A Service Science Perspective on Human-Computer Interface Issues of Online Service Applications

Claudio Pinhanez, IBM T.J. Watson Research Center, USA

ABSTRACT

150 words or less.

Keywords: 7 keywords or less

INTRODUCTION

In 1991, *Scientific American* published an extraordinary collection of essays about the upcoming era of integration of communications, computers, and networks (Dertouzos, 1991b). The issue included articles from technology visionaries such as Michael Dertouzos, Vinton Cerf, Nicholas Negroponte, Alan Kay, Mitchell Kapor, and then US Senator Al Gore. Among other things, the articles predicted the appearance of large scale broadband networks, the non-centralized structure of today's WWW, the ubiquity of e-mail, the telecommuting phenomenon, and the emergence of India as a software outsourcing powerhouse, as well as problems such as junk mail, cyber-crime, and identity theft.

But notably, all authors failed to predict that the massive interconnection of users in cyberspace would open the space for large online service providers that could mediate the relationship between users and the vast amount of data. Among others, online service providers that collect, analyze, negotiate, process, filter public and private data, and provide simplified information access as services such as *Google*, *Yahoo*, and *Mapquest;* e-retailers such as *Travelocity*, *Amazon*, and others; and service providers based on social networking such as *eBay, Skype, Facebook, Orkut*, etc.

The common thinking 15 years ago seemed to be that the access to the myriad of computers in the network and the browsing and filtering of their data, as well as the bulk of the interpersonal connections, would be performed by personal tools or software agents that would scout and explore the Internet for information relevant to their users. A good exemplar of this view is the concept of *knowledge robots*, or *knowbots*, proposed by Robert Kahn and Vinton Cerf (Kahn & Cerf, 1988), *"…programs designed by their users to travel through a network, inspecting and understanding similar kinds of information…"* as described in (Dertouzos, 1991a, pg. 35). Knowbots were to be unleashed to fulfill specific user requests for information, moving *"…from machine to machine, possibly cloning themselves […] dispatched to do our binding in a global landscape of networked computing and information resources."* (Cerf, 1991, pg. 44).

The problem with the agent-based vision of information search is that it does not scale up. In the current world of distributed information, this approach to information search would require each of us to run (and possibly store) the equivalent of *Google*'s operations of crawling the web, indexing, and search matching. What the authors of the *Scientific American* issue could not see is that, as networks and their users grow well beyond the academic, mostly engineering-minded users of the Internet in the early 90s, there are tremendous economies of scale when millions and millions of queries are handled by a central system that crawls and indexes all the information available (independent of specific queries) and provides information finding as an *online service application*.

But how different are such online service applications from traditional software applications? The goal of this article is to describe a framework for online service applications that differentiates them from traditional interactive software tools, so it can be used to explain and predict the differences between the two, particularly in issues related to the design of human-computer interface for online services. As a consequence software tools and online service applications are intrinsically different, even when used for similar tasks, and should be designed and engineered differently. There is, of course, an extensive body of practice and empirical knowledge about developing interfaces for online interactive applications — exemplified by all the knowledge built in the last decade and half about web applications, as, for example, described in (Nielsen, 2000). Also, there has been work examining HCI and usability issues in e-commerce (C.-M. Karat, Blom, & Karat, 2004; Nah & Davis, 2002; Voss, 2003), but we believe that these works suffer from not having an appropriate theoretical understanding of what an online service is and, therefore, miss an important part of the picture when reasoning about their findings.

We start by noticing that four assumptions base the understanding of how a traditional software tool is supposed to work: (1) a single user, (2) who controls the machine and data being used, (3) inputting data (4) to be automatically processed by the machine in order to produce some desired information output. In contrast, in online service applications, the usual architecture comprises a network of computer-based systems where (1) multiple, unrelated users, (2) who do not own or control the server machines and/or most of the data being used, (3) provide personal data or assets as input to an online provider (4) to receive some output, in the form of information or not, delivered automatically or with human assistance. For example, in *Google Web Search*, (1) thousands and thousands of people, virtually simultaneously, (2) who do not own or control *Google*'s machines or data, (3) type everything from trivial questions to their most intimate desires (4) to receive a list of web-links as determined by a mostly-automated process.

We take in this article a theoretical approach where we try to investigate such online service applications with concepts, ideas, and analytical tools from *Service Science*, which has been developed in the last 40 years mostly by researchers in business and management schools. (Fitzsimmons & Fitzsimmons, 2004; Lovelock & Wirtz, 2004; Zeithaml, Bitner, & Gremler, 2006) are good examples of textbooks in the area. This effort has been recently joined by computer scientists and engineers through the *SSME* initiative *(Service Science, Management, and Engineering)*, as detailed in (Spohrer, Maglio, Bailey, & Gruhl, 2007; Spohrer & Riecken, 2006).

However, to correctly apply concepts from Service Science to online service applications, we have to carefully identify which online applications have characteristics similar to traditional services. We start this process by agreeing with (Pinhanez, 2008) that one of the main differences between online applications and personal tools is that in the latter the user controls the means of production: when and how intensively to use it, where the information (often personal) is stored and who can access it, how much effort is put on a given task or goal, and even if the user is entitled and allowed to use the application. In other words, in the case of online service applications there is a high level of *user dependency* on the provider(s) of the means of production. This requires the online interface to deal with issues not usually even contemplated by a tool application, such as: trust creation and maintenance; privacy concerns when handling sensitive information; communication of user context; hard to predict interface response times due to fluctuations in demand; and many others.

Following, we argue that not all online applications are service applications, but that the online applications can be subdivided along the dimension of how fundamental user input is for the production process, or, borrowing from theoretical work on distinguishing services from goods by Sampson (Sampson & Froehle, 2006), how much *user intensity* they require,. It is important to distinguish between user intensity and personalization: the former quantifies the user input to the production process of the online application, while the latter refers to how to adapt the production process based on user's preference and information. Based on the level of *user intensity*, we draw a clear line between *online services providers* (*Google Web Search*, *Travelocity*) and *online information providers* (*cnn.com, Google finance*), similar to Pinhanez's concept of *customer-intensive systems* (Pinhanez, 2008).

We have been investigating how this framework can inform all aspects of online service applications, including design, development, testing, delivery, and evaluation. However, due to limitations in space, in this article we illustrate our approach by addressing only human-computer interface. We then how the Service Science framework reveals more clearly where and how interface design and evaluation is affected by traditional service ideas and how online service interfaces have an additional role as enablers of relationships between customers and service providers. Although there has been some work examining traditional Service Science concepts and how they apply in their counterparts in the online world — such as Ryan and Valverde's study on waiting in line effects on consumer behavior (Ryan & Valverde, 2006), and many works on service quality measurement of online services (Barnes & Vidgen, 2000; Loiacono, Watson, & Goodhue, 2007; Parasuraman, Zeithaml, & Malhotra, 2005; Szymanski & Hiseb, 2000; Webb & Webb, 2004; Wolfinbarger & Gilly, 2003; Yoo & Donthu, 2001) — our approach here is to point out a much large number of candidate areas for future research. We conclude this article discussing issues related to the practical validation of this framework and examining future and possible developments of our ideas.

