Affective Web Service Design

Insu Song and Guido Governatori

School of Information Technology & Electrical Engineering
The University of Queensland, Brisbane, QLD, 4072, Australia
{insu,guido}@itee.uq.edu.au

Abstract. In this paper, we propose that, in order to improve customer satisfaction, we need to incorporate communication modes (e.g., speech act) in the current standards of web services specifications. We show that with the communication modes, we can estimate various affects on service consumers during their interactions with web services. With this information, a web-service management system can automatically prevent and compensate potential negative affects, and even take advantage of positive affects.

Keywords: E-commerce and AI Human computer interaction

1 Introduction

In this paper, we discuss an important factor with regard to creating a successful Web services user experience that has been largely ignored. We propose that, in order to counter (detect, prevent, and resolve) negative affects that can be caused by Web services, we need to reconsider some of Web standards (e.g., WSDL) to include information about communication modes of interfaces (e.g., speech acts). Communication modes are important information for checking whether or not Web services are *well-behaved*. A Web service violating the well-behavedness can cause negative affects on its users. For instance, when a directive operation is invoked by a user (e.g., 'add item A to my shopping cart'), it is a common knowledge that the user is expecting an acknowledgement within a certain time frame. A violation of this common knowledge can cause negative affects on the users.

Despite great deal of efforts on semantic web, this factor has been largely ignored. Thus, currently a large amount of development time is spent on making sure that Web services behave as intended. For example, in most e-commerce environments, the following problems frequently arise:

1. Customers often feel ignored or uncertain because prospective events, such as delivery notices, are not informed properly.
2. For customer services personals, it is hard to feel how customers are affected by the overall processes. Therefore, it is difficult to provide more adaptive and reasonable services and differences in customers' situations are often ignored.
3. Promises are often not full-filled. E-commerce Web-sites often promise customers for certain behaviors of their services in their Web site, such as promotions, but actually certain behaviors do not meet such information.

Q. Yang and G. Webb(eds)
Pacific Rim International Conference on Artificial Intelligence, PRICAI 2006
LNCS 4099, pp. 71–80, 2006.

We propose that Web services can be guaranteed to meet certain acceptable quality by incorporating affective computing, and thus avoiding these common problems.

Affective computing is not a new concept. In designing Web applications (e.g., e-commerce applications) that carry out a certain set of goals for human users, such as purchase orders, the importance of the affects of such applications has already been put forward to system designers' attention (e.g., [24,12]). Therefore, there has been much research into evaluating human emotions [11,6], expressing emotions [17,4], and the effectiveness of such approaches to improving human-computer interactions [16,5,3,25]. Business communities also have been aware of the significance of customer satisfaction in measuring business performances (e.g., American Consumer Satisfaction Index [1]) because for most companies, repeat customers are major contributors to profit [1]. Preventing negative affects on computer users is also one of the primary goals of HCI communities [20].

However, there are still no well defined languages to represent or account for affects of Web services on human users. As a result, current Web service design approaches do not have means for representing and estimating affects on users. Thus, it is impossible for Web service management systems to prevent possible negative affects or to take advantage of positive affects.

In this paper we define well-behaved protocols of Web-services based on speech act theory; and a method to evaluate various affects on the users when the Web-services violate such protocols based on cognitive appraisal theories of emotion. The rest of the paper is organized as follows. In the next section we discuss the issue of incorporating speech acts in Web service specifications. In Section 3, we define well-behavedness of Web service interfaces. In Section 4, we develop a method to evaluate affects on users during their interactions with Web services. Finally we conclude with comparisons with other approaches and some remarks.

2 Web Service Definition

Let us investigate the information that we can obtain from web service specifications in WSDL [7] which is a W3C standard for defining Web services. A WSDL document defines a set of interfaces through which consumers can interact with a Web service. Although WSDL standard defines four operation (interface) primitives that a service can support, in order to make the presentation more readable, in this paper we consider only two types of one-way operations: (One-way) the service receives a message, (Notification) the service sends a message. Since other primitive operation types defined in WSDL can be modelled abstractly using the two one-way messages, this is not a big limitation.

