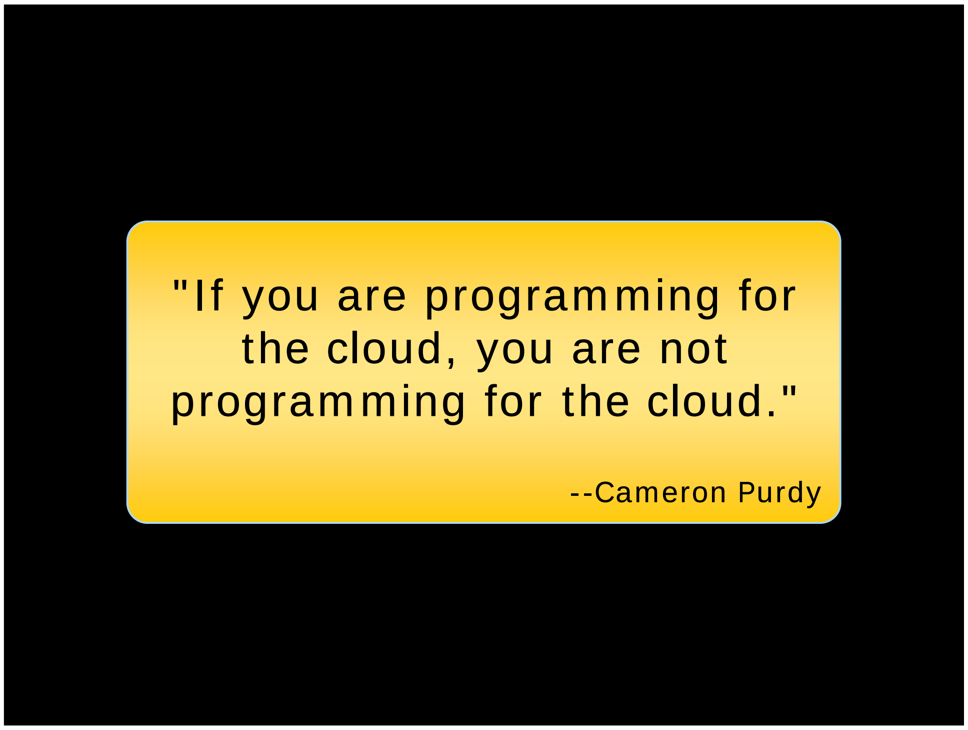
Programming the Cloud – The Internet as a Platform

Gregor Hohpe

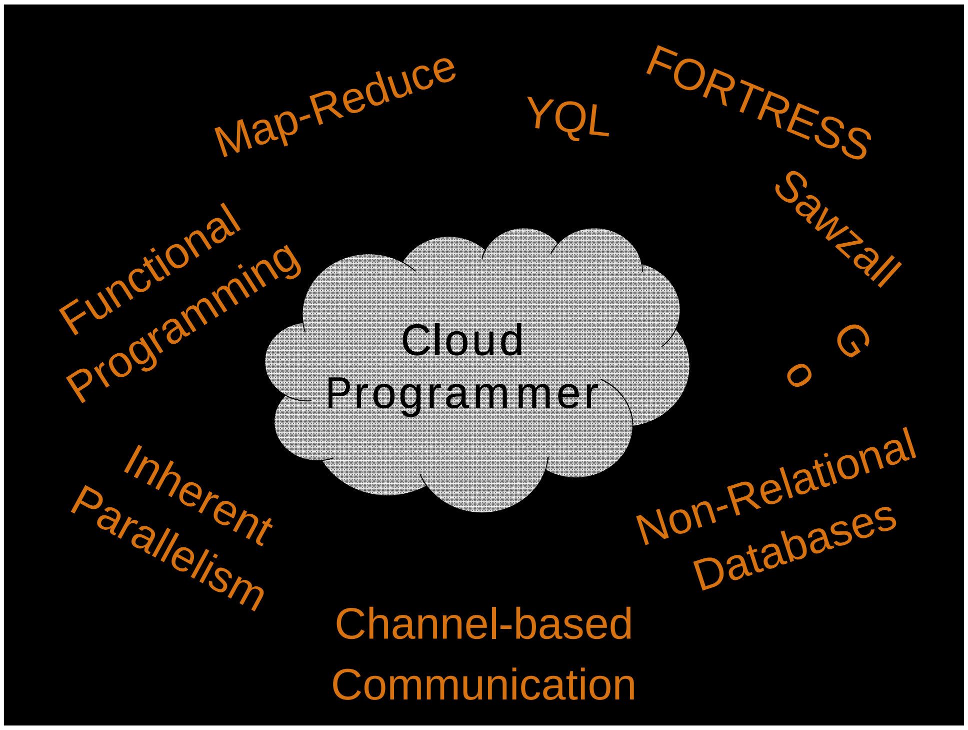
Google

www.EnterpriseIntegrationPatterns.com



"If you are programming for the cloud, you are not programming for the cloud."

