# Substituting Information for Interaction: A Framework for Personalization in Service Encounters and Service Systems

Robert J. Glushko
School of Information
University of California
Berkeley, California
e-mail: glushko@berkeley.edu

Karen Nomorosa
Rearden Commerce
Foster City, California
e-mail: karen.nomorosa@reardencommerce.com

## ABSTRACT

We compare person-to-person service encounters with those in which the service provider is an information system to identify the capabilities needed to personalize a service encounter. We suggest "substituting information for interaction" as a principle that unifies these different types of encounters whenever the information needed to create value in a service system accumulates incrementally through human or automated customer interactions. We review research and practice in computer science, artificial intelligence, data mining, machine learning, and information systems design to bring an interdisciplinary robustness to our conceptual proposal.

Human service providers and automated service systems both need (1) a SERVICE MODEL MANAGER that stores information about how a customer requests a service; (2) a CUSTOMER MODEL MANAGER that stores information about customers and preferences; (3) a RECOMMENDATION SYSTEM MANAGER that uses service models, customer models, and contextual information to adapt the service at delivery time; (4) a LEARNING SYSTEM that analyzes previous service encounters to refine service and customer models, and a 5) SERVICE MONITORING SYSTEM that monitors the status of service delivery.

The substitution concepts and mechanisms we propose highlight the range of design choices and help managers evaluate whether a human interaction or information exchange creates or undermines value in a service system.

# INTRODUCTION

The infusion of information technology into service systems and business mandates for greater service provider productivity requires us to think differently about service intensity, personalization, and other factors that define the customer experience. By replacing or supplementing interpersonal interactions with information exchanges, technology transforms experiential service encounters into more information-intensive ones. At the same time, CRM and analytics software enable service providers to personalize their offerings to customers by making recommendations or default choices, preventing the "paralysis of choice" by giving customer fewer options (Taylor 2012, p 31) while "engineering a unique shopping experience for each segment" (Verhoef et al., 2010 p. 122).

Some types of Customer Experience Management (abbreviated as CEM or CXM) software monitor in real-time how customers interact on web sites (Burke, 2010), proactively intervening to prevent customers from abandoning their shopping carts by making dynamic adjustments to catalog structure, content and pricing to induce the customer to buy an item (Shafer, Konstan, and Riedl. 2001). And for services where the request to perform the service and its result can be expressed entirely and unambiguously through information exchanges, both the service provider and service customer can be computational agents, enabling data feeds, objects with RFID tags, sensors, and myriad software applications to be parts of the "back stage" of service systems that work on behalf of the human customers without requiring them to participate in an explicit service encounter (Glushko 2010).

Our goal in this article is to propose a new conceptual framework for understanding the complicated relationships among these concepts of service intensity, customer preferences, personalization, customer or provider-initiated interactions, and service provider productivity. This framework applies when the information needed to create value in a service system accumulates incrementally through customer interactions or transactions with human service providers, with automated ones, or in combination. As noted by Berry, Carbone and Haeckel (2002), this relates to the combination of functional, humanic and mechanic clues that make up the customer experience. In the Apte and Mason classification of three types of service intensity (Apte and Mason 1995), this domain is the intersection of the symbolic manipulation and customer contact regions of activity. Apte and Mason do not pay any particular attention to this region, but we think it represents a critical part of the design space for service systems. For services in this intersection – exemplified by education, healthcare, business travel, retail sales and consulting - the balance between information-intensive and experience-intensive experience is a design decision rather than an intrinsic difference.

Others have analyzed the degree and nature of technology use in service systems and its impact on customer experience. Bitner, Brown and Meuter (2000) contrasted the goals of technology infusion to customize the service offering, to recover from service failure, or to create a memorable experience with “spontaneous delight.” Froehle and Roth (2004) distinguished five modes (or “archetypes”) of customer contact in relation to technology that range from “technology-free encounters to “technology-generated” ones and argued that customers evaluate their experiences differently in each mode. Similarly, Fitzsimmons and Fitzsimmons (2006) contrast “technology-assisted” and “technology-facilitated” services, and Glushko (2010) notes that technology can be used in a service encounter by the provider alone, by the provider and customer together, or by the customer alone. Verhoef et al (2009) note a contrast between “active” technology that requires customer interaction and “passive” technology that pushes information to customers (e.g., a shopping cart sensor that totals the cost of the goods in the cart), but they only hint at the theoretical or design implications. Similar examples in which technology reduces or eliminates customer work by shifting tasks to the service provider are presented by Campbell, Maglio, and Davis (2011) to define a category of “super-service.”

*Plan for the Article*

Our work extends these efforts to propose categories or levels of service interaction in a novel way by proposing “substituting information for interaction” as a unifying principle for analyzing and designing personalized service encounters and service systems. Previous approaches typically make a binary contrast between person-to-person encounters, with or without technology support, and self-service interaction, where the customer interacts with technology as part of the service system. We are in effect introducing a third category of “non-interaction”, in which the service system uses information to create value in a service encounter with no need for customer interaction at all. The substitution of interaction can take place in person-to-person service encounters, in automated encounters between computational agents, and throughout the complicated design space between them in which technology and information systems are part of service systems that include human actors.

Our conceptualization fits with Shostack’s (1985) definition of a service encounter: “a period of time during which a consumer directly interacts with a service.” As noted by Bitner, Booms and Tetreault (1990, p. 72), Shostack’s definition “encompasses all aspects of the service firm with which the consumer may interact, including its personnel, its physical facilities, and other visible elements,” does not “limit the encounter to the interpersonal interactions between the customer and the firm, and in fact suggests that service encounters can occur without any human interaction element.”

In the following section we set the stage for our proposal with brief descriptions of five service encounters. All of them involve encounters between a service customer and a service provider in which the customer is seeking a restaurant recommendation. But the nature of the recommendation and the nature of the interactions between the provider and customer differ in ways that suggest the necessary functional components of a conceptual framework that can personalize a service encounter by substituting information for interaction.

In the sections that follow, we describe the conceptual framework in detail, motivating it both from our analysis of the five service encounters and from a review of relevant research and practice in computer science, artificial intelligence, data mining, machine learning, and information systems design. Our complete framework has yet to be implemented, so we conclude the paper with logical arguments for necessity and sufficiency based on what would happen if we selectively remove any component we have proposed.

## FIVE SERVICE ENCOUNTERS

ENCOUNTER 1: A business traveler arrives at his hotel, and at check in, he asks the hotel front desk clerk if he can suggest "a nice pasta place" for dinner that evening. The front desk clerk says "There are several Italian restaurants in the neighborhood; the closest one is Pasta House just down the street."

ENCOUNTER 2: A business traveler arrives at his hotel, and after checking in, asks the concierge to suggest "a nice pasta place" for dinner that evening. The concierge asks if the traveler is dining alone, and after being told that there will four people at dinner, asks if they are getting together for a business meeting or for pleasure. When the traveler replies "for business" the concierge suggests Il Fornaio, describing it as "an upscale place that is quiet and spacious enough for private business conversations."

ENCOUNTER 3: A business traveler arrives at his hotel, and after checking in, asks the concierge to suggest "a nice pasta place" for dinner that evening. The traveler is a frequent hotel guest and the concierge remembers that on a previous trip the guest had liked the restaurant she'd recommended and in response to his request asks "Would you like to go back to Il Fornaio or try something different?"

ENCOUNTER 4: A business traveler arrives at his hotel, and while waiting to check in, types "pasta restaurant" into a search engine application on his cell phone. The top search result is a family-oriented chain restaurant called Pasta House, followed by several others that serve Italian cuisine, including Il Fornaio which is listed seventh. All of them are within a mile of the hotel.

ENCOUNTER 5: A business traveler arrives at his hotel, and types "pasta restaurant" into the search box on his "smart digital assistant" application on his cell phone. Within seconds the app suggests "Il Fornaio" with three available reservation times for a party of 4, a reminder of the previous dinner there, and a (Confirm/Cancel) prompt. When the traveler selects "Cancel," the app proposes two other nearby Italian restaurants with similar Zagat ratings to Il Fornaio, and he chooses the first. Pasta House is not one of the two alternatives even though it is closer to the hotel because its description and ratings make it less desirable for a business dinner.

*A Conventional Analysis of the Five Service Encounters*

An experience-centric design perspective sees a profound contrast between encounters 1-3, in which the customer interacts with a human service provider, and encounters 4-5, in which the customer interacts with an information system. Furthermore, from this perspective encounter 1 is lower in service intensity than encounter 2. In the former the desk clerk treats the guest in a generic manner, does not engage him relationally, and makes no attempt to personalize his response. The desk clerk answers the question about nearby restaurants in a way that might be deprecated for not involving much more than "the mere transactional exchange of information" (Zomerdijk and Voss 2010, p.69).

In encounter 2, because the concierge does not know the guest, she asks a few questions first. These enable her to identify the unstated requirements in the customer's query about "a nice pasta place" so that she can rely on her knowledge of typical business guests and local restaurants to recommend a restaurant that will satisfy his request.

In encounter 3, because the concierge remembers the guest from previous stays at the hotel, the encounter differs in some important ways. She does not need to ask the refining questions required in encounter 2 to classify the guest, and instead bases her recommendation on her memory of the customer's previous restaurant choice. And because she knows the traveler has been to Il Fornaio, she does not need to describe it.

Measured by the number of interactions, encounter 3 is of lower intensity than encounter 2, but the customer probably appreciates being remembered by the concierge and reminded of his enjoyable restaurant experience when he last stayed at the hotel.

Encounters 4 and 5 would both conventionally be considered self-service applications, and both make personalized recommendations. But in encounter 4, the degree of personalization is minimal, based solely on the traveler's current location, which the search engine can obtain from the cell phone's GPS capability, from a database that lists the hotel's wi-fi "hotspot," or from cell phone tower triangulation (Kansal, Goraczko, and Zhao, 2007; Roxin et al, 2007).