We represent a Web service W as a structure:

$$W = (OP_i, OP_o, M, DB, PL)$$

where OP_i is a set of input operation types, OP_o is a set of output operation types, M is a set of message types, DB is a database definition (e.g., DTD specifications for XML views of relational databases), PL is a set of plans (or procedures) each of which achieves certain goals. OP_i, OP_o, and M can be obtained directly from WSDL specifications of Web services.

Let us represent each message type in M as follows:

$$MessageTypeName(Part_1, ..., Part_n)$$

which is just an alternative representation of WSDL message definitions. We also represent each operation type in OP_i and OP_o as $OpName(m)$ where $OpName$ is the operation name of the operation and m is a message type.

Unfortunately, WSDL specifications do not tell us much about the meanings of the operations. In particular, there is no way to tell what are the speech acts of the messages exchanged between Web services.and service consumers. Without the speech act information of a message, it is impossible to know the intention of the message. Thus, we need to provide speech act information for each operations.

According to the speech act theory of Searle [26], each communication interaction is an attempt to perform an illocutionary act such as a request and an assertion. Therefore, interactions can be classified into five types (illocutionary points) of illocutionary acts, but we find that mostly only four types (directive, assertive, commissive, declarative) are used in user-service communications. We define an operation-to-speech-act mapping function as follows:

$$MSmap : O \rightarrow U \times F \times Q$$

where O is a set of operations, U is a set of users, $F = \{directive, assertive, declarative, commissive\}$ is a set of illocutionary points, Q is a set of XPath queries. A query is a specification for checking the achievement of the operation. $MSmap(op)$ returns a triple (u, f, q) for an operation instance op. We now suppose that this mapping function is given.

Example 1. Let us consider a shopping cart Web service example defined as follows:
OP_i ={AddItem(Item))} OP_o ={Response(Item))}
M={Item(name)} PL ={(AddItem(Item)→p1)}
PL has only one plan, p1, whose goal is AddItem(Item). Suppose DB is an XML view of a relational database and its schema is defined by the following DTD specification:

```
root = basket
R={basket ← customerName, basketItem*;
     basketItem ← itemName;
     customerName ← #PCDATA;
     itemName ← #PCDATA;}
```

The root item ‘basket’ represents a shopping basket of a customer in an online shopping Web service. It can have zero or more items. Now, let’s suppose the process receives the following request message from a user u through one-way operation AddItem: AddItem(Item(‘Book’)); and suppose the mapping function returns a triple (u,f,q) with the following values:

```
f= directive
q="/basket[customerName=u]/
     basketItem[itemName='Book']/itemName/text()"
```

Then, the service upon receiving the message executes plan p1 for goal AddItem(Item(‘Book’)). The plan performs some actions that will eventually lead to an update to the database so that the XPath query q will return ‘Book’.

An instance op of incoming operation can have one of the following goal types depending on the illocutionary point of the operation:

1. If f is a *directive*, the goal is op meaning that the service achieves op. For example, AddItem(Item('Book')).
2. If f is an *assertive* or *declarative*, the goal is $Bel(s, op)$ meaning that the service s believes op. For example, Details(PhoneNo('5555')).
3. If f is a *commissive*, the goal is $Bel(s, Int(u, op))$ meaning that the service s believes that the customer u intends to achieve op for the service.

where $Bel(s, p)$ means s believes p, $Int(u, p)$ means u intends p, and thus $Bel(s, Int(u, p))$ means s believes that u intends p.

For outgoing messages op, if they are commissive, the goal is just op meaning that the service promises to achieve op. Other types of outgoing messages do not create goals. They are either assertive (informing messages) or directive and treated as actions produced by services.