--Cameron Purdy



Cloud Programmer

Map-Reduce

YQL

FORTRESS

Sawzall

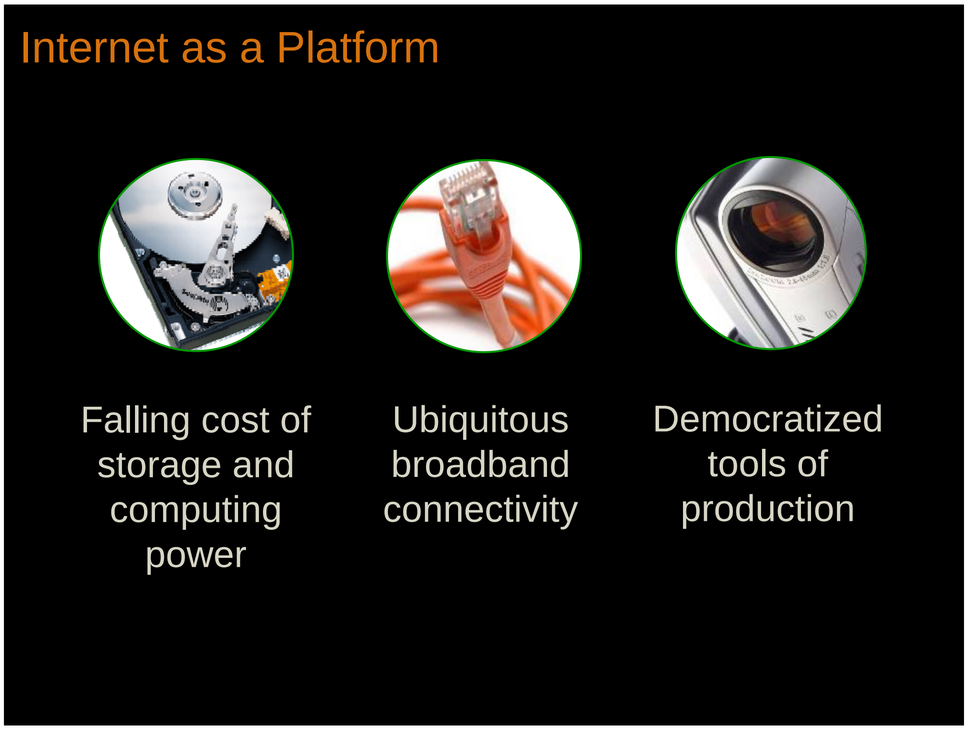
Go

Functional Programming

Inherent Parallelism

Non-Relational Databases

Channel-based Communication

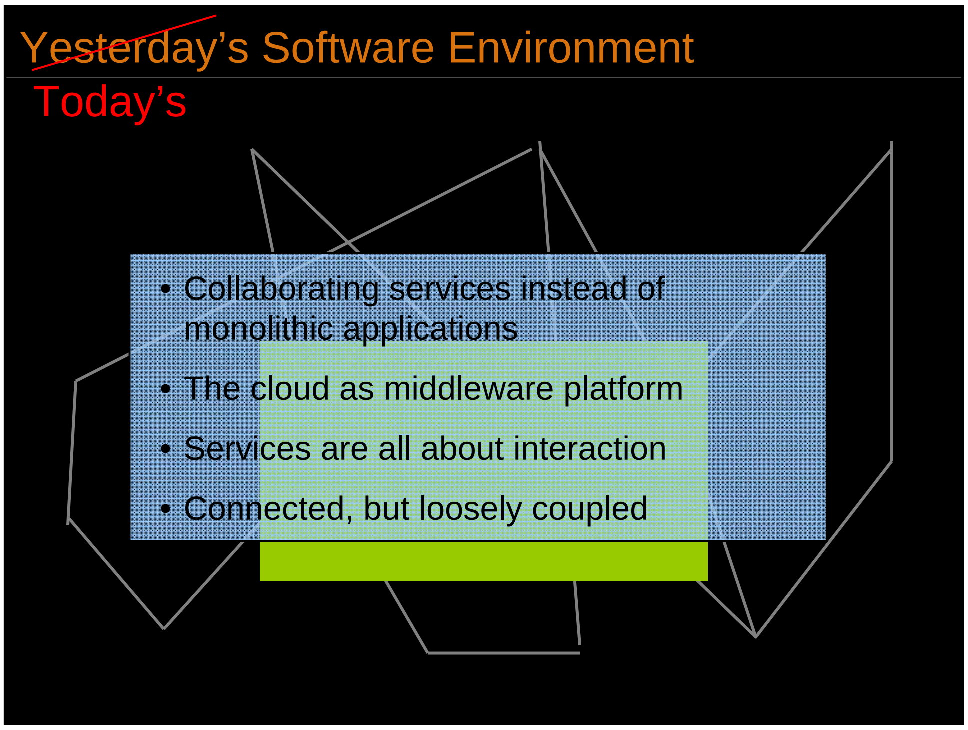


