CSC323 2010S *Software Design*

Class 24: Design Patterns (2): Creational Patterns

Held: Thursday, April 29, 2010

Summary: We consider the first category of design patterns, the so-called Creational Patterns.

Related Pages:

* EBoard.

Notes:

* For Tuesday, read about Structural Patterns in Wikipedia
  `http://en.wikipedia.org/wiki/Structural_pattern`.
* EC for Tonight's Concert-like thing.

Overview:

* Creational Patterns 1: Factories and Related Things
* Creational Patterns 2: Builders and Singletons
* Project Fun!

Three Related Patterns

* The problem: Client needs to create objects at run-time. We'd like to make it easy to switch which classes it creates belong to.
* Why would we want to do this?
* Abstract Factory: Pass in a "Factory" that Client asks to build objects.
* Factory Method: Client calls its own create(). Subclasses of Client() override create().
* Prototype: Pass in a prototypical object that Client asks to clone itself.
* In some object-oriented languages: Classes as first-class objects.
* Questions:
    * Why use Prototype rather than classes as first-class objects?
    * Why is a prototype different than a factory?
    * What heuristics would you use for choosing which of these to use?

Two Other Patterns

* Builder: Permit more complex construction of objects.
* Singleton: Ensure that only one instance of a class is created.
    * Also Registry: Provide a mapping from names to singletons.
* Questions:
    * Why Singleton rather than a static class?

Project Time

- Group switch time!
- How has Ravi's time logger been working?
- Surprise! Group presentation time. (5-10 minutes each)
    - What's in the world model right now? How do the pieces fit together?
    - What's in the GUI right now? How do the pieces fit together?
    - What's in the network right now? How do the pieces fit together?
- We'll also try to critique some code.