The Role of Architectural Styles in Successful Software ~~Product Lines~~ Ecosystems

Richard N. Taylor

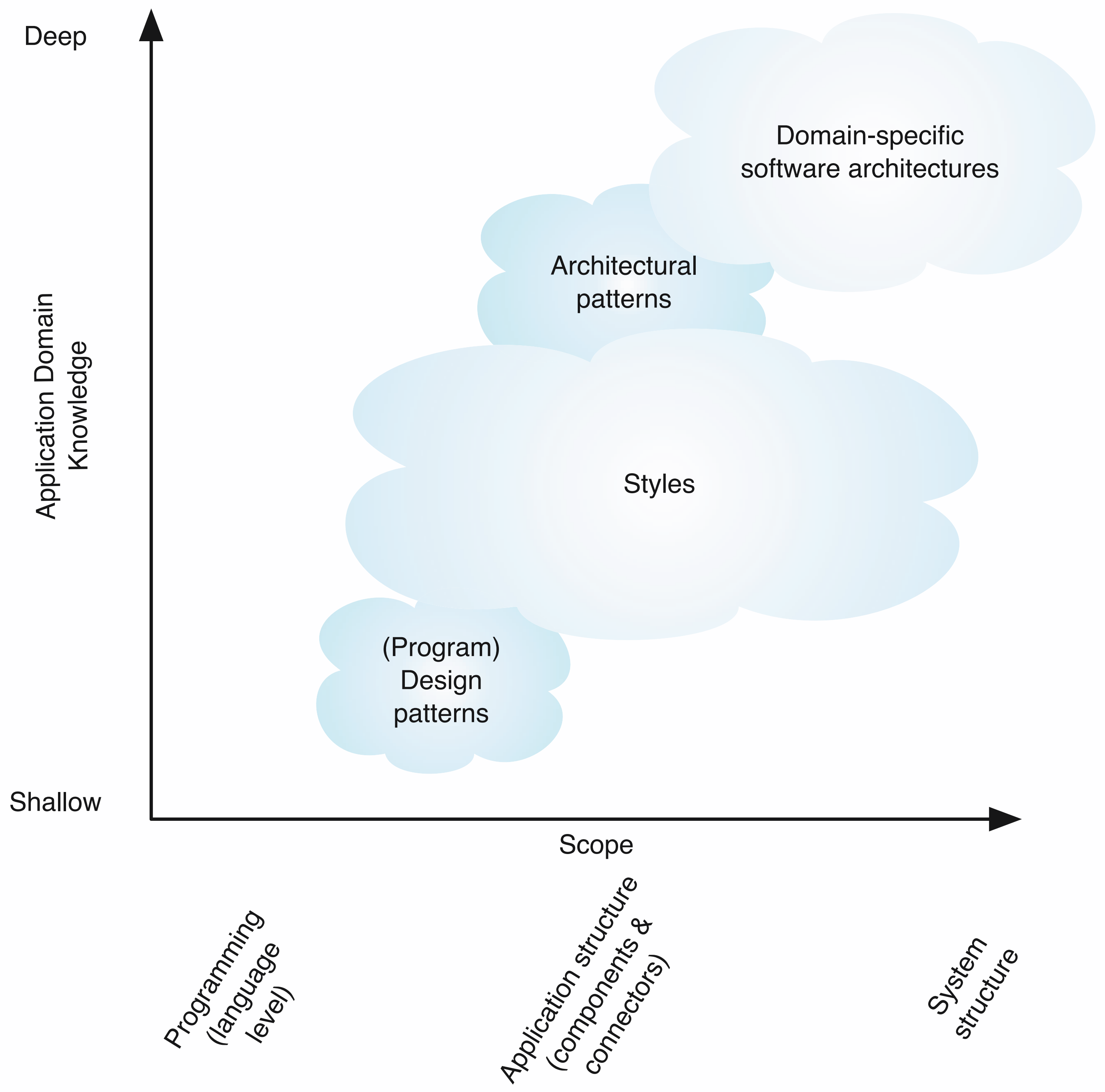
University of California, Irvine

The End at the Beginning

- Take-away #1: A well-chosen, well-designed architectural style is key to a successful product line or ecosystem
- Take-away #2: Multi-agency, decentralized applications offer special challenges and demand new approaches
  - (Especially when you care about $$$ and security)
  - (And we have some ideas…)

Unpacking the Title...

- An architectural style is a named collection of architectural design decisions that
  (1) are applicable in a given development context,
  (2) constrain architectural design decisions that are specific to a particular system within that context, and
  (3) elicit beneficial qualities in each resulting system.


Deep
Application Domain Knowledge
Shallow
Domain-specific software architectures
Architectural patterns
Styles
(Program) Design patterns
Scope
Programming (language level)
Application structure (components & connectors)
System structure

Success? What's That?

- Profit?
- Decreased time to market? Yes!
- Widespread use?
- Adaptability?

