


# Beyond the Curse of Symmetry



*Breaking Down the  
Dualism: Starfish, Milk  
Drops, Beauty, and  
Antimatter Guns*

James O. Coplien,  
Object Architect  
DAFCA, Inc.

## Computing in the Dualistic World



- ⌘ Computing: A solution to many problems
- ⌘ We solve problems
- ⌘ *Design* is the process of solving problems
- ⌘ We usually build things to solve problems:  
to create structure

## Computing as Problem

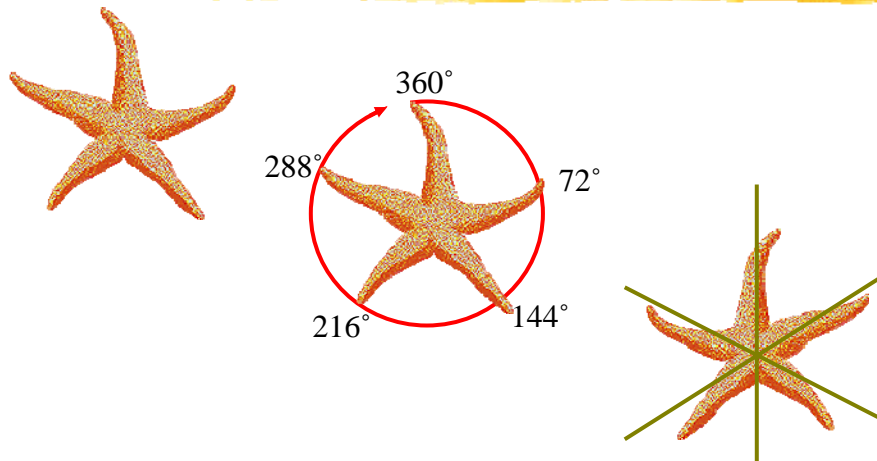
- ⌘ Alexander: “Get moral”
- ⌘ Raskin: “Get Humane”
- ⌘ Dijkstra: “Get formal”
- ⌘ Everyone’s got a gripe
- ⌘ Us and Them
- ⌘ Dualism—*about* dualism
- ⌘ But so much for philosophy—we’re just programmers!

## What do we build?

### *Structure*

- ⌘ Few of us think procedurally: we think structurally
- ⌘ *Structured* programming; data *structures*; *object-oriented*
- ⌘ We adopt architecture,
- ⌘ We throw in a bit of our academic, mathematical heritage and we get...
- ⌘ Symmetry, elegance, regularity, formalism
- ⌘ We have science-ized structure

## The versatile starfish



## Dualism: Classic Symmetry

- ⌘ Symmetry: balancing invariants and change
- ⌘ Scientific absolutism:
  - ☒ Is or Is Not object-oriented
  - ☒ Is or Is Not
- ⌘ Problem space versus solution space
- ⌘ Dualism is a manifestation of the symmetry disease

## CS manifestations of symmetry—and their failings

- ⌘ The search for tracibility and “seamlessness”: symmetry—the Holy Grail
- ⌘ Procedural programming
  - ☒ Invariant algorithms around variation in arguments
- ⌘ OOP
  - ☒ Strict (LSP) invariance in class invariants with well-structured pre-and-post conditions
  - ☒ (This is mathspeak for “type systems”)
- ⌘ Yet procedural programming doesn’t seem to fit the real world at human scale,
- ⌘ And object-oriented programming needs lots of “cheating” at human scale.

## Programs as Living Things

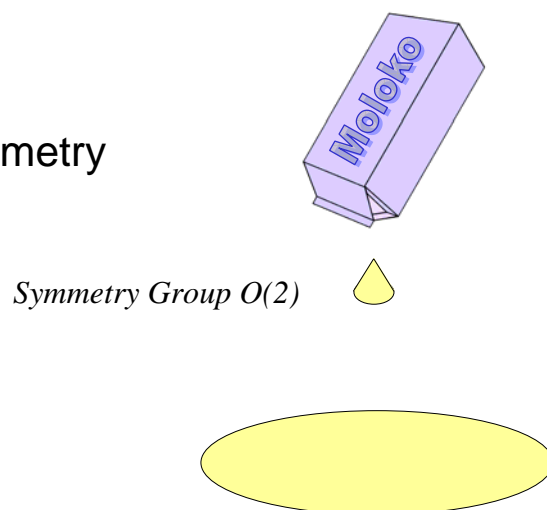
- ⌘ Breaking down the dualism
- ⌘ They *are* living things: now, let’s adopt processes that reflect that
  - ☒ They heal themselves locally as a cut, bruise, or ecosystem does
  - ☒ They are autopoietic: self-organizing; not “methodology-organized” or “OO”
  - ☒ They are full of replication and snub their nose at “reuse”
  - ☒ Their parts adapt locally, not according to “requirements”

## But it goes even deeper

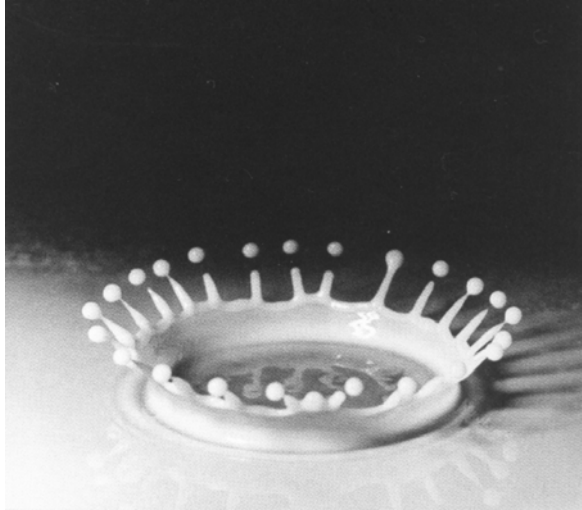
- ⌘ Programmers, as human beings, are also part of nature and do not follow
- ⌘ Even in art, we try to force the dualistic model
- ⌘ The golden mean
- ⌘ Other symmetric things in modern culture