Internet as a Platform

Falling cost of storage and computing power

Ubiquitous broadband connectivity

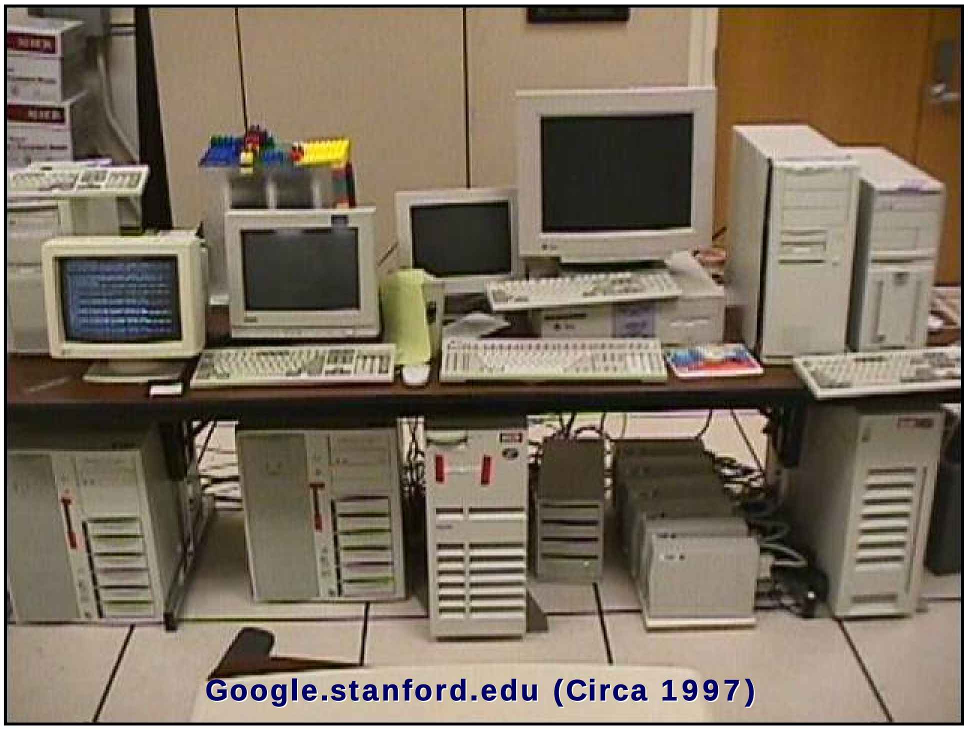
Democratized tools of production



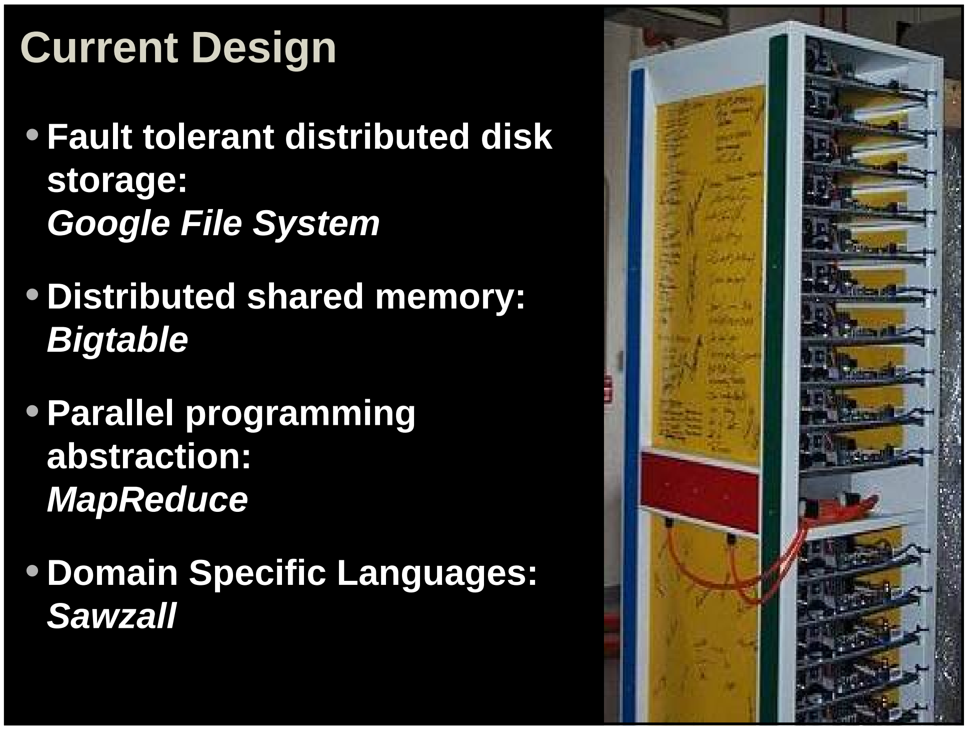
~~Yesterday~~'s Software Environment