In encounter 5, the "smart digital assistant" is the user interface to a complex and highly-integrated service system that makes highly-personalized recommendations, just as a smart human personal assistant would. The application combines information about the traveler's current itinerary and appointment calendar (where it discovers a scheduled dinner meeting for 4 people with no location specified) with three other services: the smart phone's GPS service to determine the current location, a restaurant reservation service, and a restaurant rating service. It uses information about the traveler's prior dining experiences to filter the results from the restaurant reservation service to those that are comparable in service quality to previous reservations made on business trips, and proposes a confirmation of a previous choice rather than a more open-ended selection from an unfiltered list. But this complex configuration of interdependent service alternatives is delivered with minimal interaction intensity, requiring just two taps on the screen of the smart phone to reject the suggested restaurant and make a reservation at another.

*An Alternative Analysis of the Five Service Encounters*

Focusing on the presence or absence of a human service provider is taking too narrow a view of the service encounter. Instead, we consider the relationship between the service provider and the customer and analyze how that relationship affects the service encounter, especially with respect to the nature and degree of personalization.

In encounters 1 and 4, there is no relationship between the service provider and the customer, which means that every customer gets the same response from either the front desk clerk or the search engine. The customer's location enables the response to be localized, but this is as minimal as personalization can be.

In encounter 2, because the traveler has never stayed at the hotel, he has no existing relationship with the concierge. The concierge makes a restaurant recommendation, but this is coarse personalization based on a model of customer type or customer segment, not on the basis of the customer's own experiences.

Encounters 3 and 5 are very similar in our analysis. In both situations there is a pre-existing relationship between the service provider and the customer. The personalized restaurant recommendation is based on customer-specific transaction history. Encounter 5 has the highest degree of personalization because it also satisfies several additional constraints imposed by the customer's calendar and other information services.

Most importantly, in encounters 3 and 5, some interactions between a customer and a service provider have been eliminated by substituting stored information (from the concierge's memory in encounter 3, from the digital assistant app in encounter 5) or by requesting information from other services or information sources. Furthermore, this substitution has improved the quality of the customer's experience, especially in encounter 5. Because of the efficient way in which the smart digital assistant enables the customer to choose his restaurant for the evening from three satisficing alternatives, it captures additional information that the concierge does not have and cannot easily obtain.

An experience designer might object here and argue that analyzing encounters 3 and 5 with the same framework ignores the essence of experiential services, which is that they create a relationship with the customer rather than just carry out a transaction. If the hotel followed an experience-centric service design philosophy the concierge in encounter 3 would have engaged in a longer conversation with the guest in which she more explicitly welcomed him back to the hotel, asked how his flight was, and so on. This variation of encounter 3 would have more intensity, and it would surely be considered more relational and less transactional. But it would not affect the nature or degree of the personalization in the service encounter.

## A CONCEPTUAL MODEL OF THE PERSONALIZED SERVICE ENCOUNTER

Taking this alternative perspective on the five encounters and focusing on the commonalities between the service provided by the concierge and the service provided by the smart digital assistant suggests some functions or capabilities that any service system needs to personalize a service encounter. We have also reviewed relevant research and practice in computer science, artificial intelligence, data mining, machine learning, and information systems design to bring an interdisciplinary robustness to our proposal. Our proposal refines the process-oriented framework for personalization proposed by Adomavicius and Tuzhulin (2005), who contrast provider-centric, consumer-centric, and market-centric architectures for personalization of single services rather than sets of overlapping services as we do. In addition, our ongoing work with Rearden Commerce, a software firm focused on "smart commerce" has given us unique insights into architectural and implementation implications of personalization, making our framework more ambitious than any current software implementation of a personalization platform;

only the Daidalos project being conducted by a consortium of European telecommunications operators has a comparably broad scope (Taylor et al. 2011).

Because it is important that the framework continue to apply to the person-to-person encounter, our proposal is agnostic with respect to most of the architectural decisions about how these functions or capabilities might be partitioned among software components in an automated service system. Therefore, our proposal should be taken as a future-oriented statement of requirements and conceptual model and definitely not as a software architecture or design document for any existing service system.

Both the human concierge and the information system need (1) a CUSTOMER MODEL MANAGER that stores information about customers and their preferences with respect to a requested service; (2) a SERVICE MODEL MANAGER that stores information about how a customer must describe or request a service; (3) a RECOMMENDATION SYSTEM MANAGER that uses service models, customer models, contextual information and content to adapt the service at delivery time; and (4) a SERVICE MONITORING SYSTEM, which monitors the status of service delivery so it can ensure that the service plan is successful, possibly by responding to unexpected events or service failures that would require changes to the service delivery plan. This system has only a minor role in encounters 3 and 5. The concierge and the digital assistant application design the customer's plan for a restaurant service encounter, propose the plan, and refine it after getting feedback from the customer, but neither the concierge nor the digital assistant app has any involvement in the actual delivery of the restaurant's service.

An essential part of many service systems is a fifth function or capability, (5) a LEARNING SYSTEM that analyzes information about service encounters to refine service and customer models. (All five of these functions or capabilities are involved in encounters 3 and 5 in the previous section of this paper;

they are of course “mental” or “cognitive” functions for the concierge and software functions for the smart digital assistant).

These five functional components and their relationships are shown in Figure 1. In the remainder of this paper we will discuss each of these components in turn after we introduce the foundational concepts of the CUSTOMER MODEL and the SERVICE MODEL.

<insert Figure 1 here>

## THE CUSTOMER MODEL

Every definition of “service” has a notion of “customer” in it because the essential activity in a service is a provider creating value with or for a customer.  Many businesses design services targeted for different customer segments that can be specified using numerous overlapping criteria, including demographic variables, product or behavior choices, and preferred interaction locations or channels. For example, an airline might segment its customers according to their ticketing class, travel frequency, and home airport. These offerings are not personalized to individual customers, but if the range of customer segments is broad enough, the customer might be able to select one that satisfies his individual preferences.

People naturally and without explicit effort create models of people and the contexts in which we encounter different types of people (Pruitt and Grudin, 2003).  We rely on these models to understand what people do and we can extrapolate or "run the model” with hypothetical inputs to predict behavior in new situations.  These models can be detailed and multidimensional enough that we can determine when someone's behavior is "out of character.”   This knowledge about people in the context of service delivery is often called a CUSTOMER MODEL because the properties, traits, behaviors, or preferences that characterize a customer are aggregated to describe the customer in an integrated way (Towle and Quinn, 2000, Wiederhold 1997).    We cannot directly examine the customer models in the mind of a hotel concierge the way we can examine those in a computer program, but in either case some package of knowledge can be associated with a specific customer.

The information components in the customer model differ in their persistence and useful lifetimes. The customer's name and date of birth might be valid forever, but "last purchase" and "current location" obviously are not. Some preferences and goals change in an evolutionary or continuous fashion, while others change discontinuously in response to a significant life event like gaining or losing a job, gaining or losing a spouse, moving across the country, and so on.

In person-to-person service contexts, delivering customized or personalized service is the responsibility of the frontline or customer contact employees (Gwinner et al 2005; Teboul 2006). This means that service employees must internalize the customer models for each customer segment so they can classify customers in service encounters. Service providers then use their customer models to adapt their interactions and communications with customers, to make recommendations to them, and predict their response to these interactions, communications and recommendations. The accuracy of these recommendations and predictions is greater when the model is more detailed, and a person whose job depends on making good ones, like a hotel concierge, will over time develop precise models of the types of people they encounter and will learn the questions that most efficiently identify which customer model type to apply, as we saw in encounter 2.

This "delivery time" adaptation of a service is often described as "recommendation" or "personalization" but we note that these terms are not always used in a consistent manner. In ordinary language use any proposal might be called a "recommendation," whereas a "personalization" is usually more narrowly defined as a change or adaptation based on a customer model. However, in service design contexts a "recommendation" is generally a proposal about the content of a service offering, while a "personalization" can involve a broader set of changes in the nature of the interaction between the service provider and the customer, the channel in which it takes place, or the device / technology used to carry it out.

After or instead of proposing candidate customer segments on the basis of a priori characteristics, customer segments can be refined or identified through the analysis of the aggregated record of customer transactions and interactions to identify groups or clusters of customers who have similar behaviors or who made similar choices. This kind of analysis has enabled airlines to discover that not all frequent fliers are equally profitable and has suggested ways in which they can tune their loyalty programs to reward the most profitable ones (Reinartz and Kumar, 2002). Similarly, analysis of preference judgments by frequent airline travelers showed that conventional segmentation into leisure and business travelers fails to capture the distinctive requirements of the most frequent fliers (Teichert, Shehu, and von Wartburg, 2008). In our framework this kind of refinement of customer segments is carried out by the LEARNING SYSTEM.

“User model” is a concept that is closely related to “customer model.” The notion of “user model” has a long history in a diverse set of computer applications, including intelligent user interfaces, tutoring systems, recommendation systems, help systems, and more recently and ubiquitously in search engines (e.g., Rich 1979; Kass and Finnin 1988; Shen, Tan, and Zhai 2005). The common idea across all these domains is that information about user knowledge, goals, and preferences can be used to predict the user’s next step or need for information. The techniques for creating and building user models are equally diverse, ranging from “hand-crafted” rules in expert systems to data mining and machine learning algorithms embodied in connectionist or Bayesian networks (Brusilovsky, Kobsa, and Nejdl 2007; Hastie, Tibshirani, and Friedman, 2009).

There are two important differences between “customer segments” and “user models.” The first is that even when customer segments are based on multiple factors, they create relatively broad categories of customers. Put another way, the number of customer segments is invariably much smaller than the number of customers. For example, the most commonly used customer segmentation model in direct marketing, RFM segmentation, creates 125 customer segments by sorting them into quintiles by the recency, frequency, and monetary value of their purchases (e.g., (Tsipsis and Chorianopolous 2009)).

While demographic or other slow-changing attributes are also the starting point in user models, they contrast with customer segments because they typically extend this base with greater amounts of highly granular and more dynamic data. This enables systems or services with user models to approach the goal of personalizing every interaction. For example, every student using an intelligent tutoring program is represented by a user model based on his sequence of questions and his specific pattern of correct and incorrect answers (Brusilovsky 1999).

