EARL J. WAGNER AND HENRY LIEBERMAN

# PERSONALIZED PRESENTATION OF POLICIES AND PROCESSES

## 1. INTRODUCTION

The web enables an organization to offer a unique interface to each customer using its web site. Personalization in e-commerce has traditionally focused on marketing products and services. By tracking a customer's purchases, software for personalization precisely identifies the customer's segment and offers narrowly focused product suggestions, through recommender systems, for instance. But what happens after placing the original order? A customer may have a problem with an ordered item and want to exchange it. Another customer may see an incorrect stock transfer and want to know what happened. In cases like these, there are further opportunities for personalized support beyond the original order.

## 2. CURRENT TECHNOLOGIES FOR CUSTOMER SERVICE

There's an important difference between the customer's situation before and after originally placing an order. Afterwards, the customer has already performed an action, such as placing an order, and initiated a process, such as a purchase process. Further interactions with the organization occur within the context of this ongoing process, the overall purchase. A customer may want to switch a plane ticket depending upon the airline's policy but, in order to know what her options are, she must match the abstract policy description on the airline's web site with the concrete data of his purchase. Another individual may notice incorrect information about his status on his organization's intranet. To find out what happened, he may need to visit multiple pages to collect data about the information, and examine the policies of his organization to determine how they were applied in order to understand the big picture. A recent study by Jupiter Research noted that the more complex a product or service is, the more



*[Editor(s) here] (ed.), [Book Titlen here],* 1—21.

customers will value good support for it, and travel and financial services are among the most complex (Daniels, et. al. 2003). When something goes wrong, as in these cases, and customers' initial attempts to diagnose the problem fail, they contact an organization's customer support.

Even when it is effective, however, good customer support has problems. The customer has to spend time waiting on hold, provide identifying information, explain the problem (often to multiple people), and work with a customer support representative (CSR) to trouble-shoot the problem or even just discover his options for future action. The organization has to dedicate resources on a service that customers are still often dissatisfied with (cite). Worst of all is when a customer and CSR spend time trying to diagnose a problem and discover that it was caused by a third-party. Then the customer has to start the process of diagnosis again with another organization.

A perfect interaction with customer support by phone still lacks the benefits of a web-based interface. In order to resolve a problem, a customer must be sure to call during the hours support is offered, and have a block of uninterrupted time. The web, on the other hand, is always available and if some information is not immediately at hand, a decision can be postponed. Furthermore, it is often easier to understand complex information and learn procedures when they are presented graphically as well as with text (cite - Richard Mayer). All of the advantages of a slick web site's presentation are lost when the customer has to hear his options or receive instructions over the phone. In fact, all of the advantages of having information in a digital format are lost. A customer has to be sure to write down choices or steps, rather than be able to see them, cut-and-paste them to a to-do list and so on, before possibly printing them out.

Organizations have put effort into developing sophisticated web sites for selling products and services. Recognizing the benefits of offering better support (and the associated opportunities for cross-selling and up-selling), many organizations have also invested in customer relationship management (CRM) systems for integrating and managing a customer's data. However, customer-facing interfaces for help remain fundamentally unchanged. A new trend toward “customer self-service” involves enabling customers to diagnose and resolve problems on their own by providing more sophisticated information and functionality on the web site. Because of the concept's appeal, it is expected that by 2005, more than 70% of customer service interactions for information will be automated (Kolsky 2002a). However, there is still a long way to go before the potential for self-service is realized. When facilities for self-service are offered, 46% of those who use them reported needing to complete inquiries via telephone, and only 26% found the information they sought (Kolsky, 2002b). Current tools for self-service still work at a somewhat superficial level and typically involve providing access to a knowledge-base of information via a search engine and interfaces for billing and account applications. We can see the problem more clearly by focusing on these two technologies –a knowledge-base and applications that both reach deep into the organization.

In using a knowledge-base, the customer must examine information and determine whether it applies to the problem at hand. Why can't a customer simply ask "which of these apply to me?" The organization manages customer information that is not typically used to tailor the information provided to customers. For instance, whom the customer has to contact to modify a purchase that has already been made is only applicable after the customer makes a purchase. Information about making a new purchase is not helpful when a customer is trying to resolve a problem with the current one. It's more helpful for a customer to see how an organization's policies apply to his case than to some arbitrary example. All of this is common-sense to people like customers and CSRs, but none of it is used to make the organization's web site help more effective.

The other important component of self-service is access to applications. A recent study found that three of the four most important features that customer look for in an online bank are self-service applications including online bank statements, transaction histories of 2 years or more, and the ability to change a billing address or stop payments on checks (Dhinsa, 2003). The study concluded that these services would maximize "real cost savings through contact-center productivity" and "provide experiences that deepen customer relationships over time".

An under-explored possibility exists for even more advanced self-service. When performing an action such as placing an order, a customer triggers a process that involves multiple parties, such as the vendor, the vendor's delivery service and the customer's bank. In a transaction like this one, the customer is the only entity that interacts with each of the others. Some of the information that customers most desire from the vendor, such as updates on delivery delays (Daniels, et. al. 2003) requires coordination among multiple sites. When sites don't have an existing relationship, important information cannot be provided. In conducting a study of web users, we found that most participants would like to have the ability to inspect a charge in their credit card transaction history to see the saved confirmation page for the purchase (Wagner, 2003). Supporting this feature, however, would depend upon either improbably close coordination among banks and organizations, or actively gathering information on the customer's end. While the customer places the order, her steps would have to be recorded and the corresponding pages saved. After the purchase, these records would have to be matched up with the appropriate charges in the user's transaction history at the user's bank. All of this would be tedious, if not overwhelming, for a person to perform. With computers, and software agents in particular, however, this tedium can be avoided. An agent, working on the user's computer, with the user's browser, could automatically monitor the user's transactions and match them to pages in which relevant data appears, such as a credit card transaction history page. By inspecting the data, such as a charge, the entire record of the transaction could be accessed. A software agent like this one could offer an even greater level of personalization than any one site could alone.

In this chapter we will discuss possibilities for personalization in these two areas of self-service, knowledge-base and other explanations of an organization's policies, and integrating information about a customer's accounts across web sites. In the next section, we will see what both of these possibilities depend upon, a formal representation of an organization's policies and processes.