Today's

- Collaborating services instead of monolithic applications
- The cloud as middleware platform
- Services are all about interaction
- Connected, but loosely coupled

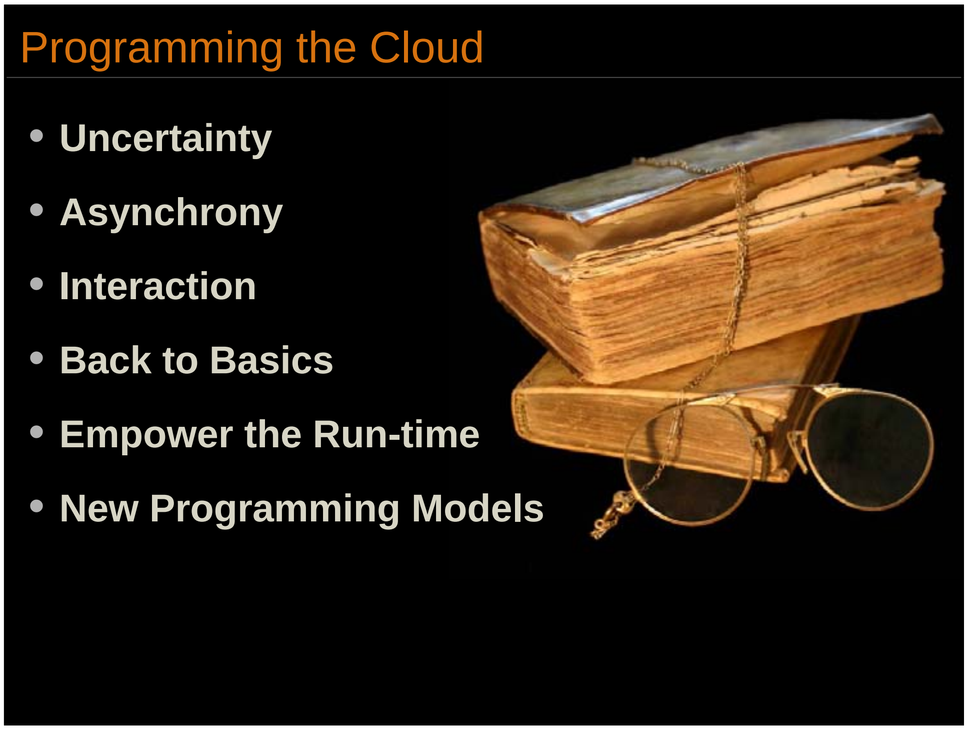


Google.stanford.edu (Circa 1997)