Representing every user with a user model requires the second important difference with customer segments, which is that the former are always implemented as computer programs. This enables a system driven by a user model to react in "real time" (or nearly so) to the user's implicit or explicit actions or information requests. Some marketers describe the real-time use of user models to adapt interactions with customers as "micro-segmentation" (Taylor and Raden 2007).

Nevertheless, despite these differences between customer segments and user models, we can ignore the difference between them and use the concept of "customer model" because the contrasts between them are not essential to our framework.

## THE CUSTOMER MODEL MANAGER

The CUSTOMER MODEL MANAGER has a primary role in managing the customer models needed to implement customer segmentation and personalization strategies for a particular service. It also manages the customer models for specific customers when a broad model for a group of customers can be individualized to the extent that information about specific individuals can be reliably obtained. Because much of the information in a customer model does not depend on the service, the overlapping components can be combined in a unified or combined model.

Individualized customer information can be elicited explicitly in person-to-person encounters, or by using questionnaires, registration forms, or similar information gathering methods. Because these interactions and information exchanges are explicit, they set clear user expectations about the degree to which a service can be personalized, because the customer knows precisely what the service provider knows about him.

But there are downsides to relying on explicit interactions to obtain information for a customer model. One is that the customer might deliberately provide inaccurate information to protect his privacy or self-esteem, undermining the accuracy of the customer model. A more general downside of explicit interactions is that they impose costs on the customer in time or effort. In some service systems customers will tolerate “batch” interactions to elicit information, as they do when they visit a doctor’s office. Patients expect or hope that the healthcare service system will be able to use the information to give them better service, perhaps by aggregating it in an electronic health record shared by numerous service providers. But few people would comply with a request at check-in to fill out a detailed preferences form that would populate an electronic hotel record.

As a result, the information in most customer models must be acquired implicitly. A service system can record every choice a customer makes and every detail of his navigation or browsing behavior for online services, and as we all know from targeted advertising by search engines, this information can be exploited nearly instantaneously to personalize interactions and information. See Castellano et al (2009) for a comprehensive review of Web-based personalization concepts and techniques.

The downside of collecting customer information implicitly is that the customer's behavior might be shaped or distorted by an unidentified role or context that creates "noise" in the customer model. This is a problem faced by online retailers who would like to recommend items to customers based on their previous purchases. If a middle-aged man occasionally buys dolls or toys for his kids, for example, it would be hard to predict what he might want to buy next.

*Context in the Customer Model*

The customer model manager has an additional function of acquiring and managing information about the customer's current context. "Context" is defined as "implicit situational information" or "any information that characterizes a situation related to the interactions between users, applications, and the surrounding environment" (Dey 2001, p. 4). In practice this open-ended definition is operationalized as "location + activity" and even more simply as "where, when, and how" is the customer consuming the service or participating in a service encounter. The current location is a key part of the customer model for many services, as demonstrated by the numerous applications for smart phones that personalize the experience or the information received by a customer using the location information provided by the phone. In many service systems this location information is used automatically to send location-based service offers to mobile phones, like invitations to or discounts for nearby stores or restaurants.

## THE SERVICE MODEL

Every service encounter involves two actors: a service provider and a service consumer. (Many other words have been proposed as synonyms. The most common alternatives to "provider" are "producer" and "server." Alternatives for "consumer" include "client," "customer," "requestor," and "co-producer.") "Actor" is used here in an abstract sense to include both human and computational entities, just as it is in "use cases" and other methods for modeling the interactions in a system (Cockburn 2000).

Service encounters that are one-to-many, like a lecture given by a professor simultaneously to many students, can always be modeled as sets of pairwise interactions between two actors.

The interactions between the two actors take place when the service consumer explicitly or implicitly requests the service. The SERVICE MODEL is the specification of the information needed by the provider to deliver the service, the results or outputs generated by the provider, and definitions of successful and failed service delivery. “Service interface” is the conventional term in discussions of computational services (Erl, 2004), but to achieve more generality and avoid confusion with the notion of “user interface,” we prefer “service model.” This SERVICE MODEL is always explicit with computational actors, where well-defined inputs and outputs specified as “application program interfaces” (APIs) or document specifications encoded as XML schemas are a prerequisite for the infusion of computation or automation, and where the interaction is intrinsically and exclusively an exchange of information (Glushko and McGrath, 2005). For example, in the two automated encounters 4 and 5, the text string “pasta restaurant” is the input request to a search service whose response is a text list of one or more restaurants.

In contrast, the service model is often implicit and underspecified in person-to-person encounters, and information exchange is only a part of what goes on. The literal request in all three of the person-to-person encounters is for a “nice pasta place” but the service provider first has to recognize that “place” refers to a restaurant. This is easy for a human but not for most automated services that process service requests in a more literal way and do not know to treat “place” as a synonym for “restaurant” in the customer’s query.

Related services will have similar and overlapping service models. If the service is “select restaurant” the service model will contain a location attribute and additional characteristics like the cuisine or price level. Since these additional characteristics were not mentioned in the customer’s request to the concierge in encounter 2, the concierge needs to ask the customer for the information needed to complete an instantiation of the service model. If the service is “make restaurant reservation” the service model must include the restaurant name, the time, and the number of people in the party.

*Simple Service Models*

Some service models are “simple” because they follow a “Query-Response” pattern in which the service requestor asks for a response or information that is not dependent on the identity or context of the requestor (ebXML, 2001). The service request can be specified with little information, and any information needed to respond is usually either static or relatively slow-changing. Simple services do not – or cannot – take account of the relationship or lack thereof between the service provider and service customer, so they cannot be personalized. “Current weather” is a simple service; a query to a weather service for “the current temperature in Honolulu” has the same answer whether the requestor is a person or a computerized web service.

Other service models are more complex because they require the service provider to apply some logic to determine or compute a response, which might be dependent on the identity or context of the requestor or on the responses to prior service requests that established a relationship between the two actors. These service models conform to “Request-Response” or “Request-Confirm” patterns in the ebXML framework. In encounters 2 and 3 the concierge’s answer to the guest’s request to recommend “a nice pasta place for dinner tonight” first requires work by the concierge to determine what “nice” means to the guest and then requires a second request to a restaurant reservation service to determine availability.

*Overlapping Service Models*

Many services have logical or causal dependencies, which mean that their service models overlap in some way. For example, all travel-related services (airlines, hotels, ground transportation) need the traveler’s identity and the time and location of his travel. A trip to New York might involve all of these services, and they need to fit together in time and location for the trip to make sense. The hotel reservation needs to begin the day the flight arrives in the destination city, the limousine service needs to meet the traveler shortly after the plane lands, and the restaurant reservation should be convenient to the hotel.

This overlap between service models can be seen in Figure 2, which shows the input forms for different web sites that a traveler might visit to plan a trip.

<insert Figure 2 here>

Another way for service models to overlap is for the output of one service to function as the input to another in a service system that combines component services; for example, the item selected from the “catalog service” in an online store becomes the object of the “picking service” at the warehouse and ultimately the focus of the “shipping service” that delivers the item to the customer. The same resource or object is involved in all three services, but it has a different role in each service and might be described differently in their service models. Service models also overlap when an enterprise offers multiple services in the same location to the same customers; in addition to offering guest rooms, a hotel might have a restaurant, a health club, a tour desk, and a business center.

Services with overlapping service models naturally come together to produce sequences or sets of service encounters that create experiences that unfold over time. From the customer’s perspective overlapping services seem inherently interconnected and the customer naturally wants to experience them in an integrated and holistic way. Advocates of experience design describe these multi-service encounters in highly metaphorical ways that express this customer intent. The customer is said to embark on a “customer journey” or “experience cycle” (Dubberly and Evenson 2008), following the “trajectory of interaction” (Benford et al. 2009), or the “customer corridor” (Meyer and Schwager 2007).

*Service Composition*

But when services are delivered by separate providers, it can be difficult or tedious for a service requestor or customer to combine their overlapping service models to create the desired integrated experience. Instead, intermediaries often emerge to take on this role, with familiar examples being a travel agent or hotel concierge who works on behalf of a customer to make a set of interdependent reservations with different service providers. The intermediary is often a person (an agent, broker, personal assistant, “middle-man”, etc.) but can also be a computational process. In our framework we call this intermediary the SERVICE MODEL MANAGER as a neutral term that covers both human and computational agents whose purpose is to combine or compose separate service models.

Starting in the late 1990s, when business services in great numbers began to be implemented as web services, service composition became a widely-researched topic in computer science (Casati, Ilnicki, and Jin, 2000; Milanovic and Malek, 2004, Rao and Su, 2005). Most research sought to apply artificial intelligence planning techniques to cross-enterprise workflow models to yield automated methods for discovering and interconnecting services into the desired business process. Because services designed by different firms might have service models that disagree in their structure or semantics - consider the number of different ways that a common concept like “address” can be described - identifying the potentially overlapping components of service models and “gluing” them together are very challenging problems. One class of solutions assumes standards for service description to prevent these vocabulary problems (Ankolekar et al, 2002; OASIS, 2006), while another proposes to repair the vocabulary mismatches with sophisticated formal representations and reasoning algorithms (Martin et al, 2007; Medjahed, Boughettaya, and Elmagaramid, 2003).

More recently, work on service composition seems to have taken a more pragmatic turn. The suitability of a service in a service composition can also depend on its trustworthiness, reliability, cost, and numerous other quality and non-functional parameters that are difficult to specify in a computer-processable manner. As a result, many researchers and most practitioners have replaced the goal of automated service discovery and composition with the goal of automatically instantiating and delivering pre-composed compositions of services from known service providers in which these other issues have been resolved through more traditional business negotiation. We call these "service composition models" to emphasize that they are built from service models.