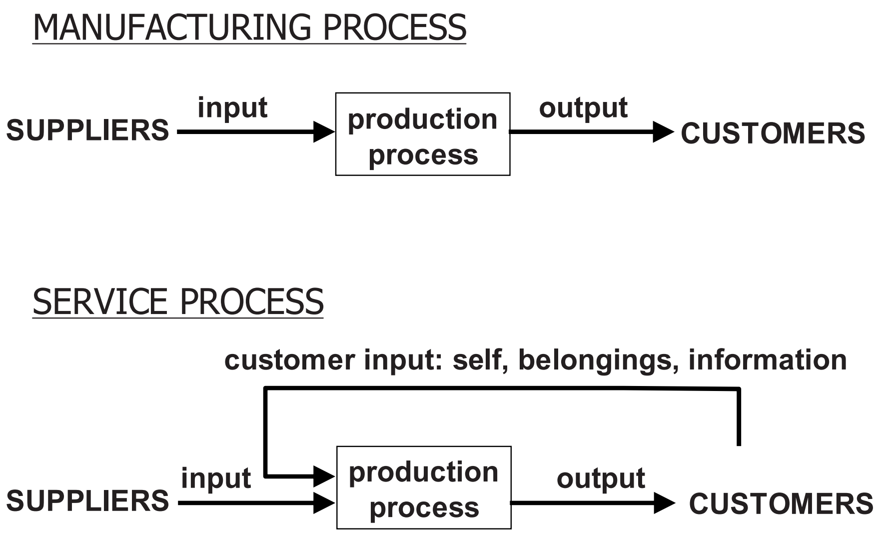
It is important to point out that there is an enormous amount of literature and practical knowledge on how to architect and develop *computer services* for online applications. The extensive work on *Service Oriented Architecture (SOA)*, described, for instance, in (Bieberstein, Sanjay, Fiammante, Jones, & Shah, 2006; Ferguson & Stockton, 2005) typifies this recent focus on understanding how computers can provide services. However, computer services, in the way addressed by SOA, are not to be consumed directly by human beings, but instead by computer applications. This feature changes, in our view, the essential nature of computer services dramatically, and makes the methods and techniques useful for their design and development radically different from the ones used on online services for humans. For instance, while establishing long-term relationships with their human customers is a key issue for online services, the key issue for computer services is likely more on the line of standardization. By constraining our discussion to online services applications as providers of services for people, we apply concepts and methods from Service Science which is traditionally focused on human customers.

A SPACE OF INFORMATION APPLICATIONS

The core of our framework relies on some recent work by Pinhanez (Pinhanez, 2008), building on some concepts from Sampson (Sampson, 2001), popularized in (Sampson & Froehle, 2006). According to Sampson's work, a necessary and sufficient condition for a production process to be a *service process* is that *"[...] the customer provides significant inputs into the production process."* (Sampson & Froehle, 2006, pg 331). This primacy of customer input is put in contrast to manufacturing processes, where *"groups of customers may contribute ideas to the design of the product, but individual customers' only participation is to select and consume the output."* (Sampson, 2001, pg 16). Figure 1 shows two diagrams depicting graphically the main differences between manufacturing and service processes.

Notice that it is implicit in this definition — and discussed at length in (Sampson & Froehle, 2006) — that customers and service producer are separate entities. As proposed by Pinhanez (Pinhanez, 2008), a better way to make this distinction is to say that the customer does not control most of the means of production. (Pinhanez, 2008) then proposes the concept of *customer-intensive systems*, which comprises systems with high user intensity and user dependency and shows that not all services are customer-intensive.

Figure 1. Manufacturing and service processes according to Sampson's theory (Sampson, 2001)

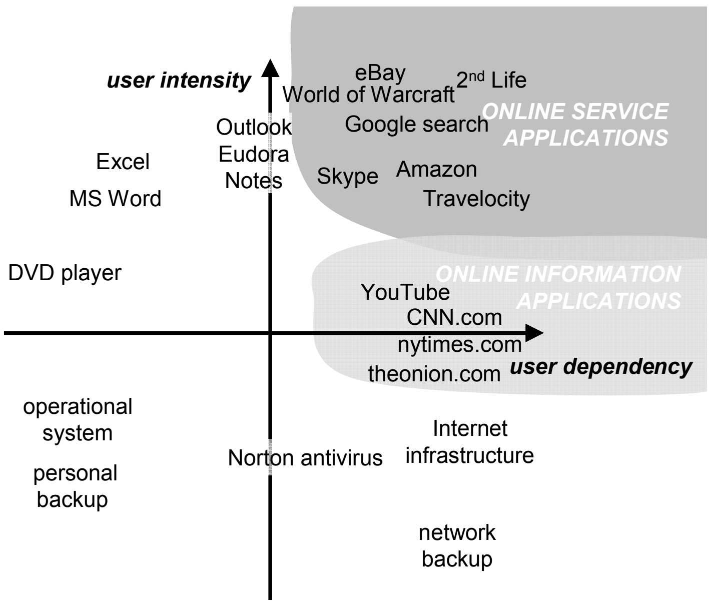


Our framework uses Pinhanez's concept and applies it to online applications. We start by taking the two key concepts, *user input intensity* and *user dependency*, and consider them as two dimensions in the space of information applications as depicted in Figure 2.

The first dimension relates to **how much the user controls the means of production** (the horizontal axis of Figure 2), or the level of ***user dependency*** on external providers. An application installed and able to run in a personal computer without network access such as traditional word processors (*MS Word*, LaTeX) is a typical example where the user controls most of the means of production. Other examples involve the basic core functions of the operational system, database applications using data stored in the user's machine, personal back-up systems, etc. As the user loses more and more control of the means of production, the closer the application gets to typical online applications such as web search and browsing, information and news provision, web retail, multiplayer online games, etc. Typical middle-point applications are e-mail applications such as *Eudora*, *Notes*, or *MS Outlook* which combine local processing and storage with extensive processing and data exchange with external servers; and local applications that depend on constant external updates such as most of today's anti-virus programs.

The second dimension tries to characterize **how much the user is an essential part of the input to the production process** (the vertical axis of Figure 2), or the level of user input ***intensity***, or simply, ***user intensity***. As described by Sampson in (Sampson, 2001; Sampson & Froehle, 2006), the user can be the input to the production process in different forms: as herself (body or mind) such as when the services of a doctor in a hospital are used; as her belongings, such as when the user's car is taken to a repair shop; or her information, as when giving financial information to get a loan from a bank.

Figure 2. The space of information applications as defined by two dimensions: user dependency on external providers and user input intensity. The positions of the applications shown here are illustrative and do not reflect specific coordinates.



Notice that in all cases, the production process is unable to even start until the user provides her input. Typical information applications which require the user to be significant part of the input to the production process are most of the interactive software we use in our everyday work such as word processors, spreadsheets, e-mail clients, computer games, etc.; and many of the web applications available today.

