J Internet Serv Appl (2011) 1: 155–164
DOI 10.1007/s13174-010-0017-4

ORIGINAL PAPER

Human aspects of internet services: considering the needs of users and providers

Claudio S. Pinhanez

Received: 20 November 2010 / Accepted: 1 December 2010 / Published online: 13 January 2011

Abstract This paper discusses the need and possible forms of human interfaces to Internet services, challenging the common notion that Internet services are simply computer-computer systems governed by machine protocols with little need of concern for human issues. We first examine the case where Internet services are provided to human users, showing that the user-system interaction becomes a typical service relationship, which can be better understood in the framework of Service Science. Based on the six basic characteristics of services we explore 15 issues which should be taken in account when designing human interfaces for Internet services. We also depart from traditional HCI by arguing that the fundamental goal of the human interface of an Internet service is to create and maintain a relationship with the user. We then look into Internet services being used by computer applications, where we discuss the need of a back-door human interface for the maintenance and control people working in the Internet service provision system. Both situations have been little explored by the Internet services or the HCI fields of research.

Keywords Internet services · Service systems · Service science · Online services · Service interfaces · Maintenance interfaces

1 Introduction

Internet services[1] are becoming the very fabric of most of the computer applications being built today. This is the result of a considerable amount of research [1, 2], especially in fields such as networking, distributed systems, and software architecture.

The main argument of this paper is that there are key human aspects of Internet services which have been mostly ignored by research in the area. We believe there is an illusion that Internet services are fundamentally only a dialogue between computers: a *client* or *computer user* (in our terminology) and an *Internet service*, a computer application which provides data requested by the computer user. In this paper we challenge this assumption by noticing that a majority of what is usually called an Internet service is in fact a system which includes people (often control and maintenance workers, but also owners). Therefore, Internet services are basically used in a context where either a computer user or a human user is interacting with a system composed of machines and people, thus creating a need of considering the human aspects of those interactions.

The investigation around those human aspects is done in the framework of *service systems* as defined in a previous work [3]. Using this structure, we clearly characterize Internet services as computer applications that are provided by service systems and differentiate two types of Internet services: those which provide services to computer users, called *computer services*, and those providing services to human users, called *online services*. In both cases, we explore the human aspects of the services and the need of special human interfaces.

For the case of online services, we take in this paper a theoretical approach based on 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 [4–6]. This effort has been recently joined by computer scientists and engineers through the *SSME* initiative (*Service Science, Management, and Engineering*) [7, 8].

[1] In this paper we use the terms *Internet service* and *web service* interchangeably.

C.S. Pinhanez (✉)
IBM Research, São Paulo, Brazil
e-mail: csantosp@br.ibm.com

The Service Science framework reveals more clearly where and how interface design and evaluation can be improved by traditional service ideas and, especially, how human-computer interfaces have an additional role as enablers of relationships between human users and online service providers. Although there has been some work examining how traditional Service Science concepts apply to online services—such as Ryan and Valverde's study on waiting in line effects on consumer behavior [9] and many works on service quality measurement of online services [10–16]—our approach here is to point out a much larger view of this issue. We also discuss how the traditional approaches for user interface design, deployment, and evaluation employed by the HCI community and practitioners may be inadequate or incomplete in the case of online services exactly because they fail to consider the service aspects of the interaction.

The human aspects of computer services are examined considering the reverse point-of-view of the needs of the people inside the service system hosting the Internet services, often maintenance and control workers. We argue that due to the increasing diversity and complexity of the computer user applications interacting with computer services, it is necessary to provide *backdoor human interfaces* for those applications. Human interfaces allow a much more effective way to handle typical needs such as state and status queries, halting and restarting requests, etc. This situation has similar aspects to human interfaces for machine maintenance and therefore we look for inspiration, methodologies, and advice from fields such as Maintenance Engineering [17].

We start by elaborating the definitions used in this paper and constructing the basic framework of Internet services and its different types.