## 3. PERSONALIZATION THROUGH FORMAL PROCESSES

With the promise of automated interactions via web-services, languages for business processes such as the Business Process Execution Language for Web Services (BPELWS) are increasingly being standardized and are expected to be supported by software vendors. Formal descriptions of an organization's business policies and processes, at least the ones a customer interacts with, create new opportunities for help. Combining them with a customer's data enables the dynamic generation of explanations of policies in which the examples themselves demonstrate a customer's data and the options he has. Alternately, by sharing these descriptions, the organization could enable its customers to compile the descriptions from multiple businesses and see the "big picture" for a process, even when it spans multiple sites. A customer could see the complete history of a purchase or other transaction and never have to bother with contacting each organization individually to assemble the record of the history. Of course, customers themselves are not likely to retrieve these descriptions themselves. Instead, we argue that a software agent, working with the web browser could retrieve this information and help customers in diagnosing problems by themselves.

In the next section we will see how an organization's policies can be combined with a customer's data to more effectively explain the customer's options and even allow the customer to consider hypothetical cases that are difficult with current technology. We will then discuss an agent we've developed, Woodstein, that works with a user's browser to explain not just an organization's policies, but also the overall process a customer is involved in, even when it spans multiple sites.

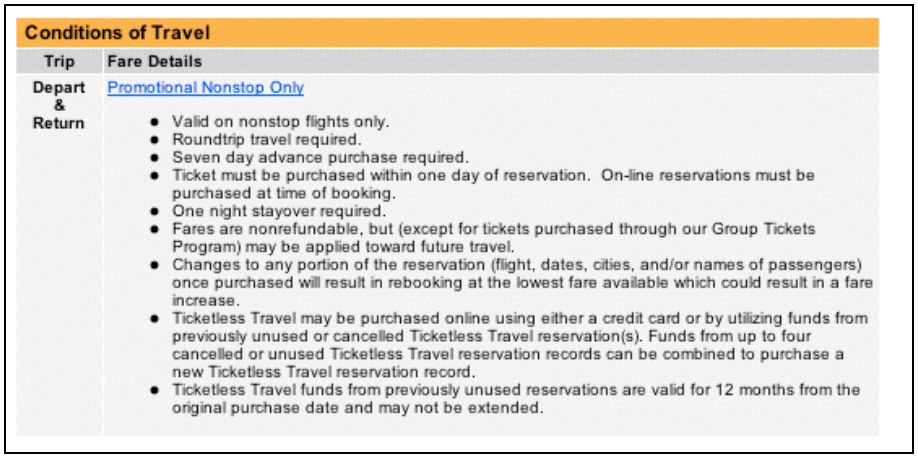
## 4. PERSONALIZED EXPLANATION OF POLICIES

Consider a situation recently faced by one of the authors. In planning a trip, he went to a discount airline's web site and ordered the cheapest ticket they offered for the dates he wanted to fly. A few days later, he realized that he'd have to move the departure date to a day earlier. He wanted to know whether changing the ticket was permitted, and what the relevant costs or penalties would be. None of this information presented on his reservations page appeared relevant, however. He looked through the rest of the web site and found a policy web page that, if printed out, would be about twenty pages long. It described all possible options for all ticket types, but because he wasn't familiar with the

details of his own ticket's category, he was still lost. He ended up calling customer support and asking for help. After waiting on hold, then explaining the situation to the CSR, he learned that he could change the ticket by paying an extra fee.

The customer would have preferred to quickly resolve the problem on the web, but had to resort to calling customer support for help. The most immediate problem was that the reservations page didn't explain his options involving the ticket. However, this page was designed for presenting just the information about his reservations in as simple a way as possible. Presenting all possible policies for modifying reservations would be too cumbersome. On the other hand, the policy page was itself too unwieldy and it wasn't clear which policies were relevant. The situation might have been improved by including some information about the reservation options. However, we see the problem more broadly.

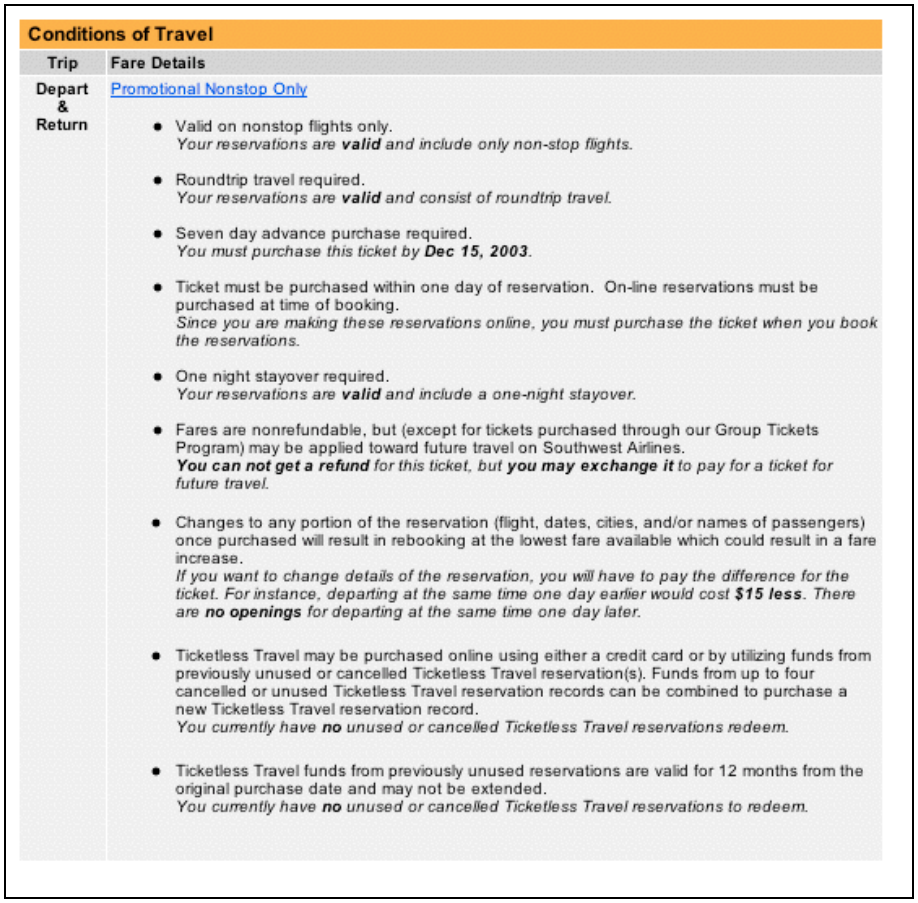
The essential problem is that abstract policies are difficult to understand. The customer has access to and knows about his data. But it is often not clear how customer data relates to the vendor's policies. What does it mean in relation to those policies? What options does the customer have? Policies are typically phrased as “When the customer purchased...” and “The customer has 14 days after...” Understanding the policies requires translating them into the customer's terms: “When you purchased...” and “You have until Thursday, May 29 to...” Figure 1 shows the set of policies for the customer's ticket class. Figure 2 shows the ticket policies supplemented with explanations computed from data about customer including information about his ticket. With an interface like this one, he could not only better understand the particular ticket he selected, but also compare the policies active for tickets in other classes without having to recompute all of the data, such as dates or prices. Today, customers either piece together these explanations themselves, or receive them through customer support. Support is expensive for the vendor, however, and time-consuming for the customer.