However, not all web applications require the user to be an essential part of the input to the production process as pointed by (Pinhanez, 2008). Typical cases are online information providers such as *cnn.com, nytimes.com,* or *theonion.com*. Although the delivery of particular pieces of information or entertainment is trigguered by user input, a large part of the production process of the information is performed without any input from the user, through the manufacturing-like processes of news gathering and filtering, and entertainment production. Although the delivery of the information is interactive, the production of content is performed as free of user input as when cars are manufactured in an assembly line. Of course, *nytimes.com* is more dependent of user input than the *The New York Times* newspaper, but it clearly has a production process less dependent on user input than online services such as *Google Web Search*, *Travelocity*, or *Amazon*.

By taking these two dimensions spawning the space of information applications, we can draw the chart shown in Figure 2 that depicts different information applications as function of the level of user dependency, with the user being more dependent on external providers as we move from the left to right; and the level of user input intensity to the production process, with increasing user intensity from bottom to top. To illustrate our argument, we plotted some typical information applications and systems on this chart on approximate positions. No metric for the two dimensions has been precisely defined, so the chart in Figure 2 should be regarded more like a topological map showing only the relative displacement of typical applications and services.

ONLINE SERVICE APPLICATIONS

One way to understand the chart in Figure 2 is to consider the top half as the space of interactive (i.e., user intensive) applications; and the right half as the space of online (i.e. user dependent) applications. The top-right quadrant can then be seen as the one of online interactive applications which encompasses most interactive web applications. We observe here that not all the applications in this quadrant have enough similarity to traditional services, but only the top-right part of this quadrant where the user becomes a significant part of the input to the production process. As noticed before, traditional news and entertainment providers have a production process more similar to manufacturing than to services.

To take in account such issues, we propose the following definition. An *online service application*, or simply an *online service*, is an application where:

1. The user does not control most means of production.
2. The user (self, belongings, information) is a significant part of the input to the production process.

Part (1) of our definition states that the user does not control the basic factors of production — resources, capital, and labor — and therefore cannot determine when and how intensively resources are used: where her information is stored and who can access it; how much effort is put on a given task or goal; and what the price of the service is and how it changes through time. This definition contrasts with traditional tool applications, which tend to assume that the user communication with her data, other databases, the World Wide Web, or other users is not mediated through a service provider. To highlight the service vs. tool difference, we use the term *customer* instead of *user* whenever we are referring to a service application. We acknowledge here the importance of the discussion led by Don Norman (Norman, 2006) about the possible drawback of depersonalizing people by using the terms "customer" and "user", but we use the terms nonetheless because the distinction significantly contributes to highlight the difference contexts of online applications.

Part (2) of the online service application definition tries to differentiate between manufacturing and service production processes. The goal of this differentiation is to assure that we only apply service-related concepts and methods to online applications that actually behave as services. We argue here that even interactive online information providers such as *cnn.com* have characteristics closer to manufacturing systems than to services and therefore are not likely to be beneficiaries of traditional services ideas and methods. Notice that the distinction hangs a lot on the interpretation given to the term "significant part" of item (2) of our definition. We acknowledge this to be a possible source of future problems, but we consider premature at this point to establish a clearer metric to completely differentiate online service providers from information providers. It suffices to say that for the scope of this article, a commonsensical interpretation of the expression "significant part of the input" does not seem to create significant theoretical problems.

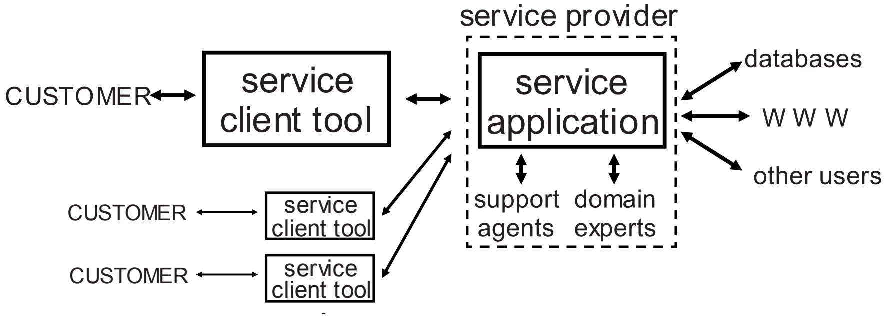
Having defined what we mean by an online service application, let us discuss typical characteristics of them.

Figure 3 shows a typical architecture of an online service application. For example, a commonly found characteristic of an online service application is that the service provider has **a multitude of customers**, as shown diagrammatically in

Figure 3. A service provider can exploit economies of scale to offer information provision at levels and cost that would be unfeasible in a tool-based approach.

Having a large number of users to share the costs also enables online service providers to employ more frequently expensive **human domain experts and contact agents** in hybrid production systems to accomplish tasks that are beyond today's computing abilities, such as situations involving common sense reasoning or ethical judgement. Also it allows the shared use of human support agents to handle situations too atypical to merit the construction of a dedicated piece of software or interface, or when there is a need of human contact, for example, to evaluate and address a customer complaint, or to detect and handle cases of customer negative or illegal behavior. This is hardly a possible solution in the context of tool software, where the user's machine has to automatically solve every task. We think that such human-machine hybrid architectures, possible mostly in the context of online services, are in fact a liberating idea for software engineering, traditionally submitted to the chains of full and complete automation, as more extensively dis-

Figure 3. Typical architecture of an online service application



cussed in the context of ubiquitous computing applications in (Pinhanez, 2007).

Finally, economies of scale are just one of the advantages of online service applications. An online service provider can also take the part of a **trusted and impartial intermediary** between two or more customers, enabling environments suitable to the establishment of relationships. For instance, consider *eBay* and how it mediates buyers and sellers in its auction environment. Similarly, **information about multiple users can be aggregated** and used to establish "cast of thousands" data handling methods such as collaborative filtering (Shardanand & Maes, 1995) and social matching (Terveen & McDonald, 2005).

HCI OF ONLINE SERVICE APPLICATIONS

There has been very little theoretical work in terms of establishing a framework to understand online applications that take the form of services in the way defined above and what is specific about how to architect, design, engineer, evaluate, deploy, and manage them. We are currently taking this approach in our research work, and so far our best insights have been related in the context of the design of the human-computer interface of online services.

In the HCI domain, most of the discussion about the design and evaluation of interfaces for online service applications tends to consider the broader class of online interactive applications — for instance, (Nielsen, 2000) — or the more restrictive class of online retailers (Loiacono et al., 2007; Nah & Davis, 2002; Voss, 2003). Our approach has to been to assure by construction, that our category of online service applications are services, at least according to Pinhanez's and Sampson's service theoretical frameworks (Pinhanez, 2008; Sampson, 2001), and to apply well-know concepts of Service Science in the context of HCI of online services.

Of course, there are many competing theories about how to characterize and classify services, as discussed, for example, in (Sampson & Froehle, 2006), especially in comparison to manufacturing. It is quite beyond the scope and need of this article to digress on the different views and to discuss how they may affect the HCI of online services.