2 Types of computer applications

The key to our argument is that there are different types of computer applications, some that are true services and some that are just tools. To make this distinction properly and consistently, we rely on our previous work [3] which provides a definition of service systems as those systems where there are people inside during their use. This definition creates an easy distinction between products/tools and services and, by distinguishing the type of user between computers and people, we arrive at four different types of computer applications as shown in the following paragraphs.

2.1 Defining service systems

Our definition of service systems builds on the initial, fundamental concept of *user systems*:

> *A* user system is a physical system that requires the engagement of an external person or organization to produce value for that person or organization [3].

The key notion embodied in this definition is engagement: a user system is not permanently connected to the external person or organization, but instead is engaged and disengaged as value needs to be created.

We call a *user* the person or organization that engages with the user system. Another important term is *use*, which encompasses the processes and results of the value-creation engagement between the user and the user system. Examples of user systems are abundant: cars, roads, gas stations, panoramic stops at roads, beaches, maids, restaurants, newspapers, churches, saltshakers, houses, and of course, Internet services.

The distinction between product and service system is provided simply by the *presence of humans inside the user system during use*:

> *A* product system is a user system which mostly does not contain people or organizations as components during use.

> *A* service system is a user system which contains a significant level of people or organizations as components during use.

An important element of this definition is that the distinction is considered by looking only what happens during the use of the system and not during what happens before or after. For instance, a printed newspaper is a product system because at the moment of use, that is, when the user is reading it, there is no other person involved in the process. However, an Internet news provider is a service system, since during the reading there are some people monitoring the servers, who, although hidden to the user, are present and part of the system when it is used.

In this paper we stretch our previous definition of user systems proposed in [3] to allow computer applications as users. Enlarging the definition to include computer programs implies that we lose the validity of some implications of the definition of service systems as having people inside during use as discussed in [3]. However, in the context of this paper the implications are irrelevant; therefore we do not see any problems in doing so.

Of course, there are many competing theories about how to characterize and classify services, as discussed, for example, by Sampson and Froehle [18]. It is quite beyond the scope and need of this paper to digress on those different views.

2.2 Defining internet services

We consider the space of computer applications along two dimensions: (1) *the type of user* of the computer application, which can be another computer application or a human being or organization; and (2) *the type of provider* of the

PROVIDER \ USER	computer	human
computer system	software tools	
	computer tool	human tool
service system	Internet services	
	computer service	online service

Fig. 1 Types of computer applications

computer application, which can be a computer system or a service system.

Figure 1 presents the four different types of computer applications that are generated by the partition of the space of computer applications along the two dimensions:

- *Computer tools:* computer applications provided by computer systems for computer applications, such as for example, an OS kernel.
- *Human tools:* computer applications provided by computer systems to human users, such as a file browser in a personal computer.
- *Computer services:* computer applications provided by service systems to computer applications, such as a web service that authenticates users in a distributed system (notice that user authentication has to be provided by a system which has constant monitoring by human operators).
- *Online services:* computer applications provided by service systems to human users, such as search engines like *Google* or *Bing*.[1]

Further, we call *computer tools* those computer applications provided solely by computer systems and *Internet services* those provided by service systems. The choice of the term *Internet service* is not accidental, since we believe that when this term is used by professionals it in fact describes a situation where a computer application is run by a service system.

However, our observations have shown that the human dimension of Internet services, as defined here, is often forgotten by developers and system architects, who often treat the interaction between the user of an Internet service, computer or human, with the service system as if the underlying service system had no people at all.

In the case of online services, the presence of people inside the system and people using it establishes human-human interactions mediated by a computer application. However, the traditional perspective of HCI is to regard online services as identical to human tools, thus considering only the human needs of the user and mostly ignoring the presence and possible need of interaction with the people inside the service system.

Similarly, in the case of computer services, this leads to a situation where the interaction between the people inside the service system and the computer application using it is mostly unconsidered. Notice that this human-computer interaction is the inverse of the traditional HCI perspective, where the norm is to consider the user as human and the provider as a machine.