**Conditions of Travel**

| Trip | Fare Details |
|---|---|
| Depart & Return | Promotional Nonstop Only<br>• Valid on nonstop flights only.<br>• Roundtrip travel required.<br>• Seven day advance purchase required.<br>• Ticket must be purchased within one day of reservation. On-line reservations must be purchased at time of booking.<br>• One night stayover required.<br>• Fares are nonrefundable, but (except for tickets purchased through our Group Tickets Program) may be applied toward future travel.<br>• Changes to any portion of the reservation (flight, dates, cities, and/or names of passengers) once purchased will result in rebooking at the lowest fare available which could result in a fare increase.<br>• Ticketless Travel may be purchased online using either a credit card or by utilizing funds from previously unused or cancelled Ticketless Travel reservation(s). Funds from up to four cancelled or unused Ticketless Travel reservation records can be combined to purchase a new Ticketless Travel reservation record.<br>• Ticketless Travel funds from previously unused reservations are valid for 12 months from the original purchase date and may not be extended. |

*Figure 1. Policies without Explanations*



**Conditions of Travel**

| Trip | Fare Details |
|---|---|
| Depart & Return | Promotional Nonstop Only |

- Valid on nonstop flights only.
  *Your reservations are **valid** and include only non-stop flights.*
- Roundtrip travel required.
  *Your reservations are **valid** and consist of roundtrip travel.*
- Seven day advance purchase required.
  *You must purchase this ticket by **Dec 15, 2003**.*
- Ticket must be purchased within one day of reservation. On-line reservations must be purchased at time of booking.
  *Since you are making these reservations online, you must purchase the ticket when you book the reservations.*
- One night stayover required.
  *Your reservations are **valid** and include a one-night stayover.*
- Fares are nonrefundable, but (except for tickets purchased through our Group Tickets Program) may be applied toward future travel on Southwest Airlines.
  ***You can not get a refund** for this ticket, but **you may exchange it** to pay for a ticket for future travel.*
- Changes to any portion of the reservation (flight, dates, cities, and/or names of passengers) once purchased will result in rebooking at the lowest fare available which could result in a fare increase.
  *If you want to change details of the reservation, you will have to pay the difference for the ticket. For instance, departing at the same time one day earlier would cost **$15 less**. There are **no openings** for departing at the same time one day later.*
- Ticketless Travel may be purchased online using either a credit card or by utilizing funds from previously unused or cancelled Ticketless Travel reservation(s). Funds from up to four cancelled or unused Ticketless Travel reservation records can be combined to purchase a new Ticketless Travel reservation record.
  *You currently have **no** unused or cancelled Ticketless Travel reservations redeem.*
- Ticketless Travel funds from previously unused reservations are valid for 12 months from the original purchase date and may not be extended.
  *You currently have **no** unused or cancelled Ticketless Travel reservations to redeem.*

*Figure 2. Policies with Explanations*

Helping people learn about abstract principles has been the focus of work on intelligent tutoring systems (ITSs) (cite) at the intersection of research in artificial intelligence (AI) and education. An ITS explains principles in an area like algebra or physics, then provides the student with problems to solve. It keeps track of information about the student, including what the student has learned and the principles the student finds difficult in order to select problems that best challenge the student and further develop his understanding of the area.

Unlike the learning in an academic environment, however, the type of learning that a customer is engaged in is extremely pragmatic and focused entirely on the problem at

hand - completely unlike the situation of a student studying algebra, for instance. A customer is interested in reaching his goals primarily and not in learning about the principles, in this case, the organization's abstract policies. Yet this is exactly what many organizations' web-sites require. A customer has to find his concrete data, his ticket type, and match it with an abstract policy of the organization, its rule on exchanging tickets, in order to understand his options for reaching his goal of changing his ticket.

Formal descriptions of policies offer the possibility of dynamically generated explanations like the ones above. Today, we have dynamically-generated data pages, such as the page the customer saw explaining his flight reservations. In the future, we may have dynamically-generated policy pages, in which policies are explained with examples featuring a customer's up-to-date data. Customers could use these examples to explore hypothetical possibilities such as “which ticket would I have to buy in order to be able to change the dates without penalties?”, and later “how do I change the departure date for my ticket”, or more generally, “what are the policies that are active for the ticket I bought?”

Within the vendor's systems, processes result from interacting policies. For processes as well as policies, customers seek relevant information: “what happened in my case?”, “how did my account get this status?”, “what's going on here?”. Customers can benefit from explanations of these processes, even when they don't have experience with formal representations of processes. Instead of interacting with these representations directly, they may turn to a software agent playing a role like that of a human agent that explains how a customer's data interacts with a vendor's policies and processes.

## 5. PERSONALIZED PRESENTATION OF PROCESSES WITH WOODSTEIN

Woodstein is a software agent that works with a user's web browser to answer questions like “How did that data get that value?” “Why did that happen” and “What's happening now?”. It monitors a user's actions on the web, such as browsing an online retailer and adding items to a shopping cart, to create a record of the user's overall process, in this case, making a purchase. It is then able to answer questions about the history and current status of the process, as well as how data in the process was set.

Woodstein matches a user's actions to the steps of an abstract model for the process. Through this process of recognition, it knows to look for more information about the process on other web pages and web sites, even if user never visited them. By seeing the user select a credit card and shipper for a purchase, Woodstein knows to go to the sites of the bank and shipper for more information about the status of the purchase, including whether it has been paid for and delivered.