In many respects this makes the work of an automated service intermediary more like that of a human intermediary. After each identifies a set of service provider types whose combined services deliver value to customers in a compatible and integrated way in a service composition model, their task is to create useful instantiations of those services to satisfy specific customers. For example, both a human travel agent and one implemented as an information service would know the service composition model called "a business trip to New York City" and know how to find combinations of reserved services to instantiate its component service models. It is not the role of the intermediary to discover new service composition models; in our framework this is the job of the LEARNING SYSTEM.

## THE SERVICE MODEL MANAGER

The SERVICE MODEL MANAGER has two essential functions. The first is a knowledge management function to maintain the separate service models from different service providers and the service composition models in which they participate to satisfy typical customer goals. The second function is to use the service composition models as templates or patterns to be instantiated to meet the specific requirements of an individual customer.

A critical underpinning of this management of service models is the use of an ontology to represent the conceptual or semantic relationships among the services. "Ontology" is a branch of philosophy concerned with what exists in reality and the general features and relations of whatever that might be (Hofweber, 2009). Computer science has adopted “ontology” to refer to any computer-processable resource that represents the relationships among words and meanings in some knowledge domain (Guarino, 1998). This definition makes even a simple digital dictionary into an ontology, but more typically an ontology is expressed using a formal logic-based language that makes every semantic assumption explicit (Hepp 2008).

Understanding the relationships between words and concepts is essential for a service system in an information-intensive domain to do better than simple information retrieval in response to a customer request. A search engine like that in our encounter 4 or an automated restaurant reservation service that only accepts simple text queries could propose restaurants whose descriptions include the words “Italian” and “pasta,” and perhaps organize the matching restaurants by location and price. Many such services are in wide use today. But the contrast between these literal responses with the nuanced and personalized responses from a hotel concierge in encounters 2 and 3 and the smart digital assistant in encounter 5 demonstrates why we propose an ontology as part of service model management.

If a hotel guest asks the concierge to “make me a reservation at an Italian restaurant because I want pasta tonight”, the concierge will select a nearby Italian restaurant whose quality of service is consistent with that of the hotel. The concierge remembers or infers the unstated attributes of location, price range, and ambiance and the acceptable values for those attributes from his knowledge of typical guests in his hotel and their previous restaurant experiences. The concierge’s recommendation will be further refined by specific information about the requesting guest’s characteristics or preferences. For example, the concierge might choose a trendier restaurant for a younger guest or a more romantic restaurant if the guest is dining with a spouse rather than with business associates.

We would not describe the concierge's knowledge of the restaurant domain as an ontology because we cannot specify how it is mentally represented and processed. But for an automated service like the smart digital assistant in encounter 5 to do as well as the concierge and obviate these refinement interactions it would need to represent knowledge about restaurants, hotels, and travelers in an ontology. This ontology would contain formal definitions for concepts (e.g. that a restaurant is a place where food is served, that pasta is a type of food that is served in Italian restaurants), how different concepts are related to each other (e.g. that bistros, diners, and snack bars are also places where food is served), and make inferences for things that are not directly known (e.g. any specific Italian restaurant probably serves pasta).

When these general assertions about the semantic relationships between different types of services are combined with facts about specific restaurants, hotels and travelers an ontology-driven travel service could make inferences to recommend a restaurant whose attributes are consistent with customer attributes to meet the traveler's unstated preferences.

If the travel ontology is broad enough to represent analogous assertions about air travel, ground transportation and other travel related concepts and computer-readable schedules are available a service system could plan an entire trip that embodied a consistent quality of service even if the traveler only provides a small part of the required information explicitly.

*Service Composition Models as Templates*

The service model manager presents the customer with an interface to a service composition model that reflects the overlapping service models and that simplifies the customer's interactions and decisions. When a customer's actions or context provides information about his intent (e.g. inquiring about a product or service, buying a product or service, using a product or service) and his organizational role (e.g., specifier, payer, user in a business or personal situation), retrieving the appropriate interface is straightforward. For example, when the customer launches his "smart digital assistant" application in encounter 5, he is explicitly invoking the relevant service composition model.. However, in many situations the customer's intentions must be inferred, and identifying the appropriate service composition model is a challenging task known in computer science as "plan recognition" or "goal recognition" (Armentano and Amandi, 2008; Carberry, 2001)

The simplification of customer interaction has two aspects. The overlap in service models that makes service composition models possible implies that the customer only needs to provide overlapping information once. (Of course, the "back end" of the service system needs to maintain the complete and redundant view of each separate service model so it can deliver the required information to each service provider). Over time, as many customers or the same customer use the same service composition model, the LEARNING SYSTEM analyzes the information residues of previous encounters and transactions to infer preferences and discover patterns. These patterns are then used by the SERVICE MODEL MANAGER to substitute information for interaction by suggesting default or predicted values for the components of service models.

The potential for substitution is directly related to the granularity of the service models involved. Granularity is a concept in data modeling that refers to the amount of detail or the number of parts in a description (Glushko and McGrath, 2005, Section 6.4.3). A more precise or granular model creates more design choices in how the information is obtained and increases the potential for substituting information for interaction.

For example, consider the information needed to plan a cross-country business trip. A travel service needs to know the origin and destination airports, travel dates, and the location and time of events in the destination city. The choice of airline, ticketing class, hotel, mode of transportation to and from the airports, restaurants, and so on might be shaped by company policy, by traveler preference, or both.

This service model is highly granular, but a travel service system might fail to exploit it. If all of the separate information components in the model are treated as an aggregate called "trip information," it is likely that a service system will collect it from the traveler in a similarly all-at-once manner. Perhaps the traveler will communicate all of the trip information in a phone call to a human travel agent, fill out a trip planning form for a human travel agent, or complete a similar comprehensive form for an online travel service. In all of these cases the traveler will be required to provide information on each trip that is redundant, like his personal preferences, or for which he is not the best source, like company travel policies.

In contrast, a travel service system could be designed to exploit the granularity in the service model and make flexible and adaptable decisions about how best to obtain the information needed to populate the model. The granularity of the model enables incremental substitution of stored or computed information for information that would otherwise be obtained through explicit customer interactions. The only interactions remaining are those needed to initiate the process of assembling the required information and then choosing among or confirm the recommendations. Furthermore, the amount of substitution of stored information for explicit requests for information from the customer increases in a gradual and incremental manner as preferences and patterns accrue over time.

This simplification of customer interaction enabled by a service composition model is shown in Figure 3, which shows a single input form to a "Trip Planning Service" that replaces the multiple input forms shown in Figure 2. The composite input form also contains default or predicted values. This hypothetical example shows how the service model manager knowledge management function and the service composition function work together to meet the specific requirements of an individual customer. Rearden Commerce's "deem@work" (www.deem.com), by integrating a suite of services that encompass travel, purchasing, expense accounting and other business services, exploits this overlap in customer knowledge and implements many aspects of service composition models to make service delivery more efficient.

<insert Figure 3 here>

## HOW THE SERVICE MODEL MANAGER AND CUSTOMER MODEL MANAGER WORK TOGETHER

The service model manager and customer model manager work together because they are in effect two sides of the same coin. The service model manager maintains models and service composition models that are identified by the LEARNING SYSTEM when services with overlapping service models repeatedly go together, and the customer model manager maintains the information about the types of customers or specific customers who typically request those services. So when the service model manager receives a request for service, it works with the customer model manager to classify the customer in a particular segment or identify him as a known individual. The service model manager then uses information from the customer model to create an instance of the service or service composition that is then “handed over” to the RECOMMENDATION SYSTEM MANAGER.

A new customer of the travel service is initially unclassified but repeated encounters and transactions enable him to be classified and instantiated with a particular customer segment model. This classification bestows on the customer a set of attributes and allows inferences about his likely preferences from the segment model. This also begins the process of turning his generic interactions in the service system to simpler yet more satisfying ones that meet his requirements with lower intensity. Different customer segments are characterized by their similar patterns of choices of service providers and service levels, and these patterns can be applied by the service model manager to eliminate choices that no one makes or to default to choices that everyone makes.

For example, most business executives travelling to midtown Manhattan use scheduled “town car” limousine services rather than taxis or hotel shuttles. It is thus unnecessary for the travel service to ask a business traveler going to New York to choose between these alternative transportation modes, especially if he has already made several similar trips. Instead, the travel service might simply ask the customer to confirm a limousine reservation timed to his airport arrival. Similarly, a customer classified as a budget traveler should not be presented with a choice that includes a scheduled limousine because a budget traveler will likely not choose that option.

## THE RECOMMENDATION SYSTEM MANAGER

The RECOMMENDATION SYSTEM MANAGER begins with a customer / service model combination with the goal of making one or more recommendations that improve the chances that the instantiated service plan satisfies all of the explicit and implicit customer requirements.“Recommendation systems” are an active research area in service marketing and computer science, so it is essential that our framework can include any kind of mechanism for adapting the content or character of a service. In our model the RECOMMENDATION SYSTEM MANAGER combines information from the Customer Model, the Service Model, and current contextual information to determine the appropriate content and channel for an interaction with the customer. The RECOMMENDATION SYSTEM MANAGER can request a suggestion from any of the available recommendation algorithms or sources that can make them, analyze and possibly combine them, and then incorporate the resulting recommendation into an adapted service delivery.

We noted in the introduction that the concepts of “recommendation” or “personalization” are not always used in a consistent manner. Recommendations are generally about content, whereas personalization can also involve changes to the interaction or dialogue structure of the service encounter. Moreover, personalization by definition relies on personal information, but only some recommendation techniques make use of a specific customer model. Researchers have devised numerous algorithms for generating recommendations that use a variety of information sources; Table 1 arranges a sample of this work using a classification scheme proposed by Wei, Shaw, and Easley (2002).

<insert Table 1 here>

"Contextual" personalization is not an explicit category in this classification, probably because recommendation algorithms typically implement context awareness by filtering recommendations after they have been generated without regard to user context (Adomavicius and Tuzhulin, 2010).

The simplest recommendations are based on popularity measures or proxy measures, like best-seller lists for books, box office receipts for movies, or restaurant ratings. Improved recommendations can be generated by sorting popularity by customer segments or by emphasizing judgments made by trusted or high-reputation sources. Personalized recommendations for a customer can be generated by choosing items that are similar to items the customer prefers, that are often associated to preferred items, or that are chosen by people who are like the customer.

