Design patterns and neo-platonism ...

by Daniel J. Sullivan

Pub Date: 11/30/11

Rev Date: 02/12/17

It is, in my view, a stupid and ridiculous waste of time to take on the "Gang of 4" head on. So I propose a more modest and simple proof:

1) Design patterns are assumed to be correct.

2) Correct is often, and for practical purposes, equated to TRUE.

3) The application of the 'design pattern' will happen in a future state -- after the Go4 book was written (after it was read).

4) This 'truth' is being applied to a future state.

5) The application of truth to a future state is purpose driven – Teleological.

6) Design patterns are seen as prototypes of what is true.

7) The 'prototype' (even in a Wittgenstein context) is a platonic concept (not an inductive/scientific one).

8) Plato's model of the universe was both Teleological and Archetype/Prototype driven.

9) This application, design patterns, is well past the Platonic Epoch of philosophical thought.

Q.E.D. --> Design Patterns are neo-platonist in nature.

So what?

Is neo-platonism bad in the context of software?

Well, yes, neo-platonism is quite bad. Neo-platonism (and design patterns) make assumptions about 'future state'. As with data normalization, we 'design' the database based upon the 'ideal' records. In truth, there is no ideal record and there is no ideal business object or process. Living things change because the universe changes. If software were dead, then yes, design patterns would be correct. In truth, we should think of our designs as theories. Theories undergo change. Theories are mutable. Theories allow (and expect) free thought.

Design theories allow for convergence on perfection - design patterns stop well short of this.

Yes, the patterns are useful. Just as Sister Una, slamming her ruler on the desk, while repeating the multiplication table, was, kind of useful. But, design patterns are training wheels and possibly (if they were given proper respect by their authors) design theories which can live on.