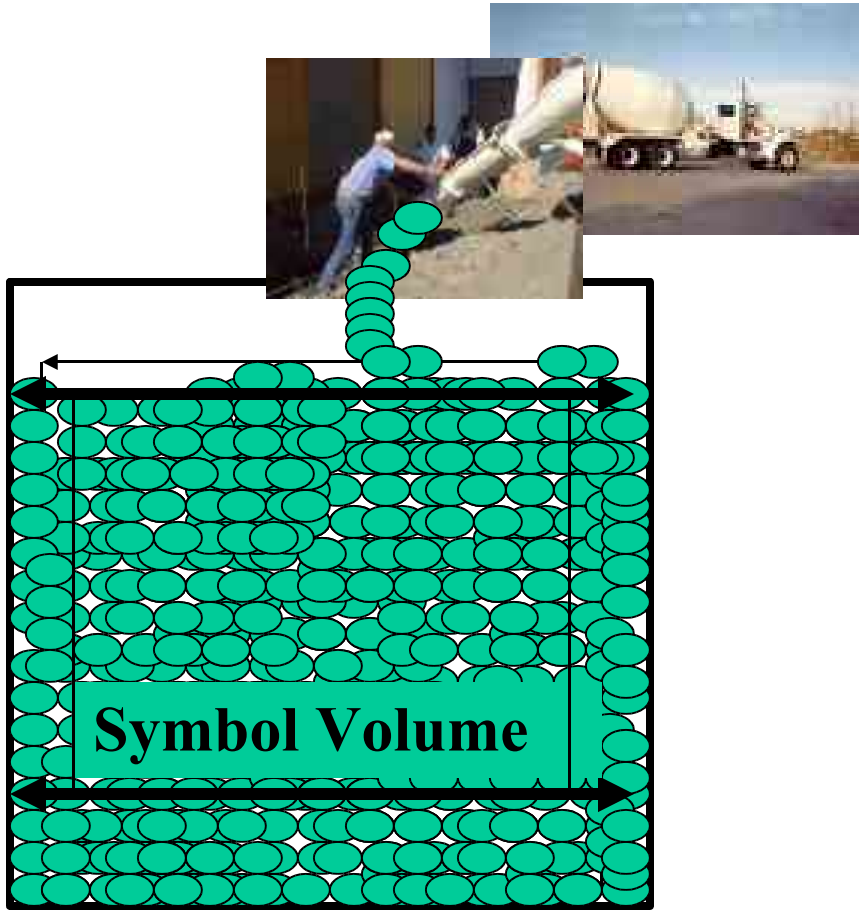
```
date      ::= leapyear '-02-29‘
            | year '-' longmonth '-31‘
            | year '-' nonfeb    '-' ('29' | '30')
            | year '-' month     '-' ('0' [1-9] |
             '1' digit | '2' [0-8])

longmonth ::= '0' [13578] | '1' [02]
nonfeb    ::= '0' [13456789] | '1' [012]
leapyear  ::= [02468] E1 | [13579] O1
E1 ::= [048] Q2 | [26] E2 | [13579] O2
O1 ::= [26] Q2 | [048] E2 | [13579] O2
Q2 ::= '0' QT | [2468] E3 | [13579] O3
E2 ::= '0' ET | [2468] E3 | [13579] O3
O2 ::= '0' OT | [2468] E3 | [13579] O3
```

```
QT ::= '0' QC | [48] QY | [26] EY | [13579] OY
ET ::= '0' EC | [48] QY | [26] EY | [13579] OY
OT ::= '0' OC | [26] QY | [48] EY | [13579] OY
Q3 ::= '0' QD | [48] QY | [26] EY | [13579] OY
E3 ::= '0' ED | [48] QY | [26] EY | [13579] OY
O3 ::= '0' OD | [26] QY | [48] EY | [13579] OY
QC ::= /* leap year */ | '0' QC | [48] QY | [26] EY | [13579] OY
EC ::= '0' QC | [48] QY | [26] EY | [13579] OY
OC ::= '0' EC | [48] QY | [26] EY | [13579] OY
QD ::= '0' QC | [48] QY | [26] EY | [13579] OY
ED ::= '0' EC | [48] QY | [26] EY | [13579] OY
OD ::= '0' OC | [26] QY | [48] EY | [13579] OY
QY ::= /* leap year */ | '0' QD | [48] QY | [26] EY | [13579] OY
EY ::= '0' ED | [48] QY | [26] EY | [13579] OY
OY ::= '0' OD | [26] QY | [48] EY | [13579] OY
```