A hotel concierge might use a combination of these approaches to make a recommendation, suggesting to a guest that a new restaurant is similar to one he's dined at before and adding that it has already become popular with business travelers. Similarly, automated recommendations systems typically employ a combination of algorithms, taking into account the contributions of each before making a recommendation. For example, Amazon.com makes numerous recommendations based on a customer's own previous purchases, items he has viewed but not purchased, and items similar to or typically bought in conjunction with those he has purchased or viewed.

Recommendations that propose content or content transformations are the most commonly researched and implemented, so we will not further discuss them. Instead, in the remainder of this paper we will focus on provider-initiated personalizations that change the service encounter by modifying the "interaction style" or "dialogue structure" between the service provider and the customer. In particular, we will emphasize personalizations that eliminate interactions in a service encounter or that eliminate services in a service composition model by using stored, computed or inferred information to make them unnecessary – as we saw in encounter 3, when the hotel concierge recalled the guest's previous restaurant choice, and in encounter 5, when the "smart digital assistant" app chose a restaurant that satisfied availability, location, and quality constraints with little customer interaction .

The degree to which a service encounter can be systematically simplified by removing user interactions mostly depends on the amount of the information needed to perform the service that the service provider already has. We can describe three levels of substitution that follow a pattern we call "graceful substitution." Defining these three levels of substitution is somewhat arbitrary, but they seem to capture the range of substitutions that we have observed and that has been designed into service systems. They provide a simple but useful vocabulary.

- *Remind, or Level 1 Substitution.*

  A Level 1 substitution replaces a choice that would be unconstrained in the generic service encounter with a reminder that shows the user's last choice, but with the other feasible alternatives still available as choices. In a travel service, this minimal level of substitution is taking place if the traveler is offered a list of hotels with the one he chose on his previous trip highlighted.

- *Confirm, or Level 2 Substitution*

  A Level 2 substitution is a request for confirmation in which the user's previous choice is shown, but other choices are no longer presented. In a travel service, a Level 2 substitution is taking place if the traveler is offered a scheduled limousine service as his mode of transportation from the airport to his hotel, and his decision is to accept or decline the service. Declining the offer would cause the service system to “fall back” to the Level 1 substitution in which other choices are also provided. A service system should only make a Level 2 substitution if its customer model can predict with high confidence that the customer does not want to consider other alternatives.

- *Eliminate, or Level 3 Substitution*

Finally, a Level 3 substitution is the complete elimination of any interaction with respect to a choice or touch point in the generic service encounter. In a travel service, for instance, a Level 3 substitution is taking place if the customer is simply given a confirmation number for a limousine reservation. If the customer rejects the substitution, the service system should fall back to Level 1 and offer alternatives to the rejected option.

The RECOMMENDATION SYSTEM MANAGER enables service personalization to an extent that goes well beyond micro-segmentation in which every customer of a service is treated differently. Service systems designed using this framework can personalize *combinations* of services by distinguishing different roles or contexts. For example, because a traveler’s preferences and transaction history are likely to differ between business trips and vacation trips, the personalization for the composed set of interrelated travel services will differ accordingly. In addition, the RECOMMENDATION SYSTEM MANAGER could employ different algorithms and confidence thresholds for making a recommendation depending on context; for example, because a business traveler might have less schedule flexibility than a leisure traveler, the Recommendation System would be less likely to propose opportunistic services to the former.

An intriguing unresolved issue for us is how the service quality concepts of “service failure” and “service recovery” apply in our substitution framework. Recent research has begun to examine service failures and service recovery in customer interactions with technology and self-service (Dabholkar and Spaid 2012, Dong, Evans and Zou 2008), but have not examined the effects of such failures or attempted recoveries when information has been substituted for interaction. A service system should only substitute if its customer model can predict with high confidence that the customer would want it, because a substitution that restricts or eliminates a desired customer choice is a service failure. Similarly, an unwanted personalization is a service failure.

In either case, the feedback to the service system can refine the customer model, and notifying the customer afterwards can be viewed as service recovery. On the other hand, a service system that does not substitute when it has enough information to do it successfully is squandering customer value by requiring unnecessary work, which seems like another kind of service failure.

## THE SERVICE MONITORING SYSTEM

The SERVICE MONITORING SYSTEM uses the instantiated customer model and service model (which might have been modified by the RECOMMENDATION SYSTEM MANAGER) to monitor the delivery of services and the interactions with the customer. Simple services that do not directly provide information about their state or the customer experience that results from them can only be monitored passively or indirectly through later customer feedback, as when a customer calls customer service to complain that a package did not arrive when expected. More comprehensive and integrated service systems often proactively monitor the delivery of component services and gather contextual information that may impact the ability to meet overall goals (e.g., a travel management service system is notified of a flight delay or receives a weather forecast that suggests that delays are likely). In business-to-business contexts, smart interactive services often engage in remote maintenance, diagnosis, troubleshooting or repair of products or services (Wünderlich, Wangenheim and Bitner, forthcoming).

Proactive monitoring of service execution makes some customer interactions unnecessary, saving time and effort and preventing the well-known consequences of unexpected service failures. For example, if a travel management service system receives information about a delayed flight, it can notify the traveler’s hotel and reschedule his limousine pickup and restaurant reservation, and then send him the revised service plan.

## THE LEARNING SYSTEM

Customers provide explicit feedback to service providers when they rant or rave to a customer-facing employee, when they fill out a customer satisfaction form, or when they press the “Like” button on a web page. Customers also provide implicit feedback through their behavior, as when they regularly return to a restaurant and order the same dishes each time. An attentive human service provider learns from this feedback and tunes his customer model so that his recommendations are consistent with the customer’s preferences.

Automated or computational service systems can capture and learn from both kinds of customer feedback more completely and effectively. For instance, when a search engine offers recommendations in response to a customer query, the engine increases the relevance ranking for any item the customer chooses and decreases it if the customer chooses none of the presented items, looks briefly at them but moves on, or submits a refined query. Similarly, if a particular business traveler to New York always books a limousine from the airport, the travel service system should stop offering a rental car option.

In our framework the LEARNING SYSTEM’s essential purpose is to analyze information about service encounters to maintain and refine service and customer models so that they can be effectively used by the RECOMMENDATION SYSTEM MANAGER and the SERVICE MONITORING SYSTEM. The information in customer models needs to be continuously verified and evolved to enable the service system to confidently predict customer behaviors and preferences and to enable substitution of information for interaction.

Just as the RECOMMENDATION SYSTEM MANAGER might make use of many different algorithms for generating recommendations, the LEARNING SYSTEM can use many different approaches for creating and refining customer models, recognizing patterns in customer behavior and service composition, and then further refining both kinds of models to create more specialized ones. For our present purposes it is sufficient to describe them collectively as "predictive analytics" techniques that share the goal of analyzing a huge amount of data to come up with a variable that can be measured to predict future behavior (McCue 2007; Minkara 2012). Multiple predictors can be combined to increase the confidence in predicting that the customer will behave in a particular way. In the context of this article, this means "will the customer accept the recommendation" to engage in some service or class of service or choose a particular service provider.

In addition to the information contained in a customer model, the LEARNING SYSTEM maintains the model itself by continually assessing the contribution that information component in the model makes toward the prediction of customer behavior and preferences. Just as customer preferences change over time, the predictive power of information about a customer changes too. A decade ago, for example, owning a mobile phone would have classified someone as a technologically sophisticated early adopter, but today that customer attribute has little power to predict behavior.

By analyzing the information residues of previous customer encounters and transactions, the LEARNING SYSTEM can refine and create narrower customer model types. For example, the service model manager in a travel service system might have a service composition model for "a trip to New York City" that captures the information overlap between service requests for an airline flight, a hotel booking, restaurant reservations, and other travel-related services. Over time, the LEARNING SYSTEM would identify a customer model type for business executives who travel frequently, who always fly in first or business class, who always stay in upscale hotels, and who always dine at upscale restaurants. A contrasting customer model type would likely emerge for budget travelers who travel less often, who fly economy, stay in inexpensive hotels and dine at chain restaurants. Machine learning and discriminant techniques can distinguish the clusters of choices that contrast sets of historical data like those of business and budget travelers (Witten, Frank, and Hall 2011).

The LEARNING SYSTEM performs a similar refinement and specialization for service composition models. In a travel service system the typical configuration of services and touch points might differ for different destinations, and different destinations might draw different proportions of customers from various segments. For example, population density and public transportation make the likelihood of renting a car significantly lower in New York than in Los Angeles, and the university town of Ann Arbor, Michigan attracts more professors, students, and football fans than other cities its size. After the LEARNING SYSTEM identifies patterns like these, they can be exploited as “standard offerings” of bundled services that offer greater value without the need to select every detail.

Most of our thinking has focused on the provider side of the co-creation of value in a service system. This led us to realize that the learning and substitution framework we propose has analogues to “Bayesian Learning” applications, in which semi-supervised machine learning systems ask users to label the instances that will train the classifier the fastest (e.g., (Nelson 2005) and (Chater and Oaksford 2008)). This “makes the provider a better provider” by finding useful questions that increase its predictive power and we predict that these techniques will be more widely applied in service systems. But the substitution framework also has analogues in Active Learning research in applications like intelligent tutoring systems that propose problems in an order that increases the efficiency of learning (Kruschke 2008). The analogy to service systems is that this approach “makes the customer a better customer,” akin to the notion of customer efficiency (Xue and Harker 2002, Xue, Hitt and Harker 2007), and we think service system designers can also benefit from more study of that side of the co-creation equation.

# MANAGERIAL IMPLICATIONS AND GENERALIZATION

Analyzing service systems as provider-customer interactions where information is exchanged and tracing how that information is used to co-create value suggest opportunities for personalizing services. Recognizing that interactions and information are substitutes for each other helps managers and service designers make better decisions about the investments needed to collect, store, and process information about customers and interactions.