We have examined online services and the issue of user-service system interaction in a previous work [19], under the framework of Service Science findings and knowledge. We summarize in the next section the key issues, ideas, and recommendations of that work and discuss them in the Internet services context. Service Science seems to be the right framework for this analysis since it agglutinates the knowledge produced by academy and practice in situations of people using and interacting with service systems. We explore the case of computer services in the following section.

In both cases, we see our view as a considerable departure from both traditional HCI and the usual concerns of developers and architects of computer applications that use computer services. We believe that this disregard for the human aspects of the service system is a frequent source of errors, difficulties, and confusion, although we have only anecdotal evidence of it.

3 Human interfaces for online services

There has been very little theoretical work in terms of establishing a framework to understand computer applications that take the form of online services in the way defined before 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 [19].

In the HCI research, 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 [20] or the more restrictive class of online retailers [10, 21, 22]. Our approach has been to define online services in such a way that we can safely apply well-known concepts of Service Science in the context of the design of their human user interfaces.

Referring again to Fig. 1, we believe that traditional user interface design has been biased toward the creation of interfaces for what we call human tools. In terms of user interface design evaluation and usability issues, a lot of effort is put in determining the typical individual usage scenarios of the tool and then to recreate in the laboratory meaningful

[1]We use the term online services to be consistent with some previous work we did on this kind of computer application [19].

test procedures. The different dynamics of online services has required user interface design practitioners to change their techniques to reflect some of the special needs of online services as discussed, for example, in the handbook by Nielsen [20].

We structure our discussion by considering the basic characteristics of services as usually considered in Service Science. We compiled and fused service characteristics listed by different authors [4–6, 23], arriving at a "compromise" list which we believe most of them would agree to: *customer-as-input*, *heterogeneity*, *simultaneity*, *perishability*, *coproduction*, and *intangibility*. We have looked into these six characteristics and examined how each of them highlights some issues which are very relevant to the design of interfaces for online services. The result is the list of 15 important issues for the design of online service applications described in detail in [19], and summarized here. Issues are agglutinated by the prevailing service characteristics that best explain their need.

3.1 Customer-as-input issues

Customer-as-input refers to the fact that in services the production process often uses significant inputs from the human user. As described by Sampson in [18, 23], 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. Notice that in all cases, the production process is unable to even start until the user provides the input.

However, not all online services require the user to be an essential part of the input to the production process as pointed by us in a previous work [24]. 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 triggered 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 on 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*.

We believe an immediate consequence of users' 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 human tools, where the privacy of data is often taken for granted and trust on the tool is often assumed to be unlimited, dealing with an online service provider always involves an exchange of *trust* between the parties. Users 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 out in Wang and Emurian's overview [25] 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*." [25, p. 21]. A study on websites credibility by Fogg et al. [26], as well as Brodie et al. study of e-commerce environments [27], share similar recommendations, also present in a well-known set of design guidelines for online experiences by Shneiderman [28]. A similar hypothesis was tested and found true in other experiments [29], although other factors seem to influence the perceived risk, including the computational literacy of the user 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 is not only on front-end issues but also on making the back-end workings of services more "transparent" and visible to the users. For 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 elicits trust by making its internal workings more visible: "trust what I do" instead of "trust what I say".

Dealing with *privacy* of information is also an issue that becomes fundamental in online services, since personal 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 user, which part to actually record, and to decide the trustworthiness of the information provided. Also, often the privacy guarantees were part of the personal relationship between user 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 asks for information that is particularly sensitive the interface should clearly inform the user why the system needs it, what the privacy policy is, for how long the personal information 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 also a good practice.

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 users is through guarantees of satisfaction, such as return policies and "your-money-back, no-questions-asked" mechanisms. Notice also that security in service systems 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 users.

3.2 Heterogeneity issues

Heterogeneity is used in Service Science to address the fact that in services users tend to be very unique in their identity and requirements, so the execution of a service is usually highly tailored and quite unique to a user request and input. One of the issues brought by 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 [30], often through the use of some sort of reasoning on top of a user model. For example, Karat et al. [31] 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, 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.