PROPYLON
Transforming Business Integration

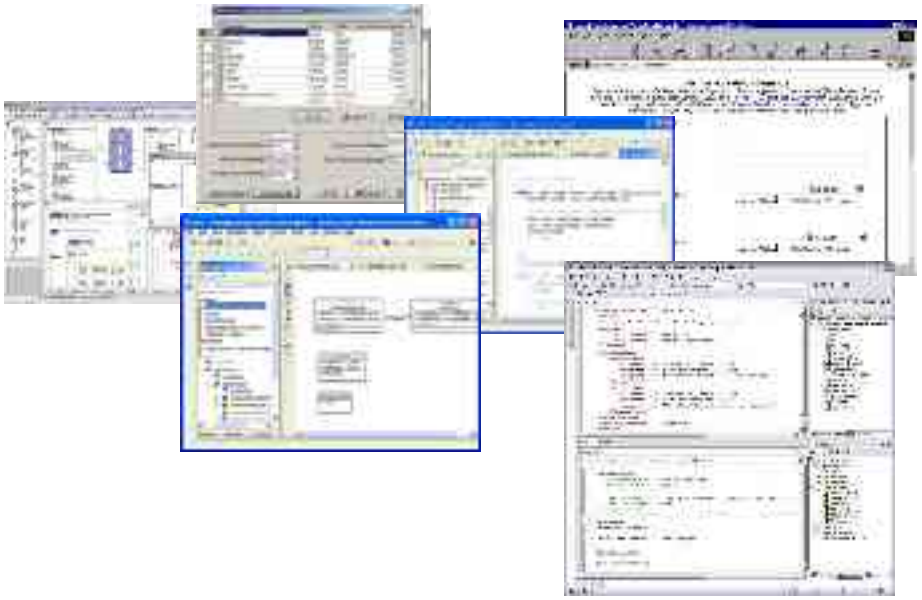
Option 2 – symbol shovelling tools



PROPYLON
Transforming Business Integration

Option 2 - Examples

Pretty much anything that generates low level symbols from a GUI ***without*** also generating high level symbols…



Decrease transparency, user understanding, evolvability & maintaintability of systems.

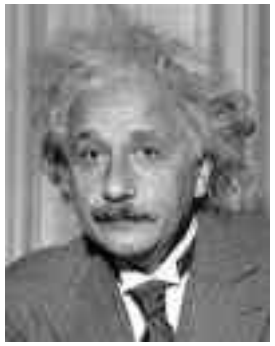
Increase "skin deep" beauty/elegance in the form of Graphical User Interfaces.

PROPYLON
Transforming Business Integration

Counter point. Small != good

Grecian Urn

Theory of General Relativity



Albert Einstein

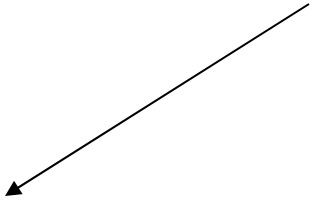
[1879-1955]

XML namespaces sanity preservation guidelines.