Later, when the user is looking at other pages with data about the process, such as the credit card transactions history page, the charge itself can be inspected. The history of the purchase process can be retrieved and reviewed, making it convenient to understand the context of the data, the charge.

Woodstein explains the history of the data and actions of the user's process through multiple views. As a user is performing a step of an action, it can show where that step fits into the overall process. Another view shows the history of a process and allows the user to revisit any point. When inspecting a data item, a view displays an automatically generated audit trail it, enabling the user to jump among pages in which it appears. The user can jump from the charge amount in the transactions page to a saved copy of the order confirmation page in which the amount appears.

In addition to providing views on the history of processes and data, Woodstein also provides help in diagnosing problems when something goes wrong. When a user notices that something doesn't "look right", simply clicking on a menu item tells the agent. Woodstein responds by identifying processes or data that could have contributed to the error or unexpected result. Through this assistance, the user is able to identify the exact step or data that either caused an error, or created an unexpected result.

Woodstein's features can best be understood through an example. We will see how Woodstein:

- supports inspection of information in pages, when the user want to know more about data and processes included within them
- explains the history of processes and data, through easily-understood views that visualize their relationships
- shows all pages related to a user action, including related pages that Woodstein retrieves that include more information

Problems arise because of mistakes by either the user or the web site, or because the user has an incorrect mental model of the process. It is important that the agent provides help in all cases, because when a problem symptom is found, it is not known what the source of the problem is or whether there even is a problem.

## 6. INSPECTING A PROCESS WITH WOODSTEIN

In this example we will see the support that Woodstein provides to an individual accessing information on his organization's intranet. Although it could apply equally well to a company employee examining his health benefits, or his stock options plan, we will focus on a masters' student who is trying to graduate from the institute of higher education which he attends. He was sure to add himself to the list of graduating students earlier in the semester and he knows he's completed all other requirements for graduation. He knows that sometimes there are bureaucratic mistakes, however, so he goes online and checks the student information web site to verify that everything is OK. Upon loading the graduated degree audit page, however, he sees that everything is not OK; it appears he's not eligible to graduate (Figure 3).

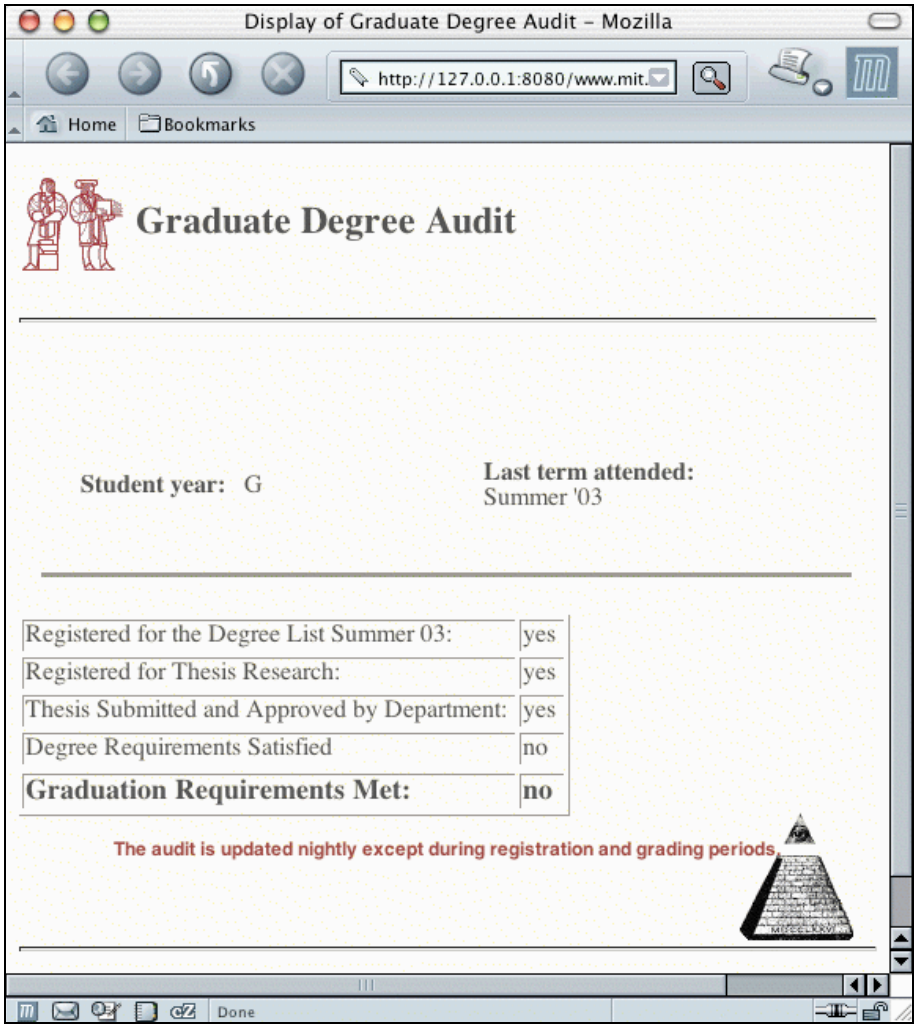



*Figure 3. Viewing the graduation degree audit page*

The user would like to know why he's not eligible. Specifically, he'd like to know which requirement in particular was not satisfied and caused him to become ineligible. It looks like his degree requirements weren't satisfied, but that too is pretty general and he'd like to narrow down the problem even more. Without Woodstein, he'd have to contact the institute's administration then perhaps be told that he'd have to talk to his

academic department, or some other unit for more information. With Woodstein, he is able to interact with the data directly. While the user browses, Woodstein adds its icon to every page he sees. Now the user wants to inspect the graduation eligibility data directly, so he turns on Woodstein's inspector by clicking on the icon (Figure 4)