A related key issue is the heterogeneity of output, which requires the service designer to consider instances where the service is not delivered successfully. It is interesting how traditional user interface design 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 [6, Chap. 8].

Another key issue that arises from user input heterogeneity is ensuring *quality consistency* of the delivered services. It has been shown that heterogeneity of user 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 downward spiral [32].

We have started to believe that this heterogeneity of input and output questions the very core foundations of the HCI practice. We have seen how difficult is to perform traditional user-centered design in the context of heterogeneity, which defies the enumeration of prototypical users and tasks, a key to most methodologies in HCI. Not only it is extremely difficult to cover a reasonable spectrum of users 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 online service 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 users instead of running in-laboratory usability experiments (a typical case are *Google labs* applications).

3.3 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 user provides her 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 very unreliable input suppliers—their own users. 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 number of users is the largest.

Performance consistency affects the perception and usefulness of an Internet service. 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 user very upset if the results returned were inappropriate. But most of us handle everyday hundreds of inappropriate search results from *Google* or *Bing*, arguably because the results are given in 2–3 seconds. Traditional user interface design tends to ignore performance issues or simply assume that performance is constant through time and task. In contrast, 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 users. For example, when multiple users 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.

At the same time, simultaneous users 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 another user. While in brick-and-mortar service providers it is often possible for users to understand that other users are "ahead" of them in a line, it is unusual to provide the same kind of feedback in an online service.

3.4 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 offers incentives such as lower prices for using the service in times of low demand.

Traditional user interface design 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. The key challenges for HCI in this area refer to systems where the main benefit stems from multiple users using the service at roughly the same time. For example, online auctions need multiple bidders to be emotionally engaging for users and profitable for sellers; online multiplayer games tend to be boring when fewer enemies are around; and long distance VoIP systems such as *Skype* require users to have their communication partners online.

How to make interfaces which incentive use in downtimes? We notice that *marketing* the use and need of an application is basically absent from interface design and evaluation of traditional tool software. So we advocate an increasing understanding and use of marketing techniques as a way to deal with perishability issues in online services. 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 users to use the service when there are fewer than needed users. 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 situations more common where computer software, such as antivirus and firewalls tools, interrupts the user and requests her attention. 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 users 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. Another possibility is to use highly perishable information media such as *Twitter*.

3.5 Coproduction issues

Coproduction refers to the common practice in services to ask the user to perform part of the service production process, often performing the labor that otherwise would have to be done by the service provider. The classical example is when users help to clean up in fast food stores by taking their trays to trash bins. Although coproduction is often introduced in a service to decrease costs (for example, the airlines' self-service kiosks), many times coproduction has a desirable effect of empowering the user and allowing more informative choices (for example, in the case of online travel services such as *Travelocity*), and even increasing user satisfaction as describe by Zeithaml, Bitner, and Gremler [6, Chap. 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.

Coproduction has often been used in online applications, though often disguised and many times misunderstood. *Google Search* is based on the notion that the user 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 user. However, we should also recognize some key benefits of coproduction in many cases. Coproduction often tends to foster *customer empowerment*. For instance, direct access to information about travel gives the user more time to reflect and weight options without the pressure of making a decision. 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 user.

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 users 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 user 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, Bitner, and Gremler [6] for some examples.

3.6 Intangibility issues

Intangibility refers to the fact that many of the key aspects related to user satisfaction in services are very hard to quantify and measure, especially in a systematic and cost-effective way.

