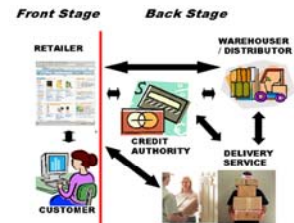
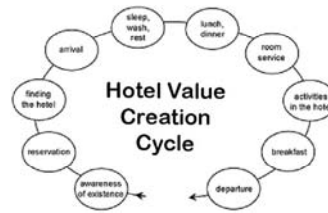


SRII Summit – 26 January 2010

	Business Administration	Business Development	Marketing Management	Marketing and Sales	Product Fulfillment	Financial Control and Accounting
Strategy	Business Planning	Market Planning	Account Planning	Sales Planning	Product Planning	Financial Planning
Operability	Business Unit Tracking	Market Management	Relationship Management	Sales Management	Product Management	Compliance
	Self Assessment	Product Management	Cost Management	Customer Relationship	Customer Relationship	Revenue Action
Effectivity	Self Assessment	Market Director	Sales Administration	Sales	Product Fulfillment	Customer Accounts
	Production Administration	Marketing Campaigns		Customer Care	Inventory Management	General Ledger
				Contract Mgmt		



Service Innovation

Using Design Patterns

Robert J. Glushko
glushko@ischool.berkeley.edu
UC Berkeley School of Information
Information & Service Design Program



Who am I, and why am I here?



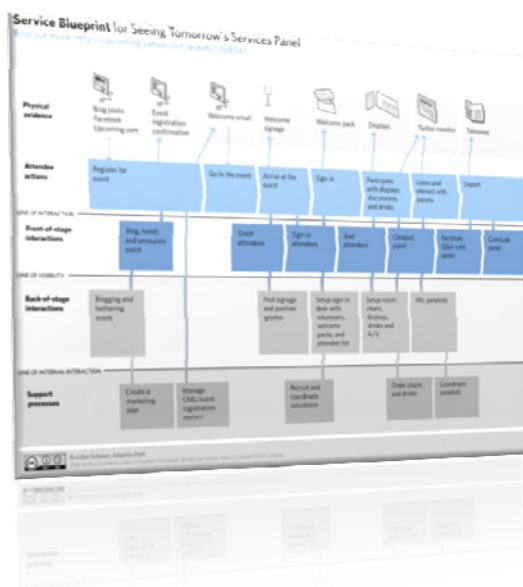
- Bob Glushko is an Adjunct Full Professor at the University of California at Berkeley in the School of Information
- He is one of the founding faculty members of the Information & Service Design program, and “design patterns” for service systems are central to his courses
- Twenty years of “real world” R&D, consulting, and entrepreneurial experience in information systems and service design, content management, electronic publishing, Internet commerce before coming to Berkeley in 2002