## *Managerial Implications*

- ***Proactive use of information.*** Personalization has most often been viewed as depending on the flow of information from customers to service providers, but it is essential to see that the reverse is also true. Information about customers and previous interactions becomes a valuable resource that service providers can use proactively to substitute for customer interactions. It is also useful to characterize levels of substitution (reminding, confirming, eliminating) because these reflect the amount of information needed to perform the service that the service provider already has, highlighting where more information can make service delivery simpler and more efficient.
- ***Improving service delivery and innovation***. Analyzing services with overlapping information requirements as composite services focuses attention on the highest-value information, suggests how it can best be obtained, and identifies where substitution of information for interaction is logically possible to provide more efficient, enhanced, or new services. This does not mean that information should be substituted for interaction whenever it is logically possible. However, services that require customers to provide information that they know the provider already has can be seen as service failures, as anyone who has been frustrated by

repeated requests to provide the same insurance account numbers, medical history, or payment and delivery information well knows.

- ***Making information an institutional resource***. Information from and about customers must be treated as an institutional resource managed in an easily accessible information system, not as separate individual resources possessed by service provider employees. All service provider employees should be incentivized and supported in creating and maintaining an institutional data resource that is relevant, complete, and high quality, not just frontline employees who use the information (for example, having hotel housekeeping staff note which fruits are in the welcome basket that a guest does not eat). This also means empowering front line employees to suggest types of information that they believe would enable them to deliver better services.
- ***Incentivizing the Customer.*** While incentivizing front line employees to collect quality information is crucial, it is also necessary to incentivize the customer to provide quality information. Some customers may be hesitant to provide information due to privacy concerns, or find it too tedious to provide what is needed, or even deliberately provide incorrect information. This missing or noisy information undermines the business goal of moving from coarse customer segmentation to precise customer models that enable efficient and effective personalized service delivery. Techniques for getting customers to provide more information of higher quality include explaining the relationship between information quality and value to customers and asking the customer whenever possible to cleanse his own data ("Is this purchase a gift? Is this a business trip or personal trip?), and then demonstrating the improvements in service offerings that better data enable. The way in which personalization is branded or characterized can profoundly affect data quality and customer expectations.

Consider the contrast between a “concierge” service that personalizes offerings on behalf of service providers and a “personal assistant” one that works on behalf of the customer.

- ***Generalizability***. This overall approach applies to any information-intensive service system where the service is inherently personalizable because customers have different requirements and preferences: healthcare, education, financial planning, and consulting, for example.

## MAKING THE CASE FOR THE CONCEPTUAL MODEL

We have proposed a new way of thinking about personalization in service encounters and service systems in which “substituting information for interaction” is the core principle that encapsulates the set of possible interactions between services and their customers. The model stands as a future-oriented conceptual model, and is not meant to be a software architecture diagram or design document, nor is it a schematic diagram for any one existing service system.

We test the different proposed components of the conceptual model by reasoning about implications if we selectively remove any component that we have added to the foundation provided by simple service models and customer models. This strengthens the case that the proposed components of the model are necessary.

- A service system that lacks a CUSTOMER MODEL MANAGER cannot combine user information about a customer obtained or inferred in one service with that from other services. This results in redundant interactions with customers to obtain information and eliminates many opportunities to personalize services.
- A service system that lacks a SERVICE MODEL MANAGER cannot exploit the overlapping information requirements of service compositions, resulting in redundant interactions with customers in service encounters;

- A SERVICE MODEL MANAGER that lacks an Ontology that represents the conceptual relationships among services is limited to literal interpretation of customer inputs and cannot infer information that is not explicitly provided by customers or service models;
- A service system that has no LEARNING SYSTEM cannot analyze information about previous service encounters to refine and specialize customer and service models;
- A service system without a RECOMMENDATION SYSTEM MANAGER has no way of taking advantage of specialized customer or service models to simplify customer encounters by personalizing them;
- A service system without a SERVICE MONITORING SYSTEM cannot recognize when the context or state of an instantiated service model sets the stage for a recommendation or suggests a potential or actual service failure.

The conceptual model we propose does not imply that information use or exchanges should be substituted for person-to-person interactions whenever it is logically possible. For example, many high-end hotels strive to personalize every customer interaction, suggesting a bias toward experiential encounters. However, our model makes it clear that person-to-person interactions and relational dialogue are not essential to personalization. Indeed, some luxury hotels strive for a style of minimalist personalization, exemplified by the Ritz-Carlton Hotel Company's credo that its experience "fulfills even the unexpressed wishes and needs of our guests" (Ritz-Carlton 2011). But this is only possible because of a very sophisticated customer information system that records interactions between employees and guests and implicit guest preferences such as which fruits a guest eats from a fruit basket, so that future baskets only contain preferred fruits (Wreden 2005).

For every customer who prefers a lazy chatty conversation with a bank teller or hotel front desk clerk there is surely one who views these interactions as nuisances and who wants a minimalist information-driven experience. Similarly, Amazon’s “one click buying” that lets customers use saved payment and shipping information to avoid filling out a check-out form exploits the strong customer preference to avoid a tedious and error-prone interaction.

As such, we are definitely not proposing that substituting information for interaction is an end in itself. The goal of a service system should always be to create value that meets customer expectations both in the level of quality and in the manner by which it is produced. But the substitution concepts and mechanisms we propose highlight the range of design choices and encourage a careful evaluation of whether a touch point and interaction creates or undermines value.

## DIRECTIONS FOR FUTURE RESEARCH

Because the model we propose emerges from a review and synthesis of research from several perspectives on service system design and from numerous disciplines, many unanswered questions remain. The overarching question reflects our attempt to bridge the gap between design methods for “experience-intensive” services and “information-intensive” one. Our work suggests there are benefits from a more holistic or end-to-end design approach, and more work needs to be done to reconcile these contrasting approaches.

We have proposed a conceptual model that embodies many specific assumptions and predictions about the complicated relationships among concepts of service intensity, customer preferences, personalization, customer or provider-initiated interactions, and service provider productivity. However, there are numerous ways to refine the model. In particular, it would be novel to refine the notion of the customer model to better understand and accommodate the usefulness over time of different pieces of information. Information differs not just in persistence but also in its “decay function,” and incorporating these nuances into the learning and substitution mechanisms should increase their efficiency and accuracy, making CRM and user experience design more robust.

A second refinement of the model would enable it to respect cultural norms or constraints on the preferences for, or acceptance of substituting information for interaction during service delivery. People differ in their preferences for self-service, and they undoubtedly differ in their preferences for the degree or rate at which a repeated service interaction is simplified over time by substitution mechanisms.

A third refinement of the model would result from research that quantifies the relationship between service failure and substitution of information for interaction. Not making a substitution when it is logically possible might be perceived as a failure, as might a substitution that violates customer expectations or preferences.

Finally, the alternate characterizations of service personalization coming from a "concierge" or provider-initiated perspective vs. a "personal assistant" or customer-initiated one suggest different implementation architectures. The latter architecture has advantages in control and privacy, but foregoes the opportunity to aggregate across service providers. Can we measure this tradeoff?

## REFERENCES

Adomavicius, Gediminas., and Alexander Tuzhulin (2011). "Context-Aware Recommender Systems," in Francesco Ricci, Lior Rokach, Bracha Shapira and Paul B. Kantor (eds.), *Recommender Systems Handbook*, Springer: New York, 217-253.

Adomavicius, Gediminas., and Alexander Tuzhulin (2005). "Personalization Technologies: A Process-oriented Perspective," *Communications of the ACM*, 48(10), 83-90.

Ankolekar, Ankolekar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David Martin, Drew McDermott, Sheila A. McIlraith, Srini Narayanan, Massimo Paolucci, Terry Payne, and Katia Sycara (2002), "DAML-S: Web Service Description for the Semantic Web," *Proc. First Int'l Semantic Web Conf*. (ISWC 02).

Apte, Uday, and Richard Mason (1995), "Global Disaggregation of Information-Intensive Services," *Management Science*, 41(7), 1250-1262.

Armentano, Marcelo, and Analia Amandi (2007), "Plan recognition for interface agents: State of the art," *Artificial Intelligence Review*, 28(2), 131-162.

Benford, Steve, Gabriella Giannachi, Boriana Koleva, and Tom Rodden (2009), "From Interaction to Trajectories: Designing Coherent Journeys Through User Experiences," *Proc. of the 27th International Conference on Human Factors in Computing Systems (CHI 2009)*, 709-718.

Berry, Leonard, Lewis Carbone and Stephan Haeckel (2002), "Managing the Total Customer Experience," *Sloan Management Review*, 43 (3), 85-9.

Bitner, Mary Jo, Bernard H. Booms and Mary Stanfield Tetreault (1990), "The Service Encounter: Diagnosing Favorable and Unfavorable Incidents," *Journal of Marketing*, 54 (Jan), 71-84.

Bitner, Mary Jo, Stephen Brown, and Matthew Meuter (2000), "Technology Infusion in Service Encounters," *Journal of the Academy of Marketing Science*, 28(1), 138-149.

Brusilovsky, Peter (1999), "Adaptive and Intelligent Technologies for Web-based Education," in Christoph Rollinger and Claus-Rainer Peylo (Eds.), *Künstliche Intelligenz 4, Special Issue on Intelligent Systems and Teleteaching*, 19-25.

Brusilovsky, Peter, Alfred Kobsa, and Wolfgang Nejdl (2007), *The Adaptive Web: Methods and Strategies of Web Personalization*. Berlin, Heidelberg: Springer-Verlag.

Burke, Raymond (2010), "The Third Wave of Marketing Intelligence," in Manfred Krafft and Murali K. Mantrala (Eds.), *Retailing in the 21st Century*, Springer Berlin Heidelberg.

Campbell, Christopher, Maglio, Paul, and Mark Davis (2011), “From self-service to super-service: a resource mapping framework for co-creating value by shifting the boundary between provider and customer,” *Information Systems and E-Business Management*, 9, 173-191.

Carberry, Sandra (2001), “Techniques for plan recognition,” *User Modeling and User-Adapted Interaction*, 11(1-2), 31-48.