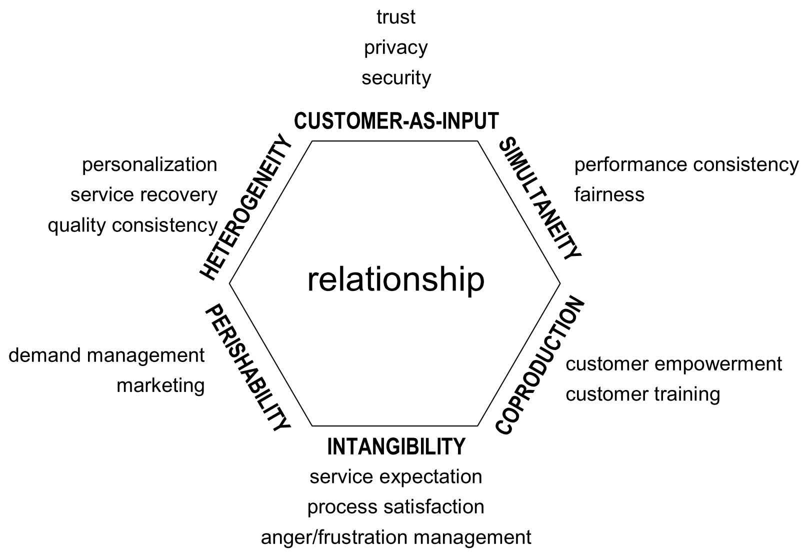

Fig. 2 Diagram showing the 15 issues for human interfaces of online services and the centricity of relationship management

One of the key distinctions between traditional human tools and online services is the importance of service quality. How to create and maintain user satisfaction cannot be an afterthought of the human interface design, but should be an essential part of the design and evaluation process. However, it is know in Service Science that users tend to have strong *service expectations* about the quality of the service they are going to receive. Users 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 the work of Schneider and White [33, Chap. 2]. For example, the most commonly used service quality instrument, *SERVQUAL* [34], 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 [33, Chap. 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 user expectations.

There is some strong evidence that, in fact, the gap-model is also the right way to measure service quality in online services [35, 36], giving rise to specific service quality instruments for online services such as *WebQual, SiteQual,* and *eTailQ* [10–16]. Interestingly, user expectations and gap measurement have been used very sporadically by the HCI community; an exception is the work of Bouch, Kuchinsky, and Bhatti [37].

Another important issue in services is *process satisfaction*. In many service situations the way the user is treated during the service process may have a larger impact on user satisfaction than the actual delivered service. For example, dieting clubs with great user experiences tend to have a high rate of user loyalty, in spite of the fact that in most cases the user does not achieve their actual goal of losing weight. Beyond the traditional goal of task completion often 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, online service interface designers have 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 the front-line employees, in online services it is rare to find an outlet to express anger and frustration with the service results or its process. Giving the intensity of such emotions and the overall impact they may have in the perception of the online service, mechanisms for anger and frustration expression should be a "must-have" in online service interfaces.

3.7 Online service interfaces as relationship managers

Service Science traditionally regards the interactions between service providers and users as long-term relationships. Also, if we examine 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 maintaining a relationship. For example, trust, privacy, security, fairness, consistency, recovery, empowerment, and anger and frustration management are clearly aspects of healthy relationships. Figure 2 summarizes the 15 issues for interfaces of online services as identified through our Service Science framework and highlights, in our opinion, the key aspect of an online service interface: being inductive to establish and maintain relationships.

This view of interfaces as relationship managers sharply contrasts to traditional understanding of computer interfaces, which have been regarded as agents for *conversation* [38], *action* [39], *direct manipulation* [40], or even *representational action* [41].

4 Human interfaces for computer services

Having considered the issues related to the human user relationship with service systems and how they may affect the design and evaluation of user interfaces in the previous section, we now turn our attention toward the more subtle but far less explored issue of addressing the needs of the people inside service systems. By our own definition of service systems, there are always people as part of a service system while it is being used, and occasionally they have to interact with the user—computer or human.

In particular, we want to examine here the situation where people inside the provider system have to interact with a user which is a computer application, that is, in computer services. Observing workers troubleshooting servers in IT outsourcing delivery systems, it is not uncommon to see situations where a problem involves the way a computer user application is interacting with machines in the service system. Often, the troubleshooters are rendered helpless when the user application seems to be misbehaving. Although sometimes the maintenance workers in the service systems can monitor to some level the traffic between the user application and server systems, in most cases there is hardly any way to directly interact with the user application to investigate issues, query information, and stop or start behaviors. In the rare cases where this is possible, it is often by mimicking the communication between the computer service and the computer user, almost always through very cumbersome procedures. After all, the language and interaction was designed to be performed by machines and not between a human being and a machine.