Software production has traditionally been considered by economists in the services category of businesses. However, a more careful analysis under the light of the discussion of the previous sections reveals that many of the production processes for software in fact resemble more manufacturing than services. This is especially true for shrink-wrap software, such as popular tools like *Microsoft Office*, *Intuit Quicken*, *Adobe Photoshop*, as well as basic and middleware software such as operating systems and database programs.

Although there is user input during the design and implementation process of software, often from focus groups, individual user needs have hardly any impact on software development. Looking back into Figure 1, shrink-wrap software has traditionally followed production processes that resemble more the manufacturing of physical goods than typical service production processes. We believe that traditional HCI has thus been biased towards this model where tools are created for generic users to support a range of typical tasks. In terms of HCI evaluation and usability issues, a lot of effort is traditionally put in determining the typical individual usage scenarios of the tool and then to recreate in the laboratory meaningful test procedures. The different dynamics of web applications has required HCI practitioners to change their techniques to reflect some of the special needs of online applications as, for example, in the excellent handbook by Nielsen (Nielsen, 2000). Taking one step further, we believe that recognizing the specific characteristics of online service applications in contrast to generic online applications allows the development of more appropriate design and evaluation tools that also take in consideration the user dependency and the user intensity aspects.

We structured our discussion by considering the basic characteristics of services. We compiled and fused service characteristics listed by different authors (Fitzsimmons & Fitzsimmons, 2004; Lovelock & Wirtz, 2004; Sampson, 2001; Zeithaml et al., 2006), arriving to a "compromise" list which we believe most of them would agree to: *customer-as-input*, *heterogeneity*, *simultaneity*, *perishability*, *co-production*, and *intangibility*. We have looked into these six characteristics and examined how concepts and techniques from Service Science related to each of them highlight some issues which are very relevant to the HCI design of online services. The result is the list of 15 important issues for the design of online service applications described below, organized by the prevailing service characteristic that best explain them.

Customer-as-Input Issues

Customer-as-input refers, as discussed before, to the fact that in services the production process significantly uses inputs from the customer, such as his body, his belongings, and/or his data. We believe an immediate consequence of customers' information as input in the context of an online service application is that trust, privacy, and security and authentication issues become key and strategic for the interface design.

Unlike in traditional tool software where the privacy of data is often taken for granted and trust on the tool is often assumed to be unlimited, dealing with a online service provider always involves an exchange of **trust** between the parties. Customers often entrust online service providers with very sensitive information about themselves, their health, their finances, their loved ones, even their most intimate desires. The HCI research community has looked into issues related to trust in many different ways. As pointed by Wang and Emurian's overview (Wang & Emurian, 2005) most research suggests that trust in online applications is a function of *"... a framework of trust-inducing interface design features, [...] namely (1) graphic design, (2) structure design, (3) content design, and (4) social-cue design."* (Wang & Emurian, 2005, pg. 21). A study on websites creditbility by Fogg et al. (Fogg et al., 2001), as well as Brodie et al. study of e-commerce environments (Brodie, Karat, & Karat, 2004), share similar recommendations, also present in a well-known set of design guidelines for online experiences by Shneiderman (Shneiderman, 2000). (Featherman, Valacich, & Wells, 2006) tested similar hypothesises and found them true in their experiments, although other factors seem to influence the perceived risk of an online e-payment service, including the computational literacy of the customer and the generic class of the online service.

When we look into traditional services knowledge and practice, the focus of techniques for building trust often focus not only on front-end issues but also in making the back-end workings of services more "transparent" and visible to the customers. For example, a restaurant can improve the trust of their patrons in its cleanliness and service by having large windows to the kitchen area; or, in an online example, a shipping service may provide detailed real-time package tracking information (as most of them do now). The difference is paramount: instead of asking for trust by improving the form of the interface, the service provider elicites trust by making its internal workings more visible: "trust what I do" instead of "trust what I say". Interestingly, increasing the visibility of back-office operations in traditional services often improves also service quality, mostly due to the added pressure on the staff (Heskett, Sasser, & Schlesinger, 1997, pg. 160), but also by empowering the customer and transforming her into a quality inspector.

A services technique/tool that is often used to help identify the best candidates in the service process to be made visible to customers is a *service blueprint*, originally proposed by Shostack (Shostack, 1984) and further developed in (Zeithaml et al., 2006, pgs. 267-276). It is a map that portrays the service system, showing the whole process of service delivery, where customer contact happens, the roles of employees and customers, the visible elements of the service, and the overal flow of information. In particular, service blueprints depict clearly what is and is not visible to customers in a service process, by separating service components above and below a line called the *line of visibility*.

Dealing with *privacy* of information is also an issue that becomes fundamental in online services. Unlike in online information providers, private user information is often an essential part of the input to online services, for example, when applying to a bank loan. Traditional services often relied on the employees' judgment to decide which information to ask a customer, which part to actually record, and to decide the trustworthiness of the information provided. Also, often the privacy guarantees were part of the human relationship between customer and employee. Unfortunately designers are still trying to find ways to translate this human-based kind of privacy management to the online world. In the meantime, a general guideline is that, when an online service application asks for information that is particularly sensitive, the interface should clearly inform the customer why the system needs it, what the privacy policy is, for how long it will be kept, and whether there are alternatives to provide that particular information. Marking clearly which elements of personal information are mandatory and which are optional is a good practice often employed by websites. Notice that handling of private data issues are likely to be increasingly important given how common *phishing* attacks have become (Dhamija, Tygar, & Hearst, 2006).

It is interesting that many online services still resort to the long, legalese-full, license-agreement style of defining their privacy and data handling policies that are reminiscent of the never-read shrink-wrap software licenses. There have been efforts to simplify the establishment and negotiation between the customer and the service provider about privacy handling issues, for example in the *SPARCLE* project where privacy statements are machine-translated and automatically matched to customer-defined privacy standards (J. Karat, Karat, Brodie, & Feng, 2005).

Security is also a key issue for online service applications. There is a bias in computer science to look into security issues from a cryptology perspective, that is, by establishing complex mathematical mechanisms of encryption of information. Services, and in particular, sales, have found through the years that one of the most effective ways to provide a sense for security for their customers is through guarantees of satisfaction, such as return policies, and your-money-back, no-questions-asked mechanisms. Also, security is a two-way problem: the service provider also has to impose mechanisms to guarantee the payment for its services, often walking the thin line of not being perceived as distrusting of the customers.

Heterogeneity Issues

Heterogeneity is used in Service Science to address to the fact that in services customers tend to be very unique in their identity and requirements, so the execution of a service production process is usually highly tailored and quite unique to a customer request and input. One of the issues brought be heterogeneity is the need of *personalization*, which, unlike most of the issues discussed here, has been in fact extensively studied in HCI. In the HCI literature, personalization refers to the use of user-specific information to tailor the interaction process (J. Karat, Karat, & Brodie, 2004), often through the use of some sort of reasoning on top of a user model. For example, (C.-M. Karat et al., 2004) compiles several studies about personalizing e-commerce experiences.