## Symmetry-preserving Curie principle

- ⌘ Cause/effect relationships preserve symmetry



## Where did the symmetry go?



*Symmetry group  $D_{24}$*

## Symmetry Breaking and Beauty

### ⌘ Distributes global symmetry locally

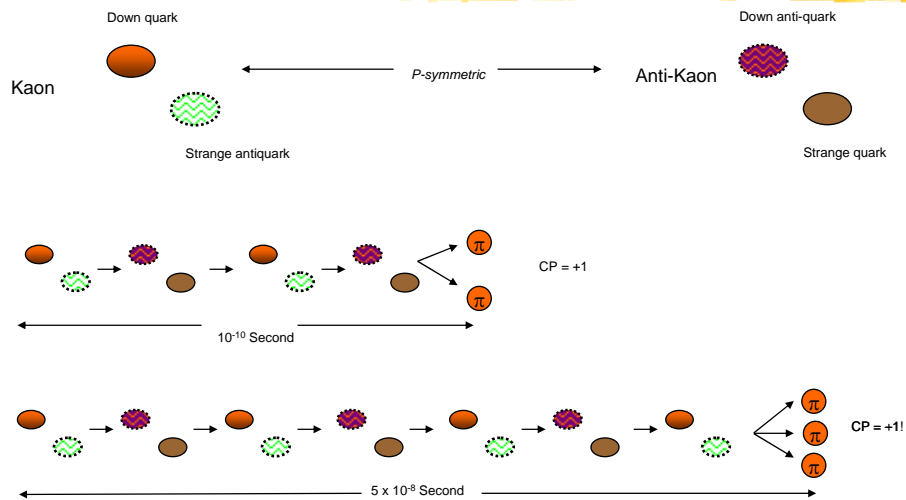
Living things, though often symmetrical, rarely have perfect symmetry. Indeed perfect symmetry is often a mark of death in things, rather than life. I believe the lack of clarity in the subject has arisen because of a failure to distinguish overall symmetry from local symmetries.

and:

In general, a large symmetry of the simplified neoclassicist type rarely contributes to the life of a thing, because in any complex whole in the world, there are nearly always complex, asymmetrical forces at work—matters of location, and context, and function—which require that symmetry be broken.

— Alexander, *Nature of Order*, “Phenomenon of Life”, p. 44

# K-meson Decay



## Implications

- ⌘ Once in every 300-500 times, we get an excess of matter over antimatter
- ⌘ Therefore, nature *prefers* matter
- ⌘ At the Big Bang, the universe was symmetric
- ⌘ It wasn't just chance that we ended with a matter-full universe

## Broken symmetry in C++

Complex



⌘ Breaks symmetry  
(structural invariant  
of `Complex`; e.g.,  
`sizeof()` )

⌘ How do we solve it?

Real

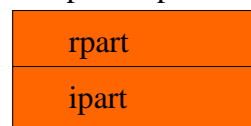


## With a *pattern*: BRIDGE

Complex



ComplexRep



Real



RealRep





## Further Progress in Software

- ⌘ AOP
- ⌘ Multi-paradigm design
- ⌘ Generative Programming
- ⌘ Patterns

## What is a pattern?

- ⌘ This paradox, that symmetry can get lost between cause and effect, is called *symmetry-breaking*. In recent years scientists and mathematicians have begun to realise that it plays a major role in the formation of patterns... From the smallest scales to the largest, many of nature's patterns are a result of broken symmetry; and our aim is to open your eyes to their mathematical unity. (*Stewart and Golubitsky, p. xviii*)
- ⌘ Nature, too, creates beautiful structures which are governed by repeated application of structure-preserving transformations. In this connection, I think it is useful to remark that what I call structure-preserving transformations are very closely related to what has become known as "symmetry breaking" in physics ...This claim is a minimum-change argument: *the system acts so as to preserve as much of its structure as possible.* (*Nature of Order, 63—4*)

## Reality is not dualistic

### ⌘ Purity as a form of dualism:

- ☒ The Moral Majority's purity
- ☒ Smalltalk object-orientation as purity

### ⌘ Reality is messy...

- ☒ ... or beautiful

### ⌘ C++ is...

- ☒ ...expressive of broken symmetry

## Broken Symmetry in C++: Violating Invariants

Symmetric Feature	Broken Symmetry
<code>struct</code>	<code>union</code>
Public Inheritance	<code>private</code> Inheritance
Function call	Overriding default argument values
<code>templates</code>	<code>template</code> specialization
Static typing	<code>dynamic_cast</code>
<code>X::operator+(X)</code>	<code>friend operator+(X,X)</code>

## Why is this important?

- ⌘ We use computers for highly repetitive tasks
- ⌘ Sometimes it's important to be fast
  - ⏏ The competitive edge, all other things being equal
- ⌘ It's Not Nice to Fool Mother Nature
  - ⏏ Good design is both intentional and reflects object-ive reality—which is rarely object-oriented