Current Design

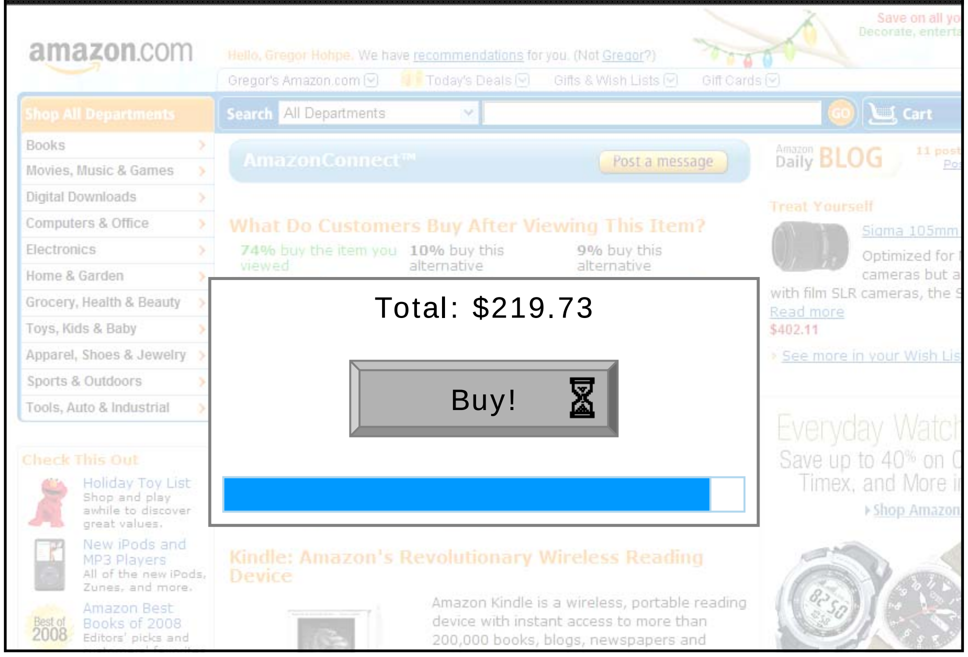
- **Fault tolerant distributed disk storage:**
  Google File System
- **Distributed shared memory:**
  Bigtable
- **Parallel programming abstraction:**
  MapReduce
- **Domain Specific Languages:**
  Sawzall



Programming the Cloud

- **Uncertainty**
- **Asynchrony**
- **Interaction**
- **Back to Basics**
- **Empower the Run-time**
- **New Programming Models**

Uncertainty



Asynchrony

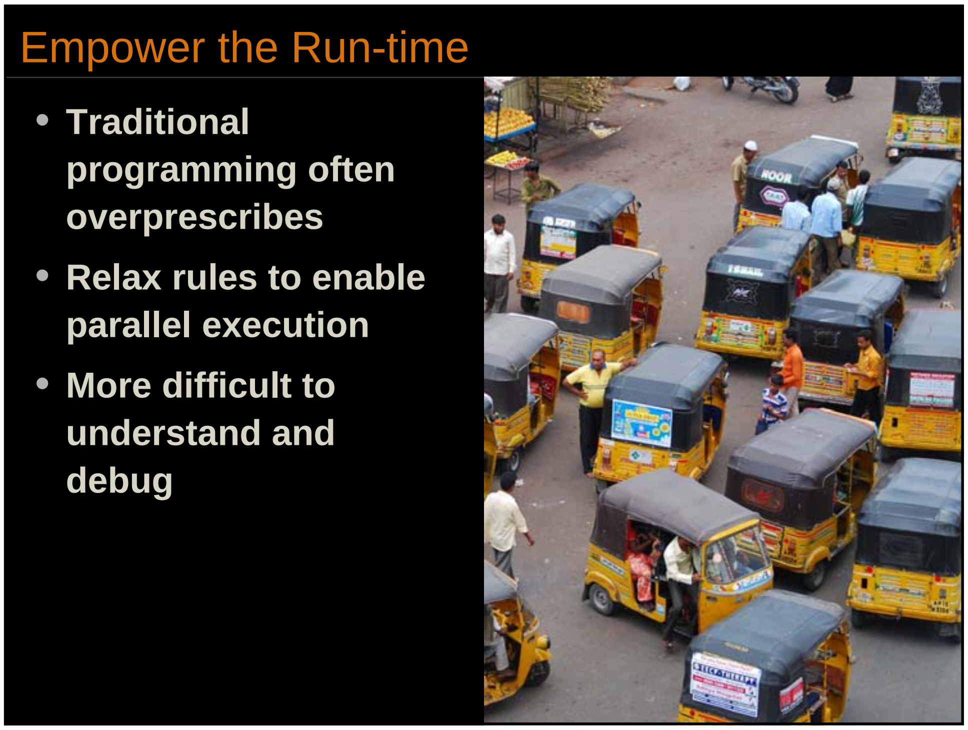
- Waiting for results is not a smart use of a 3 GHz processor
- Therefore, continue processing and handle results as they become available
- Becomes event-driven
- Out-of sequence, time-outs, conversation state
- No read consistency