But personalization of services is only one of the issues brought out by heterogeneity of input. Even if an online service does not allow interface personalization (such as *Google Search*), it still has to handle a high level of heterogeneity in its input, simply because people's lives, needs, and desires differ substantially, defying standardization at every corner. The effect of having highly heterogeneous input in the service production process is a traditional concern of Service Science. A key issue arising from heterogeneity of input is that it requires the service designer to consider all possible instances of the input and how to handle all specific cases, including those instances where the service is not delivered successfully. It is interesting how traditional HCI research rarely tries to understand how to handle tasks which are not achievable, or even how to inform the user about the limitations of a tool. In contrast, *service recovery*, or how to handle unsuccessful delivery of services, is a major theme of research in Service Science, given its known impact in service quality and customer loyalty. (Zeithaml et al., 2006, chapter 8) provides a good introduction to service recovery and to techniques used to alleviate the impact of failed delivery on customer satisfaction.

Another key issue that arises from customer input heterogeneity is ensuring *quality consistency* of the delivered services. It has been shown that heterogeneity of customer input, combined with the everyday fluctuations of the availability of human resources used in a service, create a vicious cycle that can drive service quality into a downwards spiral (Oliva & Sterman, 2001). This work also argued that to prevent erosion of service quality it is necessary to monitor it constantly and adequately, and that the most effective way to control it is to aim to delight the customer, not to please him.

We are starting to believe that this heterogeneity of input and output questions the very core foundations of the HCI practice. In particular, we have seen how difficult is to perform traditional user-centered design in the context of the lack of prototypical users and tasks created by heterogeneity. Not only it is extremely difficult to cover a reasonable spectrum of customers during usability tests but also it is hard to recreate in a laboratory the right context, diversity of tasks, and expectations. This is corroborated by the often common practice of web developers of tackling the heterogeneity issue by using extremely fast prototyping methods so they can *beta-test* the online application with a large number of actual customers instead of running in-laboratory usability experiments (a typical case are *Google labs* applications).

Simultaneity Issues

Simultaneity, also called inseparability, is the characteristic of service processes that refers to the fact that often services are produced and consumed at the same time. Production cannot start until the customer provides his share of the inputs, preventing inventory of output, a technique often used in manufacture to balance production. In other words, online services have to rely on a very unreliable input supplier, their own customers, and whether and when they need the services. Since demand for services is often very hard to predict, online service applications tend to exhibit fluctuations in performance, usually exhibiting the worst behavior when the largest number of customers uses them.

Performance consistency affects the perception and usefulness of an interface. Imagine a web search engine which, during peak times, takes 30 seconds to return the 10 best results of the search. This delay would make the customer very upset if the results returned were inappropriate. But most of us handle everyday hundreds of inappropriate search results from *Google*, for example, arguably because the results are given in 2-3 seconds. Traditional HCI tends to ignore performance issues or, simply assume that performance is constant through time and task. One way traditionally used in services to handle performance consistency issues is to have different processes, interfaces, and even content to handle differently the variations in performance. For example, sometimes news websites simplify radically the opening page when dealing with situations of extremely important news that generate levels of access beyond the delivery capabilities of the system.

Another aspect of online service software that interfaces have to take in account is **fairness** in situations involving multiple customers. For example, when multiple customers of an online auction try to post a bid almost at the same time, it is important to make it sure that their bids are processed in the exact order they are received. Unlike in a traditional auction where all bidders can see each other, it is very hard for the online bidder to evaluate whether the online auction is being fairly conducted, increasing the demands on trust and transparency of the interface.

At the same time, simultaneous customers accessing the same resource, for instance when buying the last pair of tickets for a concert, may require an interface design that clearly alerts them to the fact that even during the process the resource may be taken by other customer. While in brick-and-mortar service providers it is often possible for customers to understand that other customers are "ahead" of them in a line, it is unusual to provide the same kind of feedback in an online service. Nevertheless, service software interface designers should always try to make multiple customer resource access as much visible as possible for the sake of preserving the trust of the customers in the fairness of the service. After all, service visibility is one of the bastions of good quality service (Lovelock & Wirtz, 2004).

Perishability Issues

Perishability refers to the fact that often service production capacity is lost whenever there is no request for it. The capacity to host a guest in a hotel room is lost forever when the room is empty. Traditionally in services this issue is tackled with *demand management*, when, for instance, a service provider offer incentives (for example, lower prices) for customers to use the service in times of low demand.

Traditional HCI normally addresses only the situation of actual use of an application. Even the idea of marketing to increase the use of a tool, beyond what is needed for the purchase of the tool, tends to be the least of the concerns of designers. A similar situation occurs in highly automated service software where the only penalty for having a small number of customers may be relatively idle servers. The key challenges for HCI in this area refer to systems where the main benefit stems from multiple customers using the service at roughly the same time. For example, online chatting requires multiple users to be available at the same time to become valuable; online auctions need multiple bidders to be emotionally engaging and profitable for sellers; on-line multiplayer games tend to be boring when fewer enemies are around; and long distance VoIP systems such as *Skype* require each customer to have communication partners online to be useful.

How to make interfaces that incentives use in down times for thousands of online services? We notice that *marketing* the use and need of an application is basically absent from interface design and evaluation of traditional tool software. Even in most situations of web design, the normal posture of HCI professionals is simply to consider that usability and elegance sell by itself and to avoid the integration of marketing concerns to the interface design, with a possible exception in the case of advertisement banners (which HCI types loath, anyway). Although there are potentially some ethical issues here, there are many situations where a large number of customers is beneficial to the individual customer as discussed before. So we advocate an increasing understanding and use of marketing techniques as a way to deal with perishability issues in online service software. For instance, an online auction system may include "live" chats with human experts during low traffic times to increase overall presence.

Another way to cope with perishability is to make the service provider invite customers to use the service when there are fewer than needed customers. The 1990s witnessed a lot of discussion about pull vs. push software, and in general, people have been very resistant to software which tries to push their usage. Nevertheless, the proliferation of viruses and mal-ware has made more common situations where computer software, such as antivirus and firewalls tools, interrupts the user and requests her attention. Similarly, instant messaging and VoIP systems also generate push-like interruptions, and a host of techniques have been created to manage them, such as "busy" flags.

We believe that this context is creating a situation where push techniques became more acceptable and usable in the context of online service applications. For example, auction customers may agree to install service daemons in their machines that may warn them, through a pop-up, that an auction is going to end soon and that there is a possible bargain given the current prices. For the service provider, the issuing of commands to generate pop-ups may be managed so it avoids auctions without sufficient customers but at the same time it does not importunate customers too often. This kind of concern has been largely studied in marketing and advertisement, but it is a rarity in interface design and evaluation. Also, for HCI practice, evaluating push software with laboratory techniques is particularly hard because it requires recreating situations where the customer has to be interrupted.