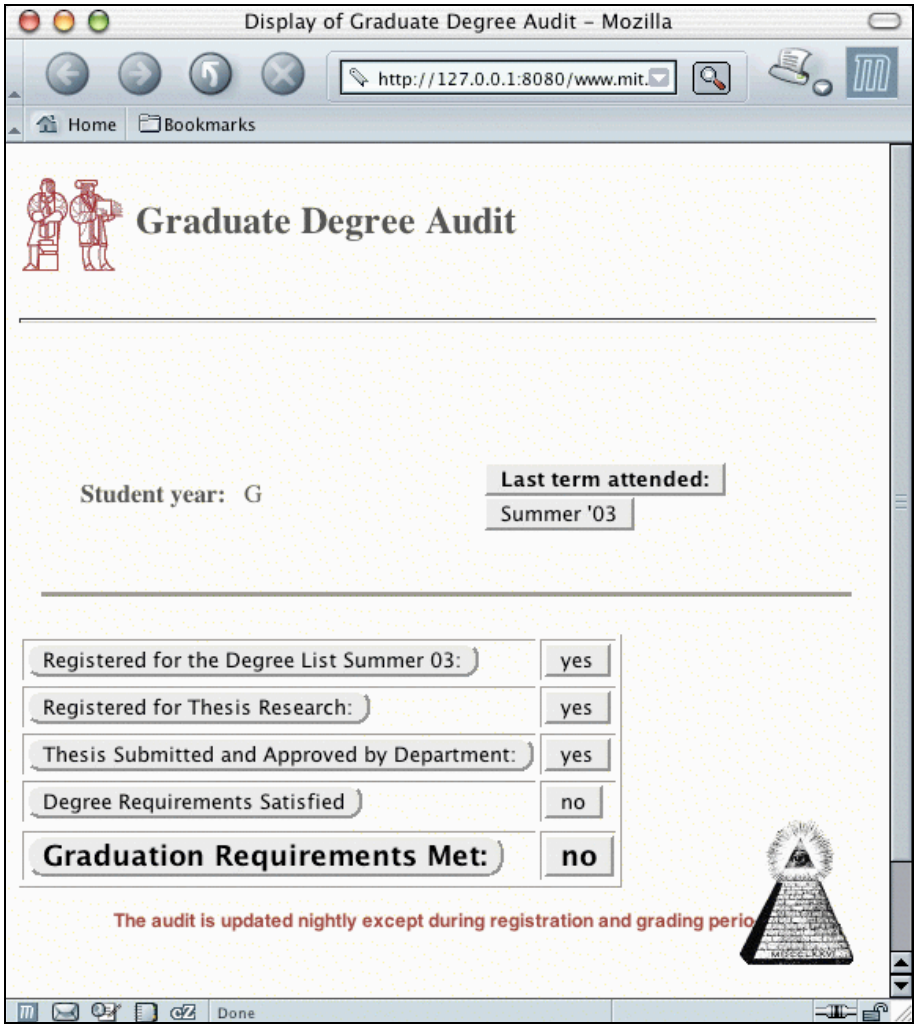



*Figure 4. Inspecting the graduation degree audit page with Woodstein*

Woodstein converts the text of each data item in the page that it is aware of to a button. The user moves the cursor to the "no" button to the right of "Graduation Requirements Met" and presses down, causing its menu to appear (Figure 5).

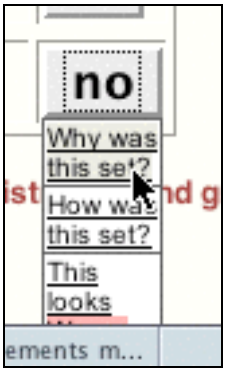



*Figure 5. Inspecting the "graduation requirements met: no" data item*

The student wants to know why his graduation requirements were not met, or in other words, why the "graduation requirements met" data item came be to be set to "no", so he selects "why was this set?". Woodstein opens a new window to show the history of how the data item was set (Figure 6).

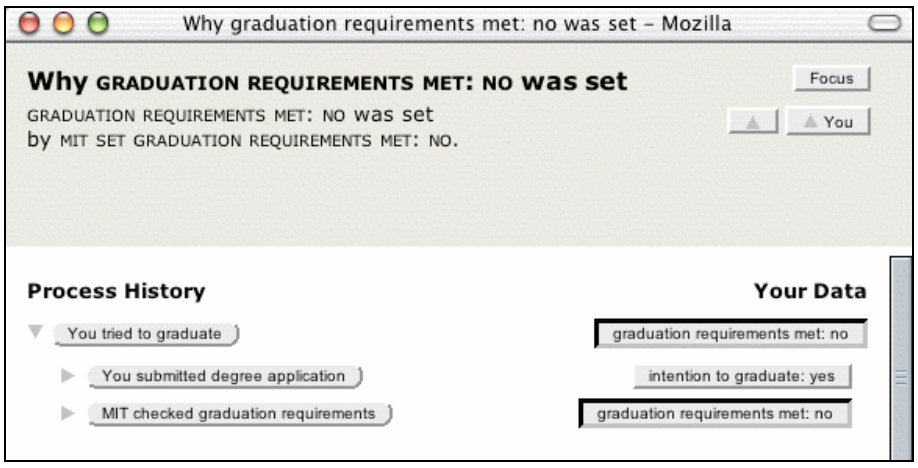



*Figure 6. Viewing why "graduation requirements met" was set to "no"*

This new window is Woodstein's "process history" view. It shows the history of the student's attempt to graduate with English descriptions of the processes and data involved. The top frame, in grey, explains some of the context of the data item he inspected. It was set as the result of the process: "Institute set graduation requirements". Below the top frame is a tree with the structured history showing the student's steps in graduating and the Institute's response. On the left are the steps of the process, while on right is the student's data resulting from the processes.

The student originally indicated to his institute that he intended to graduate by submitted an application for his advanced degree. Woodstein began creating this record after recognizing that action. It retrieved information about the computation of his graduation eligibility from both this site and other sites on the institute's intranet. Now the student is able to view the record to see what went wrong.

Woodstein manages "representations", or "reps" for both data and processes on the web. Data items include prices, quantities, and, as in this case, boolean values such as "yes" or "no". Processes have data items as inputs and set a single data item as a result. Whenever a data item appears in a page, as with the data item the student selected, Woodstein converts the actual data value, "no", to a rectangular button corresponding to the data and the label "graduation requirements met:" to a rounded button for the process that set the data item. Processes and data appear as buttons in Woodstein's views and any button can be inspected with by pressing down to choose from its menu, or just clicking on it to select it. Since the student just interacted with the "graduation requirements met: no" data item, its button appears pressed in while other buttons do not. This data item is the "selected rep" which the view describes.

In looking at the process history tree, the student sees that he tried to graduate, but was unable. He sees that the first step of this process was that he submitted his degree application, which he can see was successful because Woodstein found that the site recognized his intention to graduate. The next step, his Institute's response, was not successful however. To find out why, he clicks on the triangle on the left of the process to open it and see its individual steps (Figure 7).