PROPYLON
Transforming Business Integration

Counterpoint

Perl



```
t(a,b,c){int
d=0,e=a&~b&~c,f=1;if(a)
for(f=0;d=(e-=d)&-e;f+=t(a-
d,(b+d)*2,( c+d)/2));return
f;}main(q){scanf("%d",&q);
printf("%d\n",t(~(~0<<q),
0,0));}
```

PROPYLON
Transforming Business Integration

Counter points: TCP/IP, ISO seven layered model

Vint Cerf

Tim Berners Lee

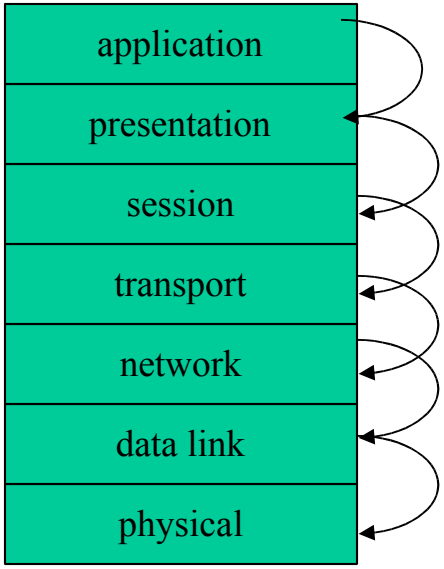
Use presentation Layer Logic Professor Frink!

Ah! Layer 6. Wa hay!

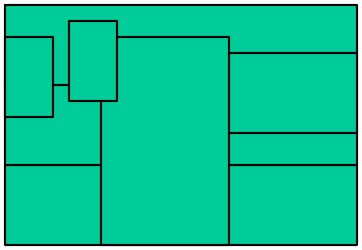
Real world

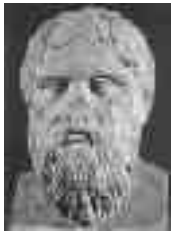
Ideal World

ISO Seven Layered Model



TCP/IP/HTTP "Stack"





Plato

PROPYLON
Transforming Business Integration

Counter point: SGML/XML

Hermes, Messenger to the Gods

M.C. Escher

1898-1972

Words
Elements
Token
Lexical

XInclude
Parameter
Entities/External
Entities
Type annotations etc.

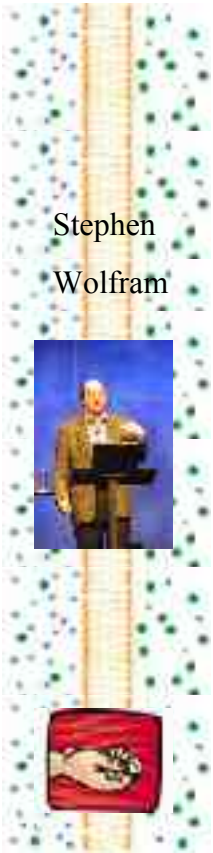
PROPYLON
Transforming Business Integration