Coproduction Issues

Coproduction refers to the common practice in services to ask the customer to perform part of the production process, often doing the labor that otherwise would have to be performed by an employee of the service provider. The classical example is when customers help to clean up in fast food stores by taking their garbage to trash bins. Although coproduction is often introduced in a service process to decrease costs (for example, the airlines' self-service kiosks), many times coproduction has a desirable effect of empowering the customer and allowing more informative choices (for example, in the case of online travel services such as *Travelocity*), and even increasing customer satisfaction as described in (Zeithaml et al., 2006, chapter 13). Also, in many services coproduction is absolutely required, for example when a change in lifestyle or behavior is required in a medical treatment. When a doctor asks a patient to take some medication or quit smoking, the patient, for all purposes, is being invited to coproduce the cure.

Software developers are traditionally trained in a mindset where the goal of software is to automate a task, in the context of input from the user. In that framework, it is not a surprise that the user is often seen as an outsider of the production process. Notice that the word "interface" itself expresses an idea that the user is external to the system. Coproduction has often been used in online applications, though often disguised and many times misunderstood. *Google Web Search* is based on the notion that the customer can do a lot of information filtering herself as long as a reasonable summary is provided and the response time is fast. Similarly, online travel service providers such as *Travelocity* have pushed most of the travel agent's job to the customer. In many of these cases there is some loss of quality when part of the production process is moved to the customer, since less expert knowledge is brought to bear by human professionals. However, we should also recognize some key benefits of coproduction in those cases. Coproduction often tends to foster *customer empowerment*. For instance, direct access to information about travel gives the customer more time to reflect and weight options without the pressure of making a decision. As much as not having a waiter ready to clean up the table at a fast food restaurant allows customers to prolong their stay as they wish, coproduction in an online service application can be used to break down different steps of the production process in a pace that can be more convenient and pleasant for the customer.

However, HCI practitioners should be careful about how coproduction works and its impact in their evaluation techniques. For instance, the duration of a task, often used in usability studies, is not an appropriate measure when customers are taking time to decide among different options, gathering more information, or weighting risks. Also, coproduction often involves some level of *customer training*, so the interface design has to consider carefully how the customer is going to learn the skills needed to coproduce effectively. There are many interesting teaching techniques that have been developed by traditional services — see (Zeithaml et al., 2006) for some examples — and the HCI community can definitely learn from them.

Intangibility Issues

Intangibility refers to the fact that many of the key aspects related to customer satisfaction in services are very hard to quantify and measure, especially in a systematic and cost-effective way. Although in the past intangibility was often used as the distinguishing mark of services from goods, current services theorists tend to downplay or even negate intangibility as a defining characteristic of services (Sampson & Froehle, 2005).

One of the key distinctions between traditional tool applications and online service applications is the importance of service quality. How to create and maintain customer satisfaction cannot be an afterthought of interface design, but an essential part of the design and evaluation process. However, it is know in Service Science that customers tend to have strong *service expectations* about the quality of the service they are going to receive. Customers tend to incorporate into such expectations the price of the service, their prior experiences with the provider and with other providers, the location of the service, etc. An extensive body of literature in Service Science has examined the role of expectations when measuring service quality; a good summary can be found in (Schneider & White, 2004, chapter 2). As a consequence, the most commonly used service quality instrument by the services industry, *SERVQUAL* (Parasuraman, Zeithaml, & Berry, 1985), is in fact based on measuring the difference between service quality perceptions and expectations, or what is commonly known as the *gap-model approach*. There are many theoretical and statistical reasons to measure the gap between perception and expectation instead of simply determining the perceived quality of the service (Schneider & White, 2004, chapter 2). However, the most obvious advantage of using the gap-model approach is that it provides actionable information —which areas of the service are below what customers expect.

There is some strong evidence that, in fact, the gap-model is also the right way to measure service quality in online services (Trocchia & Janda, 2003; Zeithaml, Parasuraman, & Malhotra, 2002), giving rise to specific service quality instruments for web sites such as *WebQual, SiteQual,* and *eTailQ* (Barnes & Vidgen, 2000; Loiacono et al., 2007; Parasuraman et al., 2005; Szymanski & Hiseb, 2000; Webb & Webb, 2004; Wolfinbarger & Gilly, 2003; Yoo & Donthu, 2001). Interestingly, user expectations and gap measurement have been used very sporadically by the HCI community, for example in the work of (Bouch, Kuchinsky, & Bhatti, 2000). This kind of work which validates and adapts a traditional service tool, *SERVQUAL,* into online-specific versions is precisely the kind of approach we are advocating in this article.

Another important issue in services is *process satisfaction*. In many service situations the way the customer is treated during the service process may have a larger impact on customer satisfaction than the actual delivered service. For example, dieting clubs with great customer experiences tend to have a higher rate of customer loyalty, in spite of the fact that in most cases the customers do not achieve their actual goal of losing weight. Beyond the traditional goal of task completion used in HCI, process satisfaction of an online service has to do with many more intangible aspects of the experience such as fairness, politeness, aesthetics, speed, humor, etc.

Finally, the fact that the customer does not control the means of production increases the need for the online service interface designers to deal with issues of *anger and frustration management*. Unlike in the case of traditional applications, where users in many cases vent their frustration on themselves, in services the existence of the service provider as a separate, "conscious", humn-like entity enables the customer to transfer the anger or frustration to the service provider.

Online Service Interfaces as Relationship Maintainers

Service Science traditionally regards the interactions between service providers and customers as long-term relationships. We can argue that the need for relationships stems from both the users' lack of control of the means of production, therefore forcing them to connect to another entity, the service provider, and establish a relationship; and from the fact that in most cases there are a multitude of competing service providers, so it is also interesting for the service provider to seek long-term relationships.

Also, if we examine to the collection of 15 issues identified as very relevant to online service interfaces, it becomes apparent that most of them are core issues when establishing and/or maintaining a relationship. For example, trust, privacy, security, fairness, consistency, recovery, empowerment, and anger/frustration management are clearly aspects of a healthy relationship.

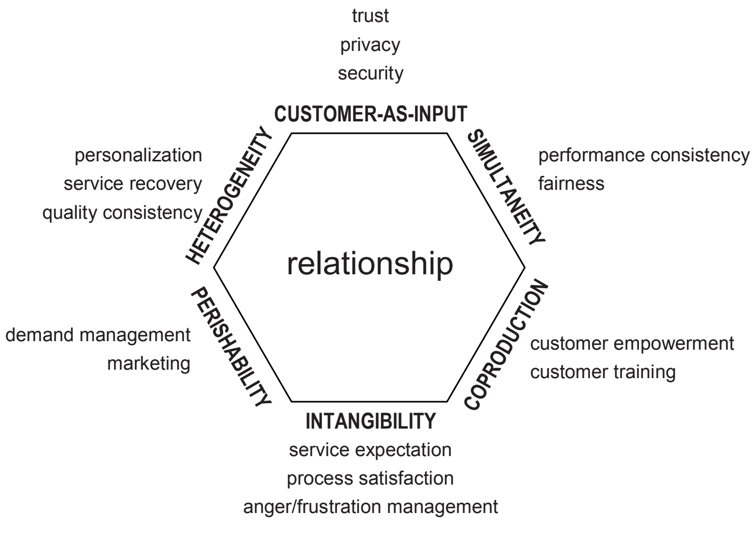
Figure 4 summarizes the important new issues for interfaces of online services as identified through our Service Science framework and highlights the key aspect of an online service interface: being inductive to establish and maintain relationships.