Product Lines & Ecosystems

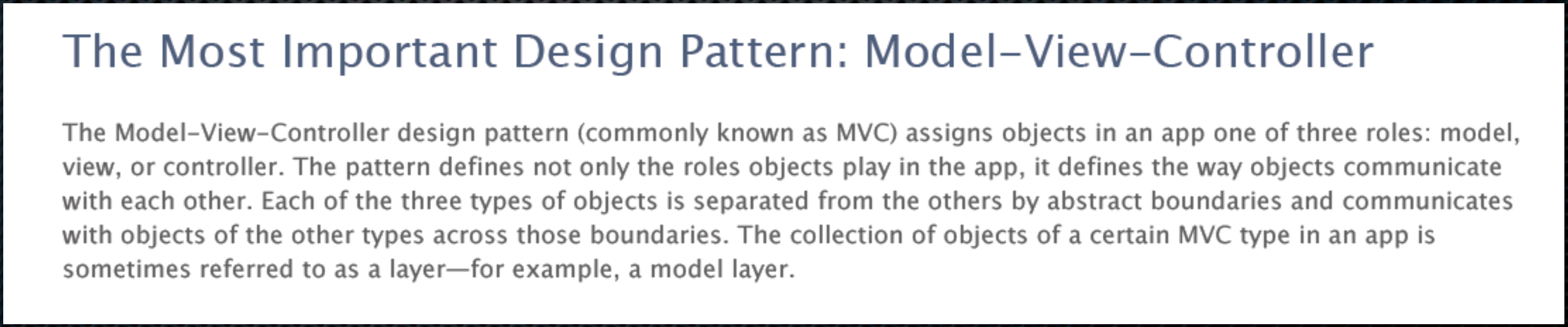
- Product Line: separate products that share significant *technical* commonality in components and structure
  - Examples: Philips TV sets; the iPhone family
- Ecosystem: a complex system composed of multiple organisms, interacting with it and with each other
  - Examples: Amazon, Photoshop, Apple's iOS Apps

Product Line Success Stories

- Philips televisions and medical devices
- Samsung consumer electronic devices
- Automotive applications
- Software product line Hall of Fame: http://splc.net/fame.html

Styles and Ecosystem Success Example #1: Apple iOS Apps

- "MVC is central to a good design for any iOS app or Mac app."



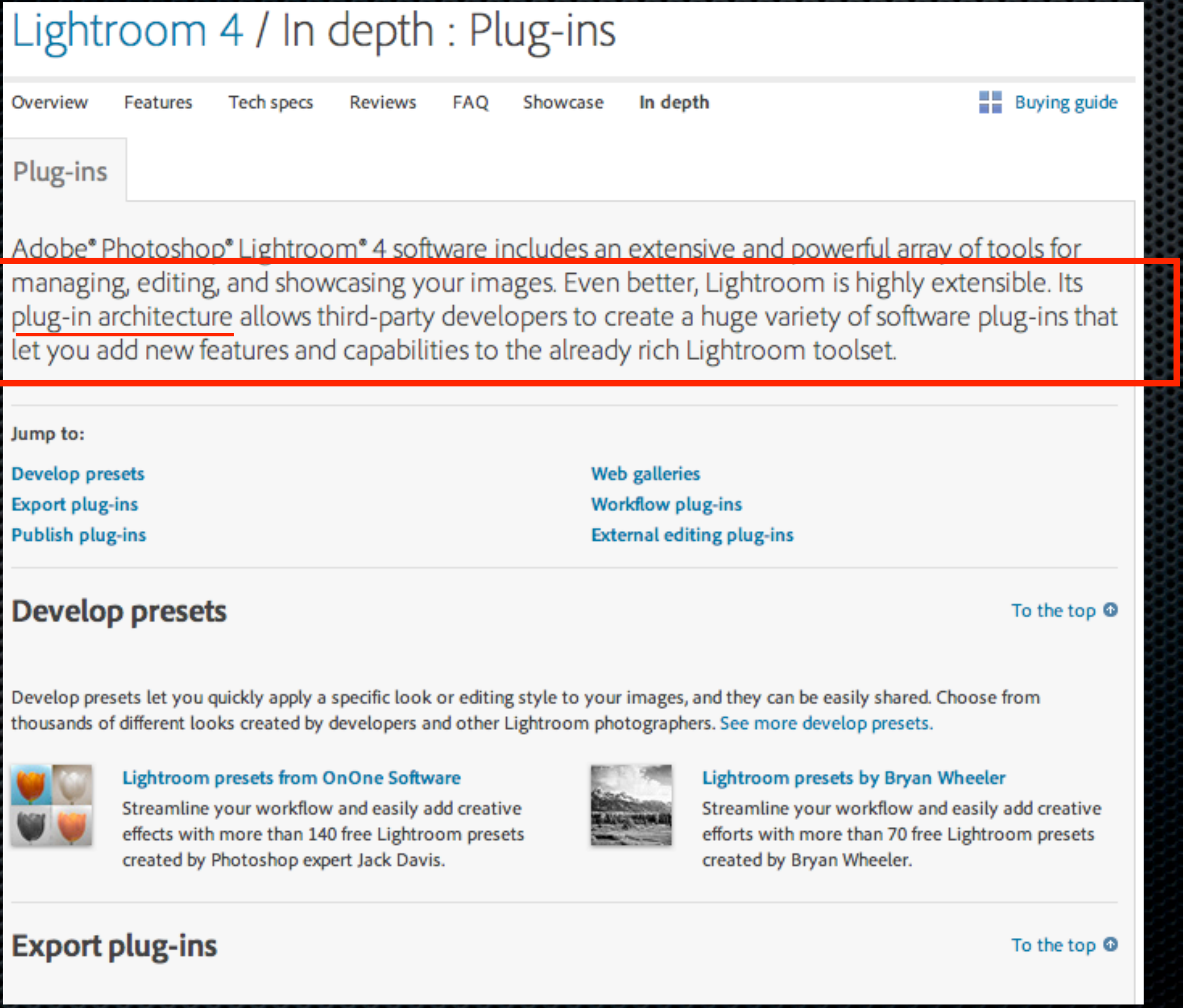
The Most Important Design Pattern: Model–View–Controller