The origin of this kind of issue is, of course, the architectural mistake of failing to acknowledge, in the case of computer services, that there are going to be people interacting with the computer application in some situations, especially of maintenance. In other words, we are advocating here that computer applications which use service systems should provide a *backdoor human interface* to be accessed by the people inside the service systems when it is needed.

Although there have been many works dealing with backdoor access to running computer applications for maintenance (for example [42, 43]), such works often establish machine protocols for communication and not a human interface. This distinction is very important indeed, because backdoor machine protocols for maintenance require the service systems workers to have a maintenance application able to read and write the particular protocol established by the computer user application. Given the normal conditions of computer services systems, which may be providing services to hundreds of computer user applications, it is hardly manageable because of the need to have specialized maintenance applications for each of them.

We see two better options here. First, a standard computer protocol could be developed for backdoor maintenance communication with computer applications in service contexts, allowing the construction of a single backdoor interface for human beings. The problem with this approach is the difficulty of describing the extreme diversity of applications and their states, actions, and metrics with a single protocol. In practice, the protocol would likely to have to restrict itself to simple actions such as halting, sending test message, or restarting.

A second option, which may be preferable in most situations, is to establish a practice for computer user applications to provide a backdoor human interface for communication and maintenance. In this way, human operators of service systems are always able to open an interface to a computer application through, let us say, a standard HTML exchange, creating a remote maintenance backdoor usable through a simple web browser.

The issue of human and machine interfaces for access and maintenance is not too much explored by HCI, but has been a quite common theme in Maintenance Engineering and related fields [44, 45]. Although the context in a typical maintenance scenario is slightly different of what we are discussing here, there have been many successes and failures in providing interfaces for people to communicate with machines during maintenance.

Two arguments can be raised against providing backdoor human interfaces for computer applications which use computer services. The first is related to security concerns: a backdoor interface could make the job of attackers easier, given that they would no longer need access to the "secret" language exchange protocols between the application and the service system. This is a valid point, which requires careful design of security mechanisms to prevent unintended access to the backdoor interface. The second issue is related to the additional cost of creating a human interface exclusively for maintenance. We first notice that having this cost is already quite standard for some applications that require human manual configuration, such as home network routers. But the best way to see this is that maintenance costs are often high and seem to be increasing in IT service systems, so efforts and expenses targeting their reduction are likely to be competitive in most situations.

By far most of the time spent by computer applications when using computer services is devoid of the presence of human beings. In that sense, including a backdoor maintenance interface can be seen as a luxury. However, given the rising complexity of computer services, where a computer application may use dozens or hundreds of other machines at the same time, sometimes in different service systems, it may pay off to design and implement backdoor maintenance interfaces. We are still lacking some of the most basic protocols to do so in an effective way, and one of the objectives of this paper is to start the discussion toward them.

5 Final remarks

In this paper we challenge the common notion that Internet services are computer-computer systems governed by machine protocols which have little need of concern for human issues. By considering the reality that Internet services are provided by service systems composed of machines, people, and processes, we show that there are important, albeit different, human aspects both in the case of computer users and human users.

We propose the use of Service Science as a reference framework to address the design, evaluation, and deployment of online services. In our experience, designers and engineers of online services are mostly unaware of Service Science concepts, so it is important to create mechanisms to allow the use of techniques developed for traditional services in the realm of online services. We have exemplified in the paper how this merge of well-established concepts 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.

We have shown 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 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.

We believe that there is an enormous opportunity to establish a framework for the design of human interfaces for online services based on Service Science. In particular, we hope that this introductory discussion creates questioning and curiosity in the field and will trigger new and more research.