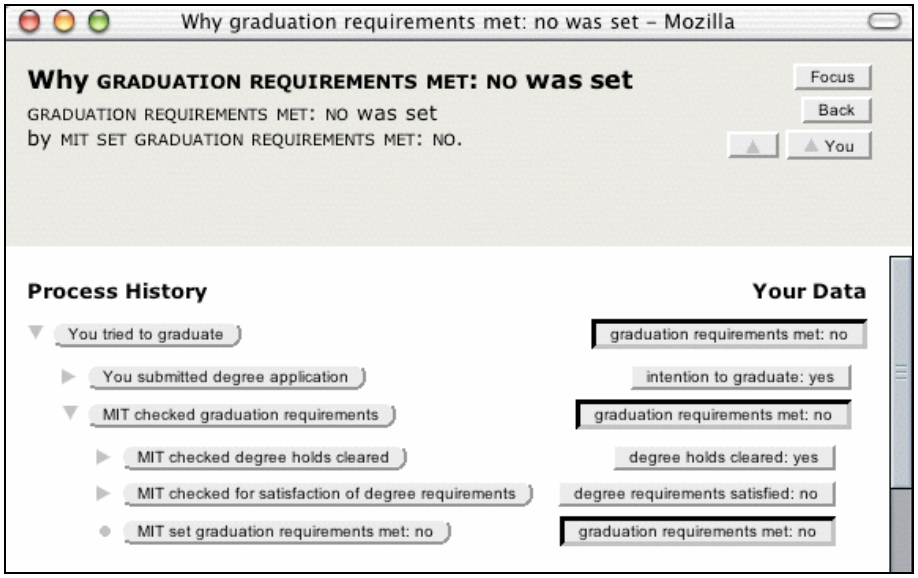



*Figure 7. Looking at the graduation requirements*

He can see that the institute checked whether any holds on his degree were cleared, and finds that they were. He also sees that his degree requirements were checked but that those requirements were not met. Because they were not met, his overall graduation requirements were set to not be met. He clicks on the triangle to open the process of checking his degree requirements to find out more (Figure 8).

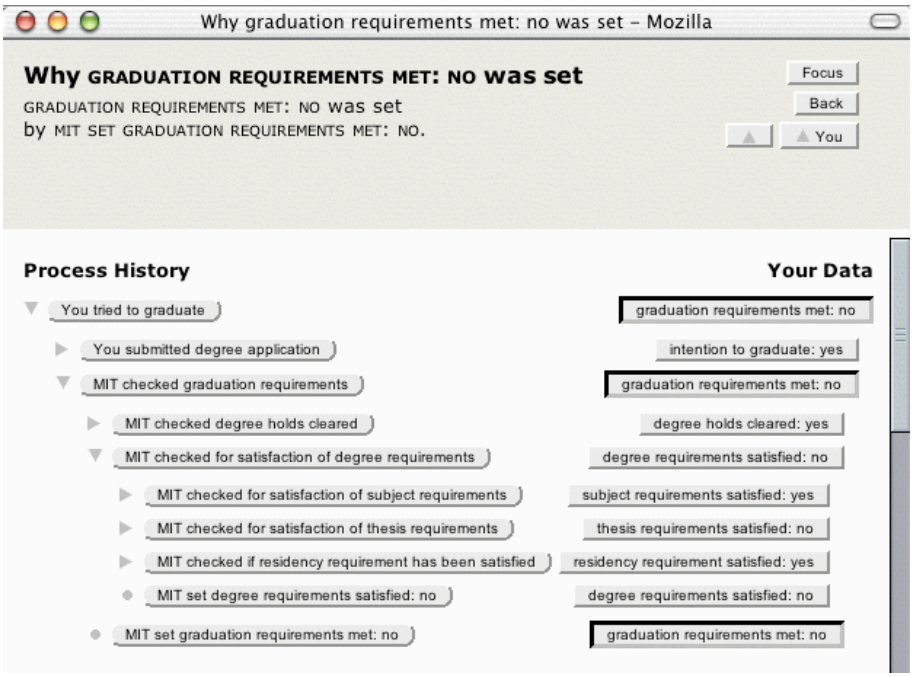



*Figure 8. Looking at the degree requirements*

He sees that although his subject requirements were satisfied, and his residency requirement was satisfied, his thesis requirements were not satisfied, so he opens the corresponding process. "Drilling-down" even further, he discovers that his thesis title had not been provided (Figure 9).

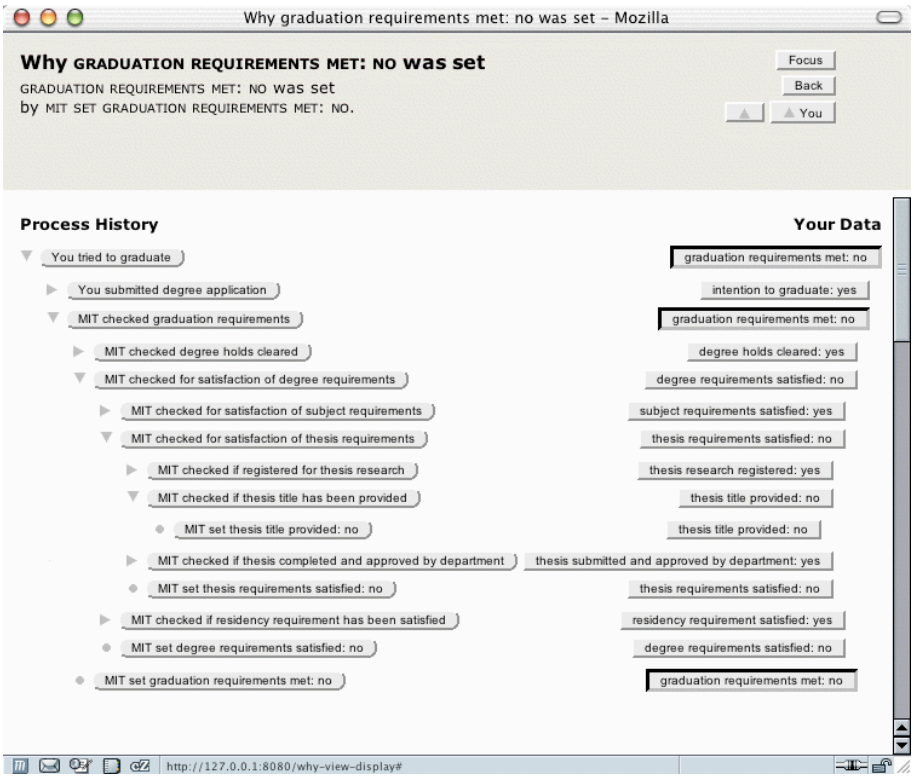



*Figure 9. Finding the unmet requirement*

Clicking on a rep's button causes Woodstein to load its saved page for the rep. The student clicks on “Institute checked if thesis title has been provided” to see the page Woodstein found that describes the rule (Figure 10).

