

Supporting Reconfigurable Object Distribution for Customizable Web Applications

Po-Hao Chang and Gul Agha
University of Illinois at Urbana-Champaign



Background

- It is difficult to build a good Web application
 - ▶ need assumptions on execution contexts
- Thin clients? Fat clients?
- Broadband? Dial-up? 3G?

Web Applications do not adapt well !!

Analysis

Rigid composition structure !!

Location: Where to run? Where to create?

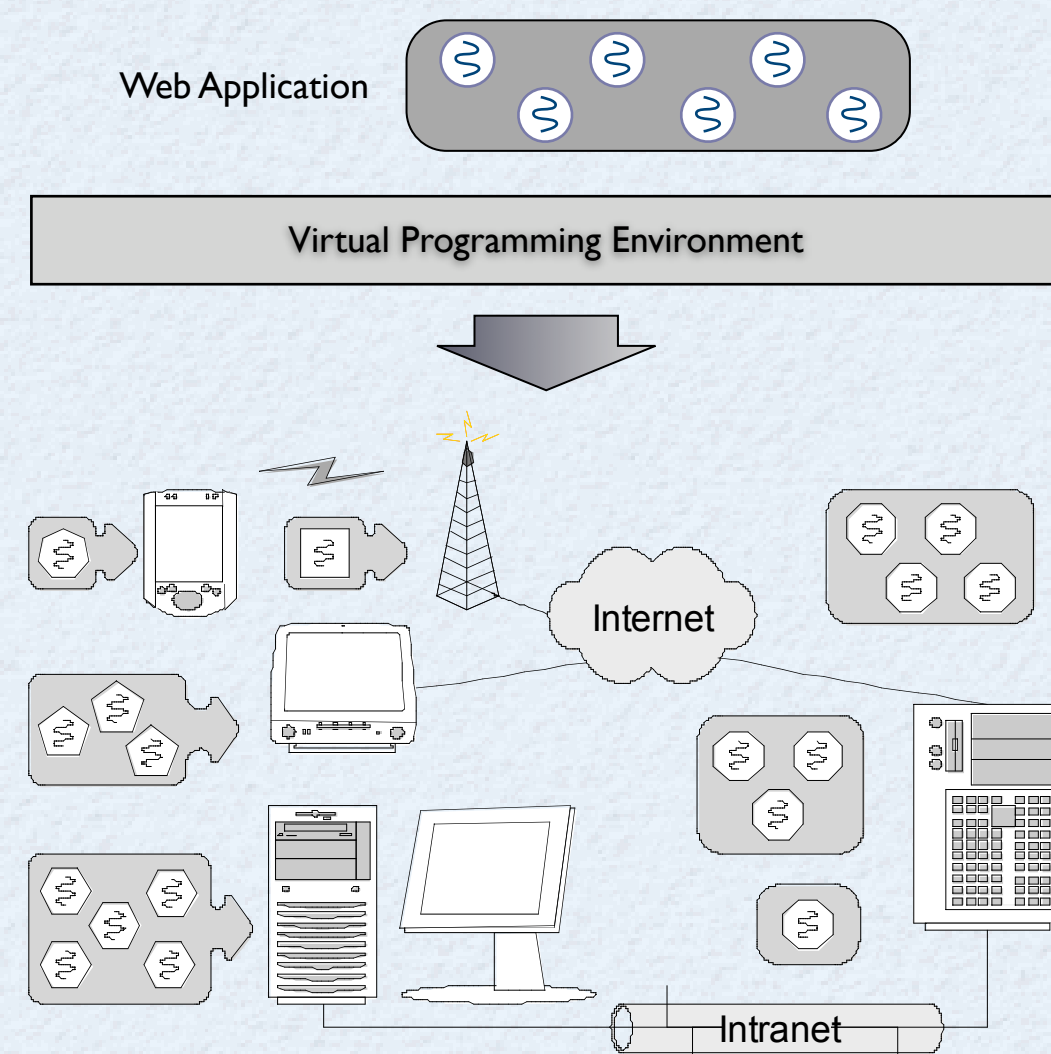
Timing: When to load? How much to load?