Of course our work can be further expanded by considering more complex situations of Internet services such as when the boundaries between users and providers blur, especially in the case of service systems which rely on user-created content (for example, *ebay* or *Wikipedia*); or in *mashups* where information from multiple service systems are integrated by a service provider. But even in such new situations we can get some insights from Service Science: in the former case, by looking into traditional markets and how they function; and in the latter, by examining multi-provider service systems such as stadiums where often the many different services (security, food, cleaning, telecommunications) are provided by a myriad of companies.

Our experience with large scale IT service systems has shown us that in the case of services provided by service systems to machines there is also a need for better interfaces to the computer user applications than we have today. In particular, we argue that the best option is to create backdoor human interfaces for the people doing the maintenance and control of the service system. This is hardly considered or developed in today's computer applications and we hope that our argumentation stimulates the beginning of a discussion about the best ways to implement such interfaces.

We are, of course, not downplaying the need of more research and development of the computer-computer aspects of Internet services, and even the centrality of those issues in the area. Our main point here was to demonstrate that there are human aspects associated with Internet services which have mostly been ignored both by the distributed systems and networking communities, and the HCI researchers. In fact, we hope to have demonstrated that there is a great opportunity for innovative and groundbreaking research on the human aspects of Internet services.

References

1. Papazoglou MP, van den Heuvel W-J (2005) Web Services Management: A Survey. In: IEEE Internet computing, pp 58–64
2. Papazoglou MP, Georgakopoulos D (2003) Service-oriented computing. Commun ACM 46(10):25–28
3. Pinhanez C (2009) "Humans inside" as the key characteristic of service systems. In: QUIS 11—Moving Forward with Service Quality. Wolfsburg, Germany
4. Lovelock CH, Wirtz J (2004) Services Marketing: People, Technology, Strategy, 5nd edn. Pearson/Prentice Hall, Upper Saddle River, xviii, 652 p
5. Fitzsimmons JA, Fitzsimmons MJ (2004) Service management: operations, strategy, and information technology. McGraw-Hill/Irwin, Boston
6. Zeithaml VA, Bitner MJ, Gremler DD (2006) Services marketing: integrating customer focus across the firm, 4th edn. McGraw-Hill/Irwin, Boston, xxvii, 708 p
7. Spohrer J, Riecken D (eds) (2006) Commun ACM 49(7):30–32. Special issue on services science
8. Spohrer J et al (2007) Steps toward a science of service systems. Computer 40(1):71–77
9. Ryan G, Valverde M (2006) Waiting in line for online services: a qualitative study of the user's perspective. Inf Syst J 16(2):181–211
10. Loiacono ET, Watson RT, Goodhue DL (2007). WebQual: An instrument for consumer evaluations of web sites. Int J Electron Commer (to appear)
11. Barnes SJ, Vidgen RT (2000) WebQual: an exploration of website quality. In: Proceedings of the 8th European conference on information systems, trends in information and communication systems for the 21st century, ECIS 2000, Vienna, Austria