Casati, Fabio, Ski Ilnicki, and LiJie Jin (2000), “Adaptive and dynamic service composition in EFlow,” In *Proceedings of 12th International Conference on Advanced Information Systems Engineering(CAiSE),* Springer Verlag.

Castellano, Giovanna, Anna Fanelli, Maria Torsello, and Lakhmi Jain, (2009), “Innovations in Web Personalization,” in Castellano, Giovanna, and Fanelli, Anna Maria (Eds.), *Web Personalization in Intelligent Environments*, Springer, 1-26.

Chater, Nick, and Mike Oaksford (2008), *The Probabilistic Mind: Prospects for Bayesian cognitive science*. New York: Oxford University Press.

Cockburn, Alistair (2000), *Writing Effective Use Cases*. Addison-Wesley: Reading.

Cranor, Lorrie, (2004), “I Didn’t buy it for Myself,” in Claire-Marie Karat, Jan O. Blom, and John Karat (Eds.), *Designing Personalized Experiences in eCommerce*, Springer-Verlag: New York, 57-73.

Dabholkar, Pratibha A. and Brian I. Spaid (2012), “Service Failure and Recovery in Using Technology-based Self-service: Effects on User Attributions and Satisfaction,” *The Service Industries Journal*, 32 (9), 1415-1432.

Dey, Anind (2001), “Understanding and Using Context,” *Human-Computer Interaction Institute*. Paper 34. http://repository.cmu.edu/hcii/34

Dong, Beibei, Kenneth R. Evans, and Shaoming Zou (2008), “The Effects of customer Participation in Co-created Service Recovery,” *Journal of the Academy of Marketing Science*, *36*(Spring), 123-137.

Dubberly, Hugh and Shelley Evenson (2008), “The Experience Cycle”, *Interactions*, 15(3), 11-15.

ebXML (2001). *ebXML Business Process Specification Schema*. http://www.ebxml.org/specs/ebBPSS.pdf

Erl, Thomas (2004). *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR.

Fitzsimmons, James A., and Mona J. Fitzsimmons, (2006). *Service Management*. McGraw Hill.

Froehle, Craig, and Aleda Roth (2004), "New Measurement Scales for Evaluating Perceptions of the Technology-mediated Customer Service Experience," *Journal of Operations Management*, 22 (February), 1-21.

Glushko, Robert (2010), "Seven Contexts for Service System Design," in Maglio, Paul P., Kieliszewski, Cheryl A. & Spohrer, James C. (Eds.), *Handbook of Service Science*, New York: Springer, 219-249.

Glushko, Robert, and Tim McGrath (2005), *Document Engineering*, Boston: MIT Press.

Guarino, Nicola (1998). “Formal ontology and information systems,” In Nicola Guarino (ed.) *Formal ontology in information systems: proceedings of the first international conference (FOIS ’98)* (pp. 3-15). Amsterdam, Netherlands: IOS Press.

Gwinner, Kevin, Mary Jo Bitner, Stephen Brown, and Ajith Kumar, (2005), “Service customization through employee adaptiveness,” *Journal of Service Research*, 8(November): 131-148.

Hastie, Trevor, Robert Tibshirani, and John Friedman, (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Second Edition), NY: Springer.

Hepp, Martin (2008). “Ontologies: State of the art, business potential, and grand challenges,” in *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, Martin Hepp, Pieter De Leenheer, Aldo De Moor, and York Sure (Eds.), New York:Springer.

Hofweber, Thomas (2009). “Logic and ontology.” In E.N. Zalta (ed.) *The Stanford Encyclopedia* of *Philosophy (Spring 2009 Edition).*

Kansal, Amal, Michel Goraczko, and Feng Zhao (2007), “Building a Sensor Network of Mobile Phones,” In *Proc of IEEE 6th Int’l IPSN Conf*., Cambridge, MA,USA, Apr 2007.

Kass, Robert and Tim Finin (1988), “Modeling the user in natural language systems,” *Computational Linguistics*, 4(3), 5-22.

Kruschke, John (2008), “Bayesian approaches to associative learning: From passive to active learning,” *Learning and Behavior*, 36(3), 210-226.

Linden, Greg, Brent Smith, and Jeremy York (2003), “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Computing*, 7(1):76-80.

Massa, Paolo, and Paolo Avesani (2007), “Trust-aware recommender systems,”. In *Proceedings of the 2007 ACM conference on Recommender systems (RecSys '07)*, October,17-24.

Martin, David, Massimo Paolucci, Sheila McIlraith, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, and Katia Sycara (2007), “Bringing semantics to web services: The OWL-S approach,” *World Wide Web*, 10(3), 243-277.

McCue, Colleen (2007), *Data Mining and Predictive Analytics*, Butterworth-Heinemann.

Medjahed, Brahim, Athman Bouguettaya, and Ahmed Elmagarmid (2003), “Composing Web services on the Semantic Web,” *The VLDB Journal*, 12(4), 333-351.

Meyer, Christopher, and Andre Schwager (2007), “Understanding Customer Experience,” *Harvard Business Review*, 85(2), 116-126.

Milanovic, Nikola, and Miroslaw Malek (2004), "Current solutions for Web service composition," *Internet Computing, IEEE* , 8(6), 51- 59.

Minkara, Omer (2012), “Customer Experience Management: Using the Power of Analytics to Optmize Custtomer Delight,” Aberdeen Group. http://www.aberdeen.com/Aberdeen-Library/7460/RA-customer-experience-satisfaction.aspx

Nelson, Jonathan (2005), “Finding useful questions: On Bayesian diagnosticity, probability, impact, and information gain,” *Psychological Review*, 112 (October), 979-999.

OASIS (2006). *OASIS Universal Business Language v2.0*, http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.pdf.

O'Donovan, John, and Barry Smyth (2005), “Trust in recommender systems,”. *In Proceedings of the 10th International Conference on Intelligent User Interfaces* (IUI '05). 167-174.

Pazzani, Michael J. (1999). “A Framework for Collaborative, Content-based and Demographic Filtering,” *Artificial Intelligence Review*, 13: 393-408.

Pennock, David, Eric Horvitz, C. Lee Giles, and Steve Lawrence, (2000). “Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-based Approach,” In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 473-480.

Pruitt, John, and Jonathan Grudin (2003), "Personas: Practice and Theory", *Proceedings of the Congerence on Designing for User Experiences*, 2003.

Rao, Jinghai, and Xiaomeng Su. (2005), “A Survey of Automated Web Service Composition Methods,” In J. Cardoso and A.Sheth (Eds.), *Semantic Web Services and Web Process Composition*. Springer.

Reinartz, Werner, and Vijay Kumar. (2000), “On the profitability of long-life customers in a noncontractual setting,” *Journal of Marketing*, 64(October): 17–35.

Rich, Elaine (1979), “User modeling via stereotypes,” *Cognitive Science*, 3(4), 329-354.

Ritz-Carlton Hotel Company (2011), The Credo, (http://corporate.ritzcarlton.com).

Roxin, A., Gaber, J., Wack, M., and A. Nait-Sidi-Moh, (2007), "Survey of Wireless Geolocation Techniques," *Globecom Workshops*, 2007.

Shafer, J. Ben, Joseph Konstan, and John Riedl (2001), “E-Commerce Recommendation Applications,” *Data Mining and Knowledge Discovery*, 5(1), 115-153.

Shen, Xuehua, Bin Tan, and Cheng Xiang Zhai (2005), “Implicit user modeling for personalized search,” *Proceedings of the 14th ACM international conference on Information and knowledge management*, 824-831.

Shostack, G. Lynn (1985), “Planning the Service Encounter,” in *The Service Encounter*, John Czepiel, Michael R. Solomon, and Carol F. Surprenant, eds., Lexington, MA: Lexington Books, 243-253.

Taylor, James (2012). *Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analysis*, IBM Press.

Taylor, James, and Neil Raden (2007), *Smart Enough Systems: How to Deliver Competitive Advantage by Automating Hidden Decisions*. Prentice Hall.

Taylor, Nicholas K., Patrick Robertson, Babak A. Farshchian, Kevin Doolin, Ioanna G Roussaki, Liam Marshall, Robert Mullins, Steffen Drüsedow, and Kajetan Dolinar, (2011), "Pervasive Computing in Daidalos," *Pervasive Computing*, IEEE , 10(1), 74-81.

Teboul, James (2006). *Service is Front Stage: Positioning Services for Value Advantage*. Insead.

Teichert, Thorsten, Edlira Shehu, and Iwan von Wartburg (2008), "Customer segmentation revisited: The case of the airline industry," *Transportation Research Part A: Policy and Practice*, 42(1), 227-242.

Towle, Brendon and Clark Quinn, (2000), "Knowledge Based Recommender Systems Using Explicit User Models", In *Knowledge-Based Electronic Markets*, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04. 74-77. Menlo Park, CA: AAAI Press.

Tsiptsis, Konstantinos, and Antonios Chorianopolous (2009), *Data Mining Techniques in CRM: Inside Customer Segmentation*. Chichester, UK: John Wiley & Sons, Ltd.

Verhoef, Peter, Katherine Lemon, A. Parasuraman, Anne Roggeveen, Michael Tsiros and Leonard Schlesinger (2009), "Customer Experience Creation: Determinants, Dynamics and Management Strategies," *Journal of Retailing*, 85 (January), 31-41.

Verhoef, Peter, Rajkumar Venkatesan, Leigh McAlister, Edward Malthouse, Manfred Krafft and Shankar Ganesan (2010), "CRM in data-rich multichannel retailing environments: A review and future research directions,". *Journal of Interactive Marketing*, 24(2), 121–137.

Wei, Chih-Ping, Michael Shaw, and Robert Easley (2002), "A Survey of Recommendation Systems in Electronic Commerce," in Rust, R, and P.K. Kannan (Eds.), *E-Service: New Directions in Theory and Practice*, 168-199.