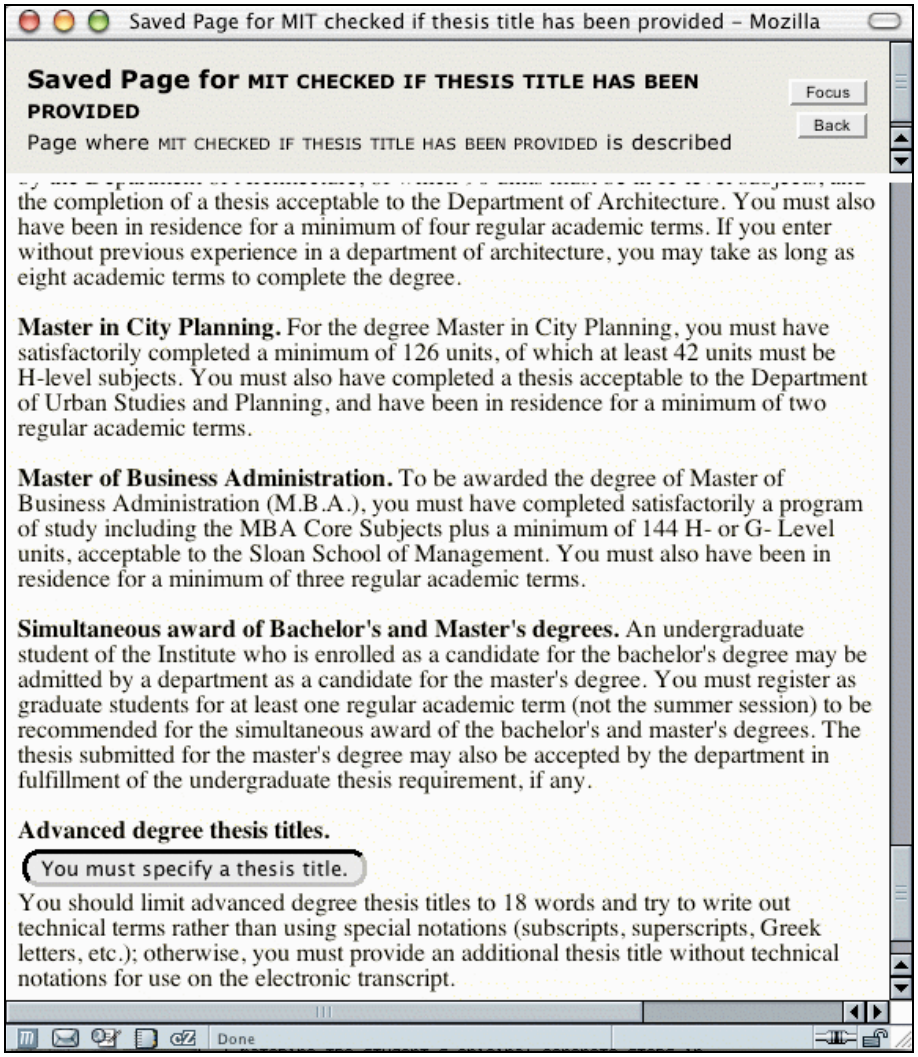



*Figure 10. Viewing the requirement that a thesis title be specified*

The student knows that he satisfied all of the details of this requirement, so he goes back to process history view and selects the "thesis title provided: no" data item. Its saved page is the page where it first appeared (Figure 11).