12. Webb HW, Webb LA (2004) SiteQual: an integrated measure of web site quality. J Enterp Inf Manag 17(6):430–440
13. Szymanski DM, Hiseb RT (2000) E-satisfaction: an initial examination. J Retail 76(3):309–322
14. Wolfinbarger M, Gilly MC (2003) eTailQ: dimensionalizing, measuring and predicting retail quality. J Retail 79(3):183–198
15. Yoo B, Donthu N (2001) Developing a scale to measure the perceived quality of an internet shopping site (SITEQUAL). Q J Electron Commer 2(1):31–47
16. Parasuraman A, Zeithaml VA, Malhotra A (2005) E-S-QUAL: a multiple-item scale for assessing electronic service quality. J Serv Res 7(3):213–233
17. Dhillon BS (2005) Reliability, quality, and safety for engineers. CRC Press, Boca Raton, p 219
18. Sampson SE, Froehle CM (2006) Foundations and implications of a proposed unified services theory. Prod Oper Manag 15(2):329–343
19. Pinhanez C (2009) A service science perspective on human-computer interface issues of online service applications. Int J Inf Syst Serv Sector 1(2):17–35
20. Nielsen J (2000) Designing web usability: the practice of simplicity. New Riders Publishing, Indianopolis, 420 p
21. Nah FF-H, Davis S (2002) HCI research issues in electronic commerce. J Electron Commer Res 3(3):8–113
22. Voss C (2003) Rethinking paradigms of service: service in a virtual environment. Int J Oper Prod Manag 23(1): 88–104
23. Sampson SE (2001) Understanding service businesses, 2nd edn. Wiley, New York, 512 p
24. Pinhanez C (2008) Service systems as customer-intensive systems and its implications for service science and engineering. In: Proc. of Hawaiian international conference on system sciences, HICSS-41, Big Island, Hawaii
25. Wang YD, Emurian HH (2005) An overview of online trust: concepts, elements, and implications. Comput Hum Behav 21:105–125
26. Fogg BJ et al (2001) What makes web sites credible? A report on a large quantitative study. In: Proc. of CHI'01, Seattle, Washington
27. Brodie C, Karat C-M, Karat J (2004) Creating an e-commerce environment where consumers are willing to share personal information. In: Karat C-M, Blom JO, Karat J (eds) Designing personalized user experiences in eCommerce. Kluwer Academic, Dordrecht, pp 185–206
28. Shneiderman B (2000) Designing trust into online experiences. Commun ACM 43(12):57–59
29. Featherman MS, Valacich JS, Wells JD (2006) Is that authentic or artificial? Understanding consumer perceptions of risk in e-service encounters. Information Systems Journal 16(2):107–134
30. Karat J, Karat C-M, Brodie C (2004) Personalizing interaction: directions for HCI research. In: Karat C-M, Blom JO, Karat J (eds) Designing personalized user experiences in ecommerce. Kluwer Academic, Dordrecht, pp 7–17
31. Karat C-M, Blom JO, Karat J (eds.) (2004) Designing personalized user experiences in eCommerce. Kluwer Academic, Dordrecht, xii, 348 p
32. Oliva R, Sterman JD (2001) Cutting corners and working overtime: quality erosion in the service industry. Manag Sci 47(7):894–914
33. Schneider B, White SS (2004) Service quality: research perspectives. Foundations for organizational science. Sage, Thousand Oaks, xiii, 185 p
34. Parasuraman A, Zeithaml VA, Berry LL (1985) A conceptual model of service quality and its implications for future research. J Mark 49:41–50
35. Zeithaml VA, Parasuraman A, Malhotra A (2002) Service quality delivery through web sites: a critical review of extant knowledge. J Acad Mark Sci 30(4):362–375
36. Trocchia PJ, Janda S (2003) How do consumers evaluate internet retail service quality? J Serv Mark 17(3):243–253
37. Bouch A, Kuchinsky A, Bhatti N (2000) Quality is in the eye of the beholder: meeting users' requirements for internet quality of service. In: Proc. of CHI'00. The Hague, the Netherlands.
38. Walker J (1990) Through the looking glass. In: Laurel B (ed) The art of human-computer interface design. Addison-Wesley, Reading
39. Norman DA (1988) The psychology of everyday things. Basic Books, New York
40. Shneiderman B (1987) Designing the user interface: strategies for effective human-computer interaction. Addison-Wesley, Reading
41. Laurel B (1991) Computers as theatre. Addison-Wesley, Reading
42. Bohra A et al (2004) Remote repair of operating system state using backdoors. In: Proc. of international conference on autonomic computing 2004. New York, NY
43. Sultan F et al (2005) Recovering internet service sessions from operating system failures. IEEE Internet Comput 2:27
44. Dhillon BS (1999) Design reliability: fundamentals and applications. CRC Press, Boca Raton, 396 p
45. Dhillon BS (2002) Engineering maintenance: a modern approach. CRC Press, Boca Raton, 224 p