Basic Ideas

- A generic Web application = a composition of objects
 - ▶ the base design of a product line
- Adaptable objects
- Reconfigurable object distribution

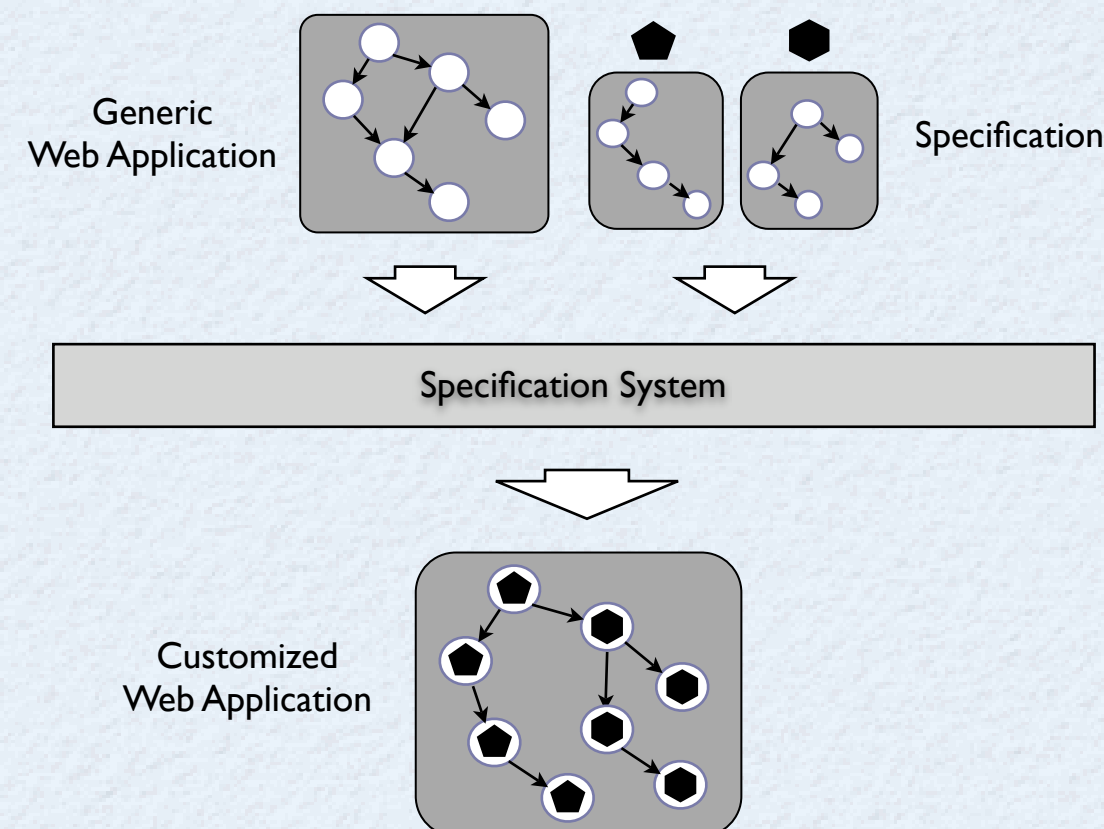
Customizable Web applications !!