The Model–View–Controller design pattern (commonly known as MVC) assigns objects in an app one of three roles: model, view, or controller. The pattern defines not only the roles objects play in the app, it defines the way objects communicate with each other. Each of the three types of objects is separated from the others by abstract boundaries and communicates with objects of the other types across those boundaries. The collection of objects of a certain MVC type in an app is sometimes referred to as a layer—for example, a model layer.

iOS App Design & Dev

- Architectural Styles (aka Design Patterns)
  - MVC
  - Event Notification
- Frameworks
  - Cocoa and Quartz
  - Foundation, UIkit, Core Graphics
- Guidance and guidelines
  - “iOS Human Interface Guidelines”
- XCode SDK

Example #2: Photoshop



Lightroom 4 / In depth : Plug-ins

Overview Features Tech specs Reviews FAQ Showcase **In depth** Buying guide

Plug-ins

Adobe® Photoshop® Lightroom® 4 software includes an extensive and powerful array of tools for managing, editing, and showcasing your images. Even better, Lightroom is highly extensible. Its plug-in architecture allows third-party developers to create a huge variety of software plug-ins that let you add new features and capabilities to the already rich Lightroom toolset.

Jump to:

Develop presets
Export plug-ins
Publish plug-ins
Web galleries
Workflow plug-ins
External editing plug-ins

Develop presets

To the top

Develop presets let you quickly apply a specific look or editing style to your images, and they can be easily shared. Choose from thousands of different looks created by developers and other Lightroom photographers. See more develop presets.

Lightroom presets from OnOne Software
Streamline your workflow and easily add creative effects with more than 140 free Lightroom presets created by Photoshop expert Jack Davis.

Lightroom presets by Bryan Wheeler
Streamline your workflow and easily add creative efforts with more than 70 free Lightroom presets created by Bryan Wheeler.

Export plug-ins

To the top

Example #3: Web Services

- "The world of web services has been on fast track to supernova ever since the architect astronauts spotted another meme to rocket out of pragmatism and into the universe of enterprises. But, thankfully, all is not lost. A renaissance of HTTP appreciation is building and, under the banner of REST, shows a credible alternative to what the merchants of complexity are trying to ram down everyone's throats; a simple set of principles that every day developers can use to connect applications in a style native to the Web." -- David Heinemeier Hansson, Foreword to RESTful Web Services.

Web Services for the Real World
RESTful
Web Services
O'REILLY
Leonard Richardson & Sam Ruby
Foreword by David Heinemeier Hansson

RESTful Design Principles

- Addressability of information (via URLs)
- Context-free interactions (application state on the client; resource state on the server)
- Links and connectedness (HATEOAS)
- Appropriate use of the uniform interface (i.e. GET, PUT, DELETE, HEAD, POST)

Product Lines v. Ecosystems

- Product Lines
  - often single agency
  - success criteria: reduced dev costs; faster time to market; higher quality (esp. initial quality)
- Ecosystems:
  - multi-agency
  - widely varying success criteria: profit, visibility, reach, “coolness,” mindshare, functionality

Ecosystems

- Multiple examples
  - e-commerce, healthcare, defense, space systems, power grids, highways, factory automation, ...
- Multiple objectives
  - Not all of which are shared
  - Not all of which are compatible
  - Not all of which are benign

Invasive species can be hazardous to public lands and waters! Prevent their introduction and spread by cleaning your vehicles, machinery, and boats.

FICMNEW

STOP AQUATIC HITCHHIKERS!

Styles, Going Forward

- Styles remain a key element in Product Line/ Ecosystem success
  - A well-chosen, well-designed architectural style is key to a successful product line or ecosystem
- Why? Styles carry experience, aid

  Conceptual Integrity d

  design, yield predictable benefits
- But what style for open, decentralized, critical ecosystems, with weeds?

COmputAtional State Transfer (COAST)

- The COAST style:
  - For decentralized applications (the context)
  - Based on mobile computations, communication constraints, POLA (the constraints)
  - Yields dynamic adaptability, pervasive security, ... (some of the beneficial qualities as architectural consequences)
- Status:
  - Full infrastructure in place for evaluative applications
  - Seeking application partners
  - Working an electronic healthcare record scenario