Counter points : Superstrings and Omega Numbers

Brian Green

Ed Witten

Michael Green



Stephen Wolfram

Alan Turing

Kurt Godel

Gregory Chaitin

Bertrand Russell

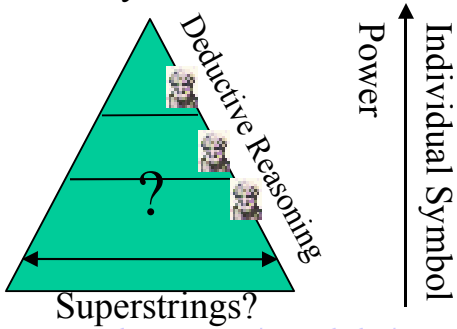
General Relativity

Quantum Mechanics

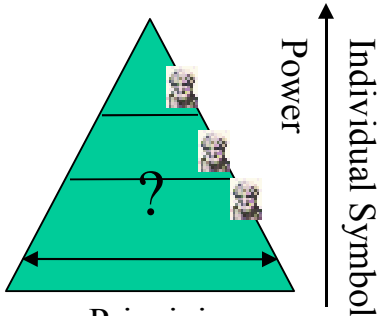
GUTs

&

TOEs



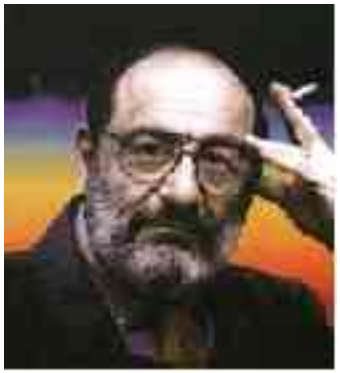
Calculus, Algorithms



http://www.pbs.org/wgbh/nova/elegant/

PROPYLON
Transforming Business Integration

Counter points: symbol overlap



Umberto Eco

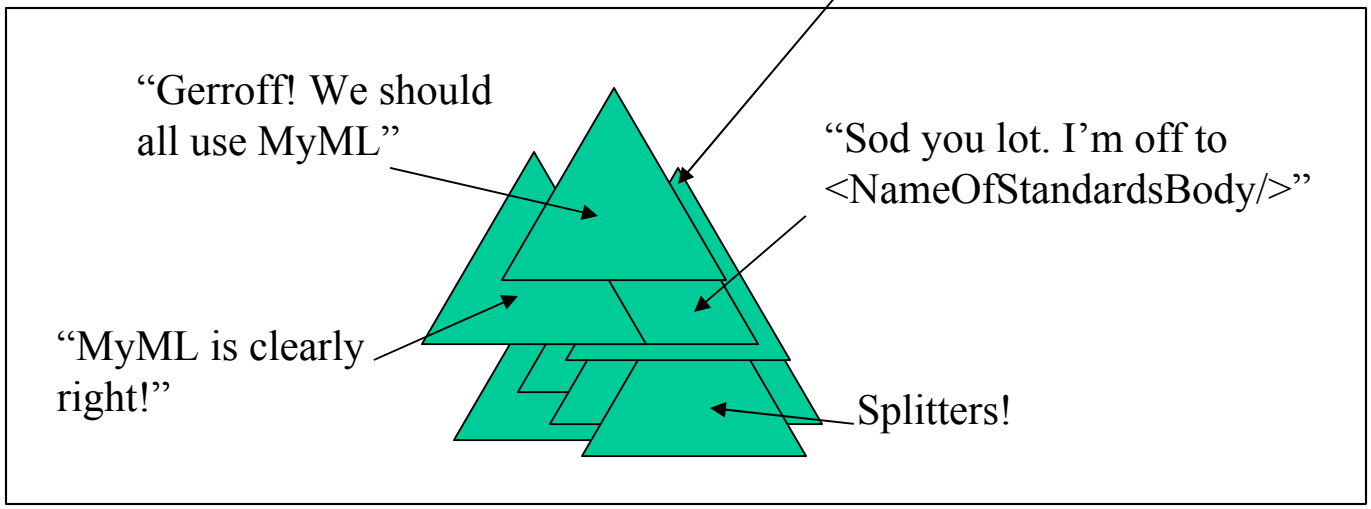
"The Search for the Perfect Language" ISBN: 0631205101

Tower of Bable

Carl Shapiro

Hal Varian

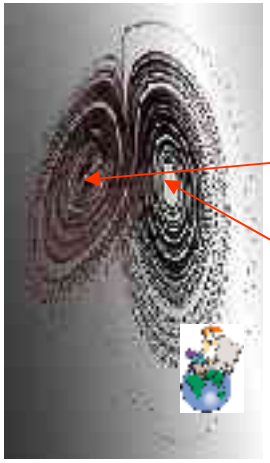
"Information Rules" ISBN: 087584863X



The "ML" wars

PROPYLON
Transforming Business Integration

In conclusion



Lets use high level but limited symbols (ebXML, HL-7) etc.

Lets use low level but comprehensive symbols (RDF, Topic Maps etc.)

[Edward Lorenz]

- High level symbols are good.
- High level symbols *reducing* deductively to lower level symbols are fundamentally good.
- Mere symbol shovellers have their place but…
  - *True* progress lies in powerful symbols and powerful symbol reducers, *not* in more powerful shovels – case in point: Relax NG Compact Syntax
  - Beware powerful shovels that encourage symbol complexity through lack of high level symbols.

"*it is better to avoid gratuitous complexity than to rely on tools to manage it*."
http://www.manageability.org/blog/stuff/rod-johnson-j2ee-without-ejb

Thank you.