Wiederhold, Gio (1997), "Customer Models for Effective Presentation of Information"; In Jim Flanagan, Tom Huang, Patricia Jones, Simon Kasif (eds): *Human-Centered Systems: Information, Interactivity, and Intelligence*, National Science Foundation, 218-221.

Witten, Ian, Elbe Frank, and Mark Hall (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd Edition). Burlington, MA: Morgan Kauffman.

Wreden, N. (2005) *Profitbrand: How to increase the Profitability, Accountability and Sustainability of Brands*, Kogan Page.

Wünderlich, Nancy V., Florian Van Wangenheim, and Mary Jo Bitner (2012), "High Tech and High Touch: A Framework for Understanding User Attitudes and Behaviors Related to Smart Interactive Services," *Journal of Service Research*, forthcoming.

Xue, Mei and Patrick T. Harker (2002), "Customer Efficiency: Concept and Its Impact on E-Business Management," *Journal of Service Research*, *4*(May), 253.

Xue, Mei, Lorin M. Hitt and Patrick T. Harker (2007), "Customer Efficiency, Channel Usage, and Firm Performance in Retail Banking," *Manufacturing & Service Operations Management*, *9*(4), 535-558.

Zomerdijk, Leonieke, and Christopher Voss (2010), "Service Design for Experience-Centric Services," *Journal of Service Research*, 13(1), 67-82.

FIGURE 1

A Conceptual Model for Personalization

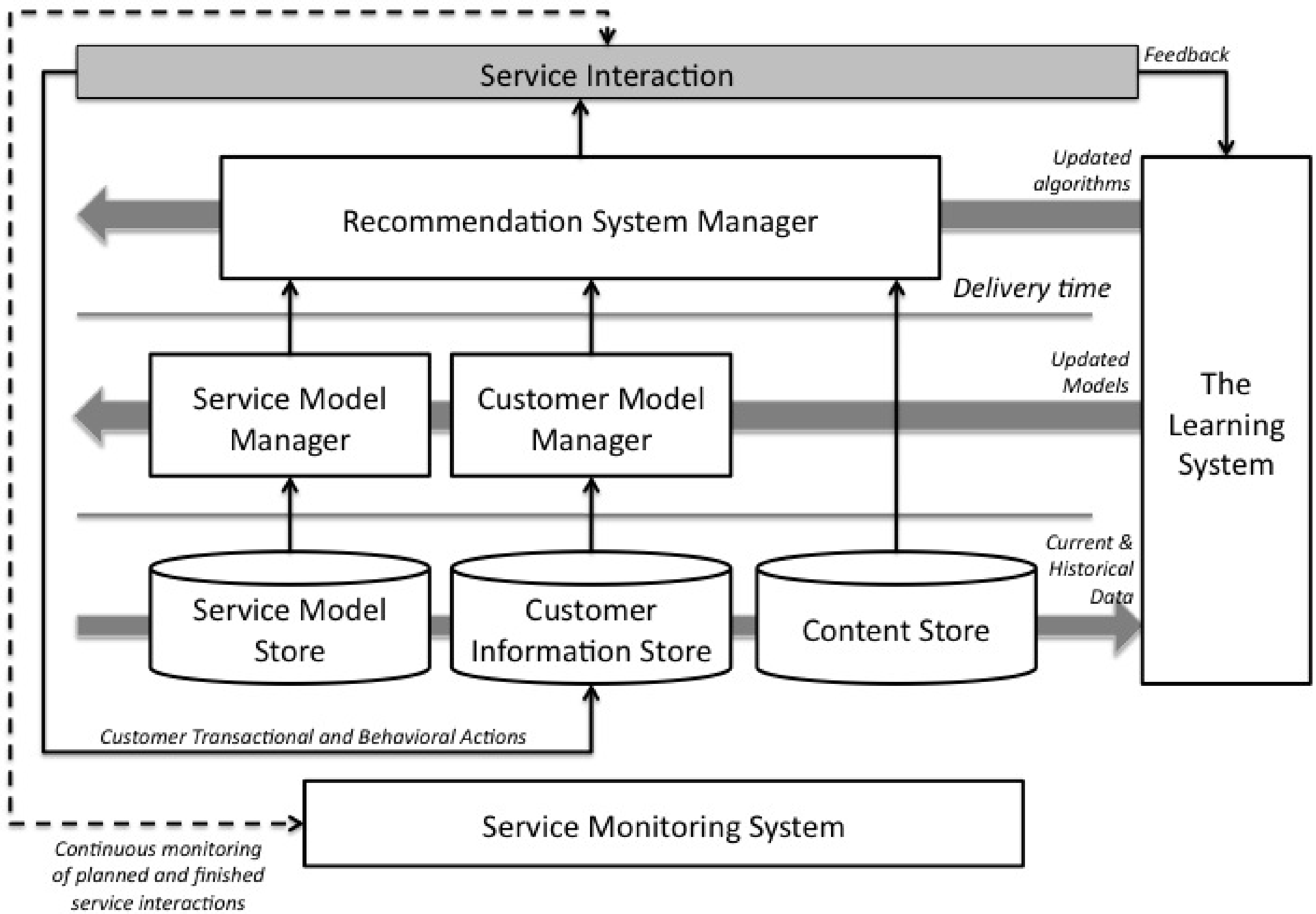

FIGURE 2

Overlapping Service Models from Four Travel-Related Websites

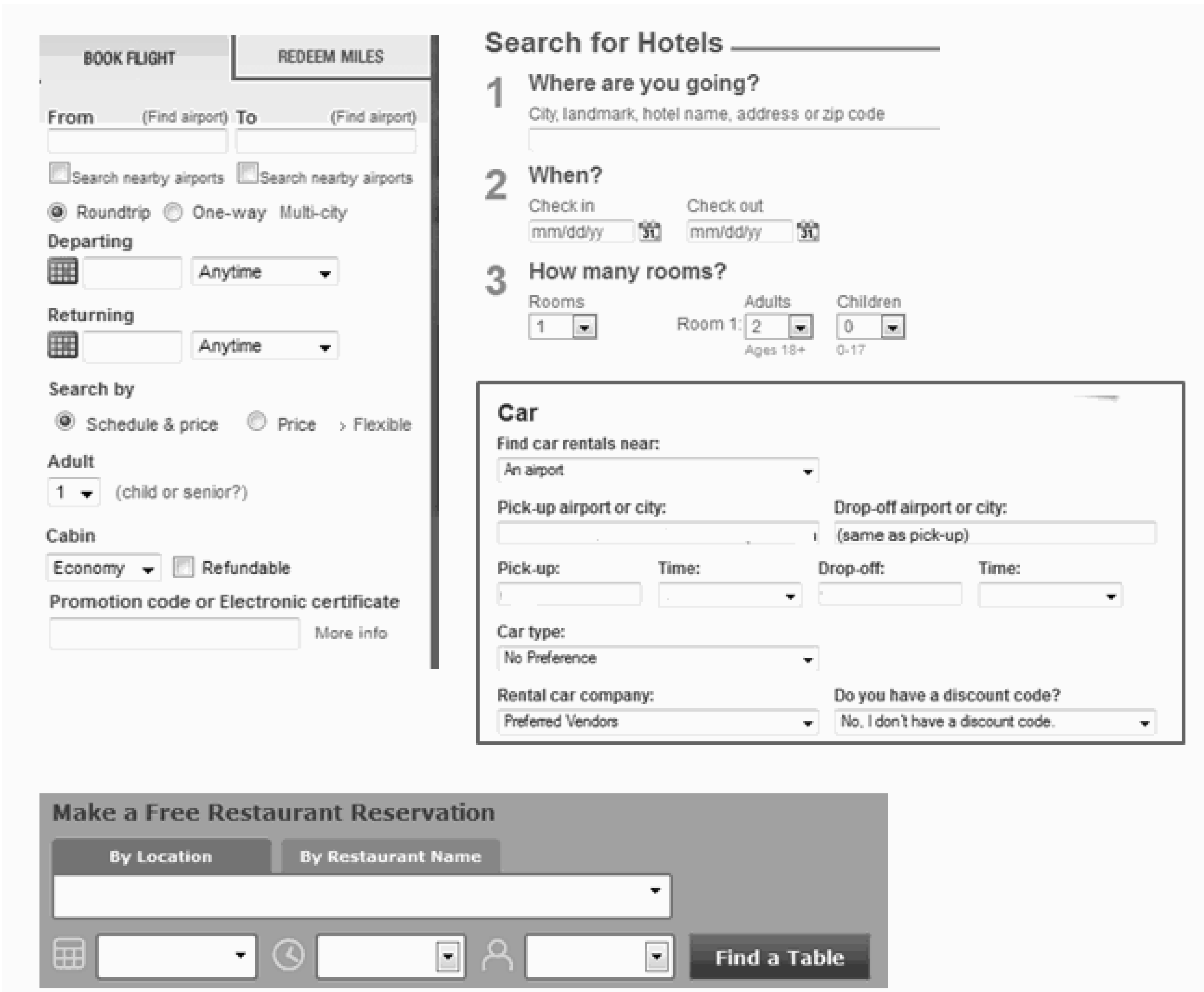

FIGURE 3

Input form for a “Trip Planning Service” whose Service Composition Model captures the overlap in the four separate service models in Figure 2. The composite input form contains default or predicted values.

TABLE 1

Classification of Recommendation Systems

after Wei, Shaw, and Easley (2002)

| RECOMMENDATION APPROACH | INFORMAL EXPLANATION | REPRESENTATIVE PAPERS |
|---|---|---|
| Content Filtering | What items are similar to items that I have liked? | Cranor (2004); Pazzani (1999) |
| Collaborative Filtering | Who is like me and what do they like? | Adomavicius & Tuzhulin (2005); Linden, Smith, & York (2003); Pennock et al (2000) |
| Reputation / Trust-based Filtering | Who are the people / groups / entities I trust and what do they like? | Massa & Avesani (2007) O'Donovan & Smyth (2005) |
| Popularity-based | What are the items that are most liked by the general population? | |
| Demographics-based | What are the items that are most liked by people in the same demographics group as I am? | Pazzani (1999); Towle and Quinn (2000) |