This view of interfaces as relationship maintainers sharply contrasts to traditional understanding of software tool interfaces, which have been regarded as agents for *conversation* (Walker, 1990), *action* (Norman, 1988), *direct manipulation* (Shneiderman, 1987), or *representational action* (Laurel, 1991). Furthering the understanding the relationship aspect of services and online services, in particular, is currently a key part of our research efforts.

DISCUSSION AND FUTURE WORK

In this article we have proposed a definition of online service applications which allow the use Service Science as a reference framework to address their design, evaluation, and deployment issues. In our experience, designers and engineers of online services are mostly unaware of Service Science concepts, and therefore, by establishing this definition we are able to create a "safe" bridge which allows the use of techniques developed for traditional services in the realm of online services. We then exemplified how this merge of well-established concepts

Figure 4. Important issues for interfaces of online services



and a sound theoretical definition can be used by HCI practitioners to create a new reference framework for design and evaluation of online services based on the concept of relationship, in contrast with the traditional viewpoints of "action", "conversation", "representational action", "direct manipulation", etc.

At this point the reader can be understandably questioning where the evidence is that the Service Science framework and techniques are really important for online service applications. We have seen at least two documented cases — the line waiting study of Ryan and Valverde (Ryan & Valverde, 2006) and the application of *SERVQUAL*-like methods for evaluation of online services (Barnes & Vidgen, 2000; Loiacono et al., 2007; Parasuraman et al., 2005; Szymanski & Hiseb, 2000; Webb & Webb, 2004; Wolfinbarger & Gilly, 2003; Yoo & Donthu, 2001)— where traditional services techniques are shown to be more appropriate to the online services domain than traditional HCI methods.

Said that, we have discussed here, notably in the previous section, many examples where simply by taking the six basic characteristics of services as a springboard, we were able to provide a better explanation for common difficulties facing online services interface design and evaluation and/or suggest new techniques and approaches to solve known HCI problems in the area. An example of the former is how input and output heterogeneity, a known and often studied issue in Service Science, seems to shed light onto the difficulties of online services evaluation. A typical example of the latter case is the discussion we had about *SERVQUAL* and how measuring service quality as the gap from expectations and perceptions are not only often more realistic but also more informative than simply measuring satisfaction.

As researchers with an HCI background, we were astonished when we realized the depth, breadth, and quality of the service quality literature, their relevance to online services, and how the HCI community is mostly unaware of it. Notice, however, that most of the literature in *SERVQUAL*-like evaluation methods (Barnes & Vidgen, 2000; Loiacono et al., 2007; Parasuraman et al., 2005; Szymanski & Hiseb, 2000; Webb & Webb, 2004; Wolfinbarger & Gilly, 2003; Yoo & Donthu, 2001) tends to look into usability issues in a very simplistic way: for instance, by asking the user whether a website has good usability. As well-known in the HCI community, it is not always the case that the user is aware of usability problems, which often need to be detected through direct observation of users working with the interface. Therefore, we see that there is a promising opportunity for the development of direct-observation evaluation techniques of the user experience of online services that integrate the concept of expectation gap of Service Science with the task-based evaluation methods traditionally used in HCI.

Though promising, those two examples of application of Service Science are more the exception than the rule in the practice of designing and building online services. Therefore, we believe there is an enormous opportunity to establish a framework for online service applications based on Service Science. In particular, we hope that this introductory discussion creates questioning and curiosity in the field and trigger further research. In our case, we are focusing efforts in codifying and adapting traditional services methodologies to the design and evaluation of online service applications. For instance, we have been trying to use *Service Design* methodologies (Shostack, 1984; Zeithaml et al., 2006, chapter 9) to reinvent the user experience in online technical support.

From online multiplayer games to web search, many of the new components of the online information landscape are structured as services, but they are still often designed and evaluated under the traditional "tool" view of computing. We hope we are contributing to change this mindset, towards increasing not only the quality of customer experience but also the efficiency and adequacy of online service applications.

REFERENCES

Barnes, S. J., & Vidgen, R. T. (2000, July 3-5). *WebQual: An Exploration of Web-Site Quality.* Paper presented at the Proceedings of the 8th European Conference on Information Systems, Trends in Information and Communication Systems for the 21st Century, ECIS 2000, Vienna, Austria.

Bieberstein, N., Sanjay, B., Fiammante, M., Jones, K., & Shah, R. (2006). *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*. Upper Saddle River, NJ: IBM Press.

Bouch, A., Kuchinsky, A., & Bhatti, N. (2000, April 1-6). *Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service.* Paper presented at the Proc. of CHI'00, The Hague, The Netherlands.

Brodie, C., Karat, C.-M., & Karat, J. (2004). Creating an e-Commerce Environment Where Consumers are Willing to Share Personal Information. In C.-M. Karat, J. O. Blom & J. Karat (Eds.), *Designing Personalized User Experiences in eCommerce* (pp. 185-206): Kluwer Academic Publishers.

Cerf, V. G. (1991). Networks. *Scientific American, 265*(3), 42-51.

Dertouzos, M. L. (1991a). Communications, Computers and Networks. *Scientific American, 265*(3), 30-37.

Dertouzos, M. L. (Ed.). (1991b). *Scientific American, Special Issue on Communications, Computers, and Networks* (Vol. 265 (3)).

Dhamija, R., Tygar, J. D., & Hearst, M. (2006). *Why Phishing Works.* Paper presented at the Proc. of CHI'06, Montreal, Canada.

Featherman, M. S., Valacich, J. S., & Wells, J. D. (2006). Is That Autentic or Artificial? Understanding Consumer Perceptions of Risk in e-Service Encounters. *Information Systems Journal, 16*(2), pgs. 107-134.

Ferguson, D. F., & Stockton, M. L. (2005). Service-Oriented Architecture: Programming Model and Product Architecture. *IBM Systems Journal, 44*(4), 753-780.

Fitzsimmons, J. A., & Fitzsimmons, M. J. (2004). *Service Management: Operations, Strategy, and Information Technology*

Fogg, B. J., Marshall, J., Laraki, O., Osipovich, A., Varma, C., Fang, N., et al. (2001, Mar 31-Apr 5). *What Makes Web Sites Credible? A Report on a Large Quantitative Study.* Paper presented at the Proc. of CHI'01, Seattle, Washington.

Heskett, J. L., Sasser, W. E., & Schlesinger, L. A. (1997). *The Service Profit Chain: How Leading Companies Link Profit and Growth to Loyalty, Satisfaction, and Value*. New York: Free Press.