Interaction

- Not free
- Conversations to overcome uncertainty
- Conversation State

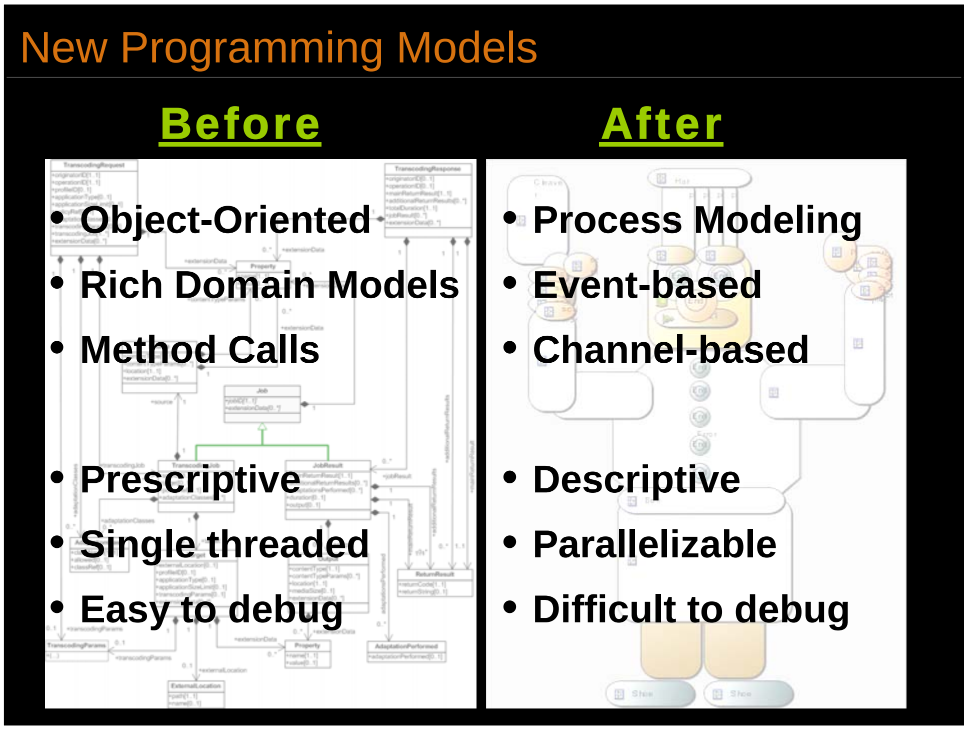
Back to Basics

- Even simple things become complicated in a distributed environment
- Bigtable, not Bigdatabase
- A well understood failure scenario better than an incomprehensible and unproven “failsafe” system