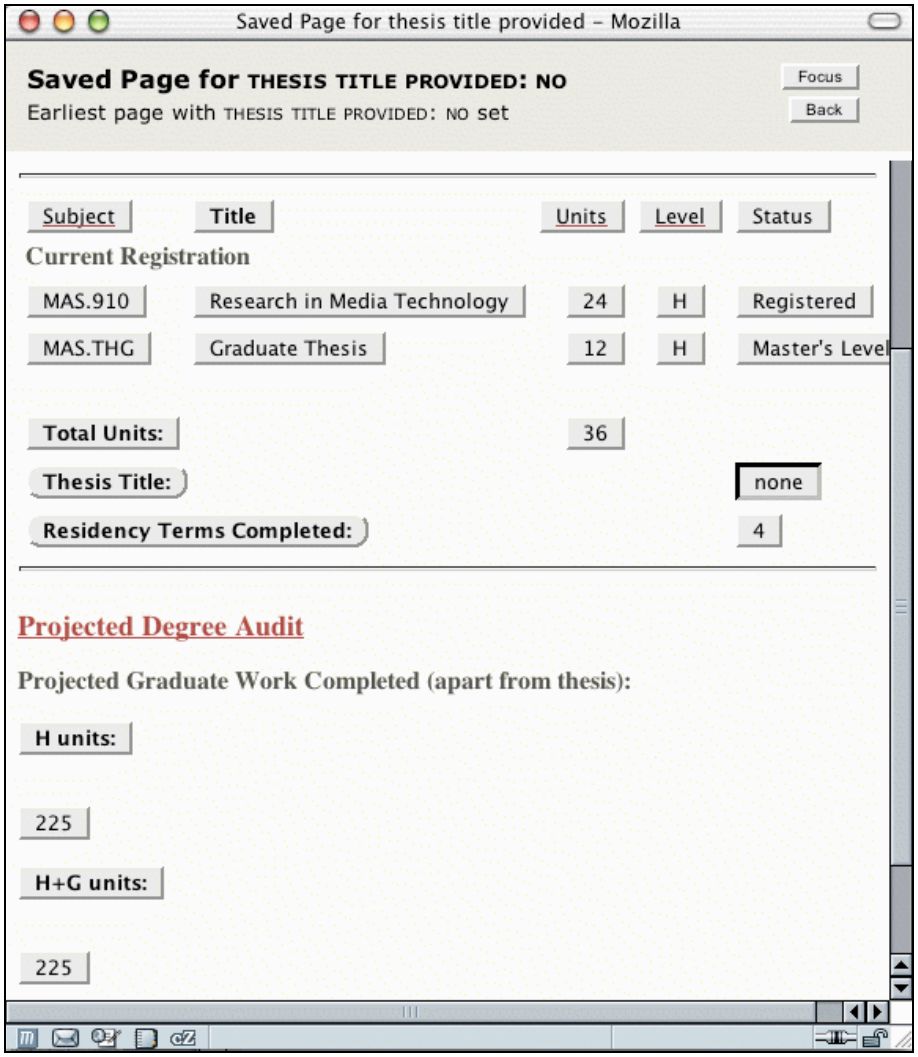



*Figure 11. Viewing the page with the incorrect requirement*

The student knows that there is some problem, but this is as far as he can go. At this point, he wants to complain about the data item, so he presses down on it to bring up its menu and he selects "This looks Wrong" (Figure 12). Woodstein creates a new mail window with an automatically generated complaint indicating the data item that looks

wrong and the page that it appears on, as well as the student's path in finding it (Figure 13). The student could customize this email before sending it and remind them about his thesis title, for instance.

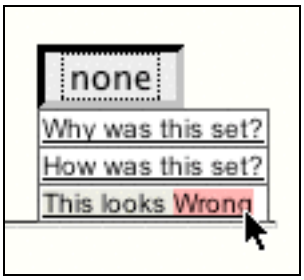



*Figure 12. Noting that the requirement looks incorrect*

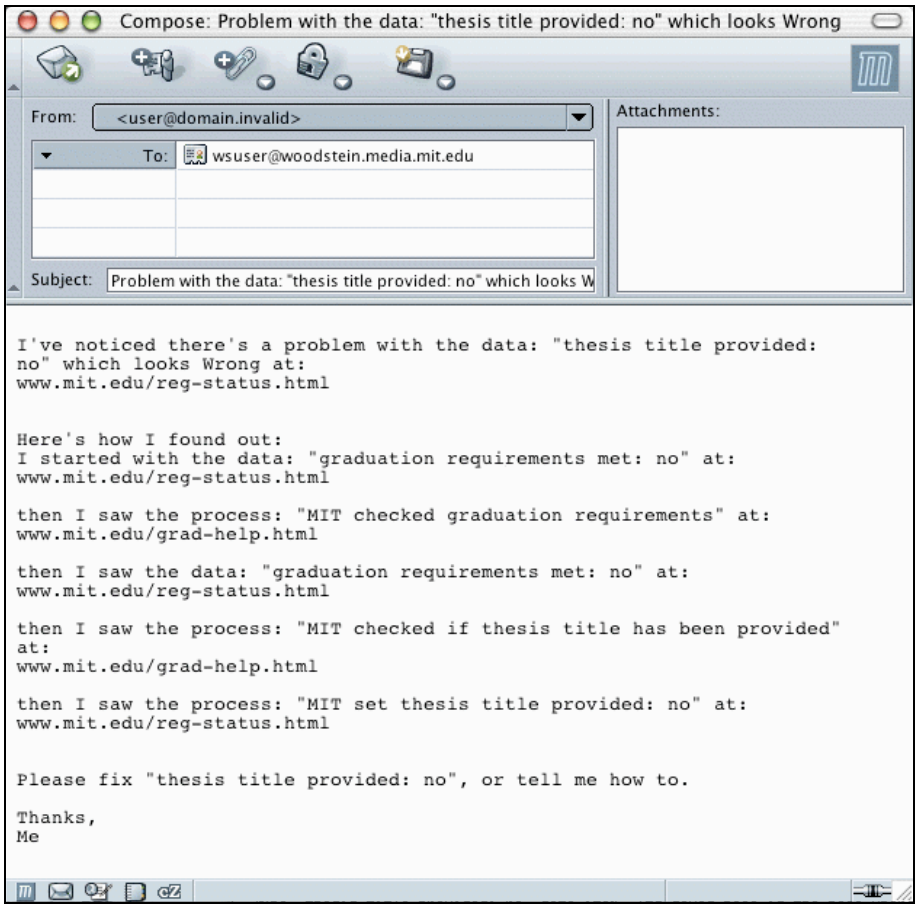



*Figure 13. Complaining about the incorrect requirement*

In this scenario we have seen how Woodstein can help a user inspect a data item in a page to see the overall process that includes it. With the steps of the process organized hierarchically, the user is able to drill down to quickly find the individual requirement that is not met. Finally, Woodstein enabled the user to easily generate a complaint about the incorrect data item.

In some cases, as with the policies of the airline in the previous example, a single site manages all of the user's relevant data and can generate a detailed list of how its policies apply to the user. In other cases, a site computes results with a user's data from multiple

sites, in which case information must be compiled from the multiple sites. Woodstein can play an important role here by compiling this information for the user and allowing him to see the overall history for the process and how his data is involved in it. Although a site can generate explanations about a limited set of policies, by presenting the process history, Woodstein enables the user to see the processes involving any of his data. In the case of the airline tickets, Woodstein would enable the user to see the process that caused some fee to be added to his ticket as well as the page in which the fee is described, or the exact requirements of how a ticket may be exchanged.

## 7. SUMMARY

Research on personalization in e-commerce has traditionally focused on marketing products and services to customers prior to a sale. Meanwhile, online organizations are interested in supporting customers in finding solutions to their problems through self-service. In this chapter, we have argued for the personalization of techniques for self-service, both in explaining an organization's policies in terms of the customer's own data and context, and in enabling an agents working on behalf of the customer to collect and integrate information about his actions online.

## 8. BIBLIOGRAPHY

Daniels, D., Matiesanu, C. and Schatsky, D. (2003) Jupiter Consumer Survey Report: The State of Customer Service, 2003. Jupiter Research.

Dhinsa, R. (2003) Self-service: The New Category of Services to Facilitate the Adoption of Online Banking. Jupiter Research.

Kolsky, E. (2002) Is Self-Service the Panacea? Gartner, Inc.

Kolsky, E. (2002) The Six Steps for Web Self-Service in Customer Service. Gartner, Inc.

Wagner, E. (2003) Woodstein: A Web Interface Agent for Debugging E-Commerce. MIT M.S. Thesis.