Kahn, R. E., & Cerf, V. G. (1988). *The Digital Library Project, volume I: The World of Knowbots*. Unpublished manuscript, Reston, VA.

Karat, C.-M., Blom, J. O., & Karat, J. (Eds.). (2004). *Designing Personalized User Experiences in eCommerce*: Kluwer Academic Publishers.

Karat, J., Karat, C.-M., & Brodie, C. (2004). Personalizing Interaction: Directions for HCI Research. In C.-M. Karat, J. O. Blom & J. Karat (Eds.), *Designing Personalized User Experiences in eCommerce* (pp. 7-17): Kluwer Academic Publishers.

Karat, J., Karat, C.-M., Brodie, C., & Feng, J. (2005). Privacy in Information Technology: Designing to Enable Privacy Policy Management in Organizations. *International Journal of Human Computer Studies, 63*, 153-174.

Laurel, B. (1991). *Computers as Theatre*. Reading, Massachusetts: Addison-Wesley.

Loiacono, E. T., Watson, R. T., & Goodhue, D. L. (2007). WebQual: An Instrument for Consumer Evaluations of Web Sites. *To appear in the Internation Journal of Electronic Commerce*.

Lovelock, C. H., & Wirtz, J. (2004). *Services Marketing: People, Technology, Strategy* (5th ed.). Upper Saddle River, N.J.: Pearson/Prentice Hall.

Nah, F. F.-H., & Davis, S. (2002). HCI Research Issues in Electronic Commerce. *Journal of Electronic Commerce Research, 3*(3), pg. 98-113.

Nielsen, J. (2000). *Designing Web Usability: The Practice of Simplicity*. Indianopolis, Indiana: New Riders Publishing.

Norman, D. A. (1988). *The Psychology of Everyday Things*. New York, New York: Basic Books.

Norman, D. A. (2006, Sep+Oct). Words Matter. Talk About People -- Not Customers, Not Consumers, Not Users. *Interactions, 13,* 49,63.

Oliva, R., & Sterman, J. D. (2001). Cutting Corners and Working Overtime: Quality Erosion in the Service Industry. *Management Science, 47*(7), 894-914.

Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1985). A Conceptual Model of Service Quality and Its Implications for Future Research. *Journal of Marketing, 49*(Fall 1985), 41-50.

Parasuraman, A., Zeithaml, V. A., & Malhotra, A. (2005). E-S-QUAL: A Multiple-Item Scale for Assessing Electronic Service Quality. *Journal of Service Research, 7*(3), 213-233.

Pinhanez, C. (2007, July 10-13). *Ubiquitous Services.* Paper presented at the Proc. of IEEE Services Computing Conference - SCC'07, Salt Lake City, Utah.

Pinhanez, C. (2008). *Service Systems as Customer-Intensive Systems and its Implications for Service Science and Engineering.* Paper presented at the Proc. of Hawaiian International Conference on System Sciences, HICSS-41, Big Island, Hawaii.

Ryan, G., & Valverde, M. (2006). Waiting in Line for Online Services: a Qualitative Study of the User's Perspective. *Information Systems Journal, 16*(2), pgs. 181-211.

Sampson, S. E. (2001). *Understanding Service Businesses* (2nd ed.): John Wiley & Sons.

Sampson, S. E., & Froehle, C. M. (2005). Foundations and Implications of a Proposed Unified Services Theory. *Submitted to Production and Operations Management*.

Sampson, S. E., & Froehle, C. M. (2006). Foundations and Implications of a Proposed Unified Services Theory. *Production and Operations Management, 15*(2), 329-343.

Schneider, B., & White, S. S. (2004). *Service Quality: Research Perspectives*. Thousand Oaks, Calif.: Sage Publications.

Shardanand, U., & Maes, P. (1995, May 7-11). *Social Information Filtering: Algorithms for Automating Word of Mouth.* Paper presented at the Proc. of CHI'95, Denver, Colorado.

Shneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, Massachusetts: Addison-Wesley.

Shneiderman, B. (2000). Designing Trust into Online Experiences. *Communications of the ACM, 43*(12), 57-59.

Shostack, G. L. (1984). Designing Services that Deliver. *Harvard Business Review, 62*(1), 133-139.

Spohrer, J., Maglio, P. P., Bailey, J., & Gruhl, D. (2007). Steps Toward a Science of Service Systems. *IEEE Computer*, 71-77.

Spohrer, J., & Riecken, D. (Eds.). (2006). *Communications of ACM: Special Issue on Services Science* (Vol. 49:7).

Szymanski, D. M., & Hiseb, R. T. (2000). E-satisfaction: An Initial Examination. *Journal of Retailing, 76*(3), 309-322.

Terveen, L., & McDonald, D. W. (2005). Social Matching: A Framework and Research Agenda. *ACM Transactions on Computer-Human Interaction, 12*(3), 401-434.

Trocchia, P. J., & Janda, S. (2003). How Do Consumers Evaluate Internet Retail Service Quality? *Journal of Services Marketing, 17*(3), 243-253.

Voss, C. (2003). Rethinking Paradigms of Service: Service in a Virtual Environment. *Internation Journal of Operations & Production Management (IJOP), 23*(1), 88-104.

Walker, J. (1990). Through the Looking Glass. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design*. Reading, Massachusetts: Addison-Wesley.

Wang, Y. D., & Emurian, H. H. (2005). An Overview of Online Trust: Concepts, Elements, and Implications. *Computer in Human Behavior, 21*, 105-125.

Webb, H. W., & Webb, L. A. (2004). SiteQual: An Integrated Measure of Web Site Quality. *Journal of Enterprise Information Management, 17*(6), 430-440.

Wolfinbarger, M., & Gilly, M. C. (2003). eTailQ: Dimensionalizing, Measuring and Predicting etail Quality. *Journal of Retailing, 79*(3), 183-198.

Yoo, B., & Donthu, N. (2001). Developing a Scale to Measure the Perceived Quality of an Internet Shopping Site (SITEQUAL). *Quarterly Journal of Electronic Commerce, 2*(1), 31-47.

Zeithaml, V. A., Bitner, M. J., & Gremler, D. D. (2006). *Services Marketing: Integrating Customer*

Focus Across the Firm (4th ed.). Boston: McGraw-Hill/Irwin.

Zeithaml, V. A., Parasuraman, A., & Malhotra, A. (2002). Service Quality Delivery through Web Sites: A Critical Review of Extant Knowledge. *Journal of the Academy of Marketing Sciences, 30*(4), 362-375.

Claudio Pinhanez is a computer scientist, a service scientist, and a media artist. He has been a research scientist at the IBM Thomas J. Watson Research Center since 1999, where he currently conducts research on service science (theory, tools, methodology, and applications); and in ubiquitous computing and advanced computer interfaces. Pinhanez was born in Brazil, where he received a bachelor's degree in mathematics and a MSc in computer science from the University of São Paulo. He obtained his PhD in media arts and sciences in 1999 from the MIT Media Laboratory. Pinhanez has more than 90 international publications, including 3 best paper awards, and several patents in the US, Japan, and EU.