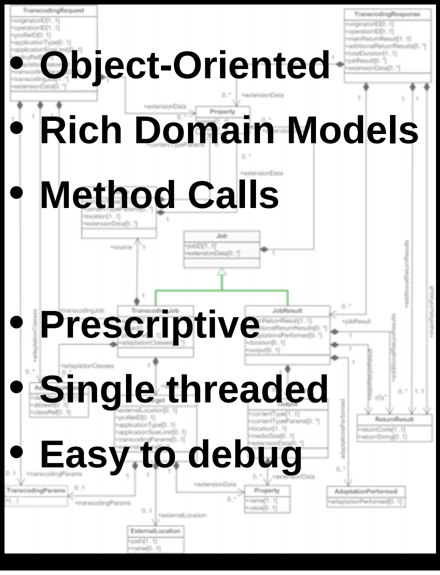


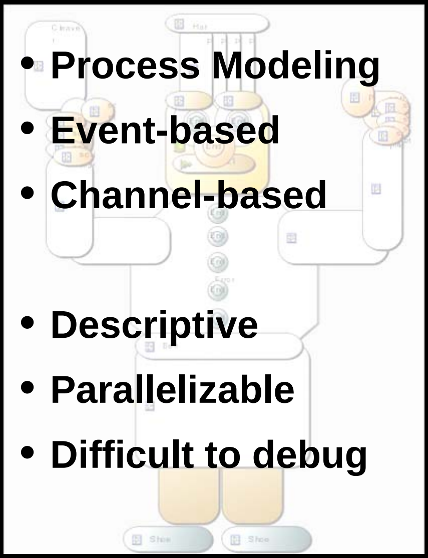
Empower the Run-time

- Traditional programming often overprescribes
- Relax rules to enable parallel execution
- More difficult to understand and debug

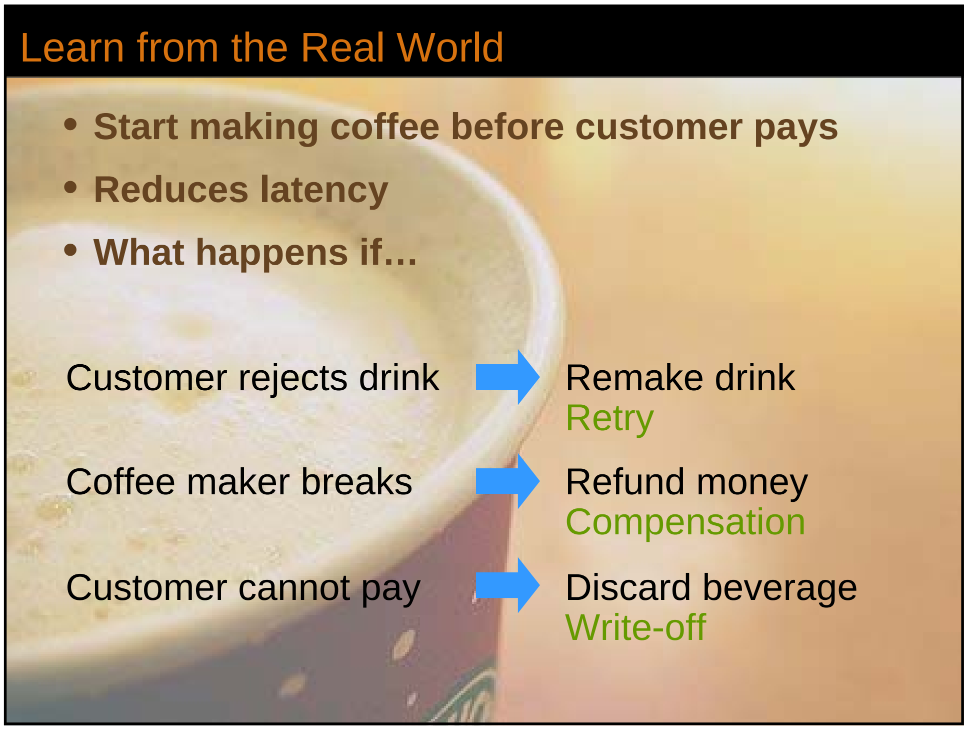


New Programming Models





Before	**After**
Object-Oriented	Process Modeling
Rich Domain Models	Event-based
Method Calls	Channel-based
Prescriptive	Descriptive
Single threaded	Parallelizable
Easy to debug	Difficult to debug



Learn from the Real World

- **Start making coffee before customer pays**
- **Reduces latency**
- **What happens if…**

Customer rejects drink	Remake drink Retry
Coffee maker breaks	Refund money Compensation
Customer cannot pay	Discard beverage Write-off

www.EnterpriseIntegrationPatterns.com

The Addison-Wesley Signature Series
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
Gregor Hohpe
Bobby Woolf
97 Things Every Software Architect Should Know
Collective Wisdom from the Experts
O'REILLY
SOA-Expertenwissen
Methoden, Konzepte und Praxis serviceorientierter Architekturen
dpunkt.verlag

Thank You!

http://www.flickr.com/photos/lorkano/3277969419
http://www.flickr.com/photos/dm-set/3409508275/