A Taxonomy of Service Design Patterns

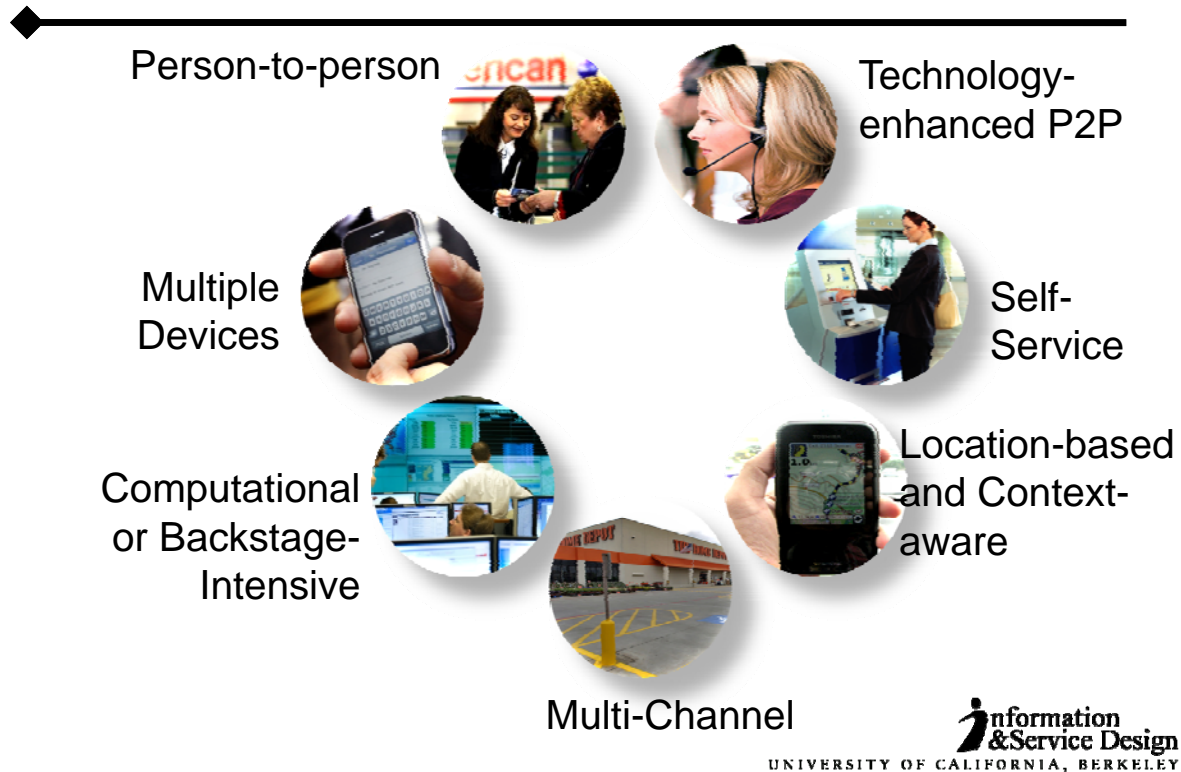
- Patterns that describe interconnected services or processes
 - Component business models
 - Process reference models (SCOR, RosettaNet, ...)
- Patterns that describe “service families” – systematic design alternatives (or evolutionary roadmap) for a particular service
- Patterns that describe service systems

Service System Design Patterns



- Seven contexts – building blocks for service systems
- Adjusting the absolute and relative amount of interpersonal, physical and informational interaction
- Adjusting the line of visibility between the front and back stages
- The number of “touch points” or “stored information equivalents”
- Transparent substitutability

Service Design Contexts



Contexts as Building Blocks

- Describing and designing service systems in terms of the seven contexts makes it much easier to consider **alternative service system designs**:
 - replacing or augmenting a person-to-person service with self-service
 - substituting one service provider for another in the same role (e.g, through outsourcing)
 - eliminating a person-to-person interaction with automation or stored information

Front Stage and Back Stage

- **FRONT STAGE:** Where interactions with the service customer/ consumer happen
- **BACK STAGE:** Produces information and “stuff” needed by the front stage
- Placement of **LINE OF VISIBILITY** is a design parameter

“Touch Points” and Service Intensity / Quality

- Services differ intrinsically in the number of touch points they require to create value; this is often called the **service intensity**
- Traditional P2P service system design assumes that intensity is positively correlated with service quality
- This view lets us treat **intensity as a design parameter** to differentiate service offerings of the same type or industry domain
- The “generic” service offering is a design pattern that can be **increased or reduced in intensity by changing the number of touch points**

Information and Interaction Substitutability

- Capturing, managing, integrating and retrieving information allows service providers to ***substitute information for interaction***
- You don't need high intensity or many touch points if stored information makes interaction unnecessary
- A hotel clerk with a database doesn't need to ask for your room preferences; Amazon doesn't need to ask you about what type of books you like
- Design implication: ***hidden computational services are interchangeable with customer-facing "touch points"***

Transparent Substitutability

- Providers and consumers ***interact by exchanging information through "service interfaces"*** that specify the inputs and outputs of each service
- These interfaces are implicit in P2P encounters, but always explicit for non-human actors
- In the purest vision of "service oriented architecture," the interfaces are abstract, enabling ***transparent substitution of one provider for another*** to optimize service quality for each consumer

Summary

- Design patterns have a central place in engineering, architecture and computing but haven't been a focus of service system design and operations
- The more abstract conception of services and service interfaces embodied in design patterns assists in analysis, encourages best practices, and facilitates innovation
- These design patterns can be taught to and used successfully by university students and practitioners