Virtualization



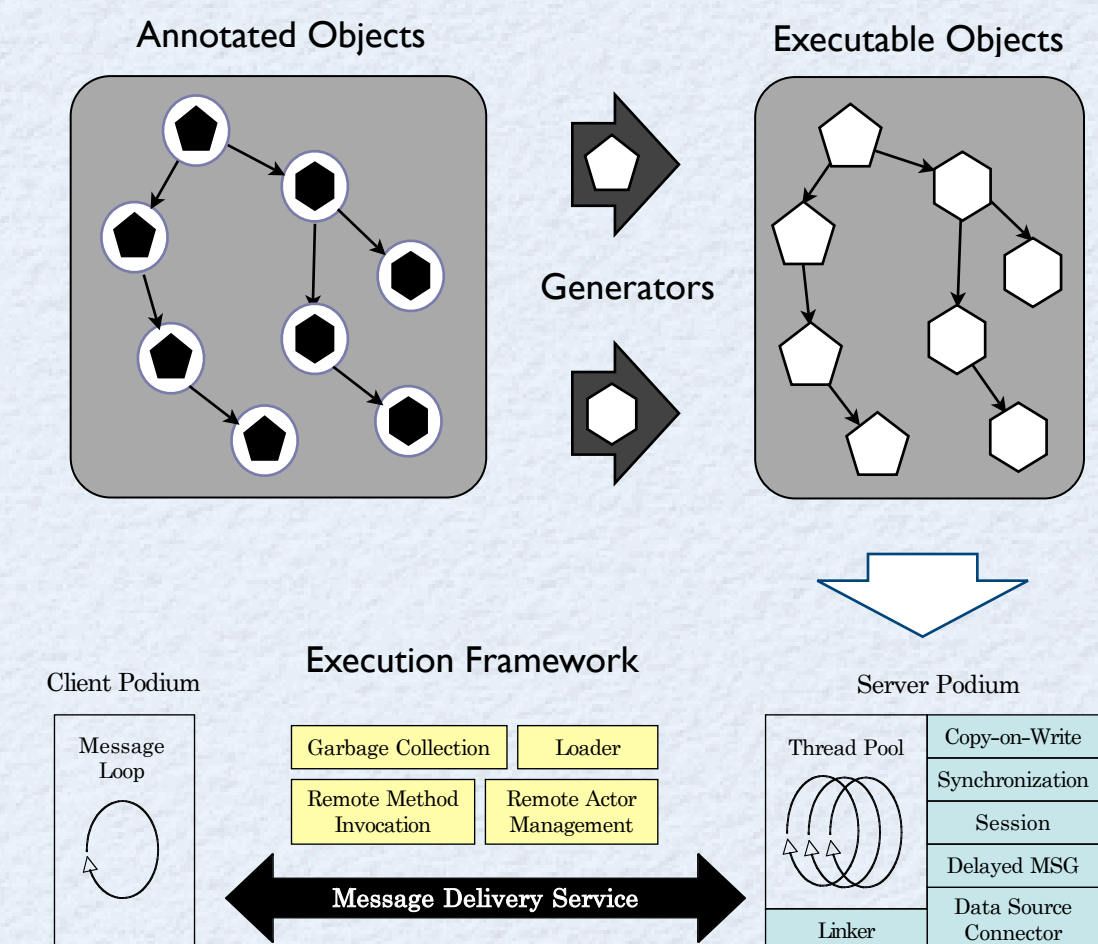
- Virtual programming environment
 - ▶ platform independent
 - ▶ location agnostic
 - Applications are adaptable to a variety of
 - ▶ execution platforms
 - ▶ distribution plans
- Focus on building objects and their interaction !!*

Separation of Concerns



- How to express concerns?
 - ▶ execution platform
 - ▶ loading policy
 - How to annotate objects?
 - ▶ by creation prototype (type, class)
 - ▶ by object genealogy
 - Specification = rules of annotation
 - How to enforce specification?
 - ▶ application transformation
 - ▶ annotation prototypes
 - ▶ prototype splitting
- Object annotation !!*

Generative Components



- Runtime execution environment
 - ▶ full-fledged middleware? *impractical*
 - ▶ retargetable compiler? *non-scalable*
 - Light-weight object execution framework
 - ▶ support object interaction only
 - ▶ host native objects
 - Native (executable) object generators
 - ▶ convert object logic into target languages
- Balanced strategy !!*