3 Well-Behaved Web Service Interface

With the information of illocutionary point of each message, we can now define how a Web service should interact with its users. In this paper, we consider two cases: directive one-way operations and commissive notification operations. When a user of a Web service knows that the service has received a message containing the user's goal and the service is responsible for achieving it, the user expects the service to inform the user an acknowledgement or whether the goal is achieved or not. The acknowledgement means that (a) the message is received; and (b) the goal will be achieved within a certain time limit; and (c) if the goal is achieved, the user will be informed. If the goal is a promised goal, acknowledgement is not necessary. In both cases, if the goal cannot be achieved in a certain time limit, the process must send a delay message telling the user to wait for a certain time. These are just basic protocols that are expected by most human users. Similar protocols are also defined in an agent communication language called FIPA-ACL [14].

Let us call Web service interfaces conforming to the above descriptions *well-behaved-interfaces*. Figure 1 shows state flow diagrams for well-behaved-interfaces for both a directive one-way operation and a commissive notification operation. In the figure, upon receiving a directive operation, the service must response within a certain response time rt_g with one of the following messages:

1. a confirming message $confim_g$ or
2. a disconfirming message $disconfirm_g$.or
3. a delay message $delay_g(d)$ or
4. an acknowledgement message ack_g.

If the response is either $confirm_g$ or $disconfirm_g$, the communication ends. But, if the response is $delay_g(d)$ or ack_g, the user is expecting one of the above messages (except ack_g) again within a certain goal achievement time, gt_g. If the following response is

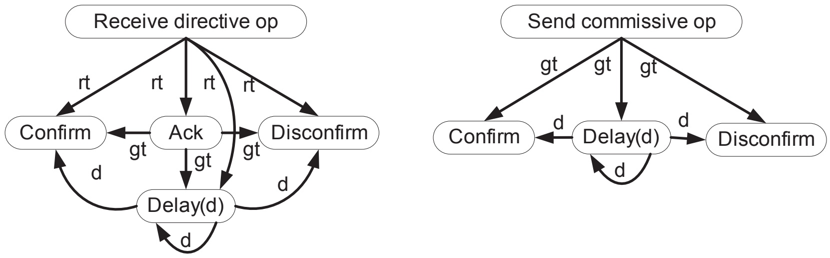

Fig. 1. Well behaved web-service interface state flow diagrams for a directive one-way operation and a commissive notification operation. It is assumed that the response time, rt, is smaller (earlier) than the goal achievement time, gt: $rt \leq gt$.

$delay_g(d)$, this time the user is expecting one of the above messages (except ack_g) within d. The response behavior for a commissive notification operation is similar to a directive operation except that there is no acknowledgement as shown in Figure 1.

4 Detecting Affects on Users

Now, we consider how human users might be affected (or felt) if Web services are not well-behaved. There are many reasons that Web services cannot conform to the definitions of well-behaved-interface – the Internet is not always reliable and there are situations that Web-service designers have not anticipated or they are beyond the control of the services such as delay of back order items and natural disasters. We propose a simple and effective method to estimate affects on the users during their interactions with Web services. The method relies on prospective events which are main derivers of emotional states of human users according to cognitive appraisal theories of emotion (e.g., Ortony, Collins and Clore (OCC) [23]).

4.1 Prospective Events of Web Services

We consider two classes of goals: *directive goals* created by directive one-way operations and *promised goals* created by commissive notification operations. Given the two classes of goals, we can now enumerate the events that are relevant to these goals. But first, we make two important assumptions. When a user interacts with the service, it is reasonable to assume the followings:

1. Users know the meaning of each interface;
2. Users are aware of (or accustomed to) all prospective events related with the goals.

Based on these assumptions, we obtain the *user side time events* listed in Table 2. The service responsible for the goals must struggles to prevent these events occurring by producing *informing events* listed in the same table. Table 1 shows the descriptions of

Table 1. Prospective events for a goal g and OCC classification of the events.

Event Names	**Symbols**
goal failure time event	gfe$_g$
response failure time event	rfe$_g$
confirming event	ce$_g$
disconfirming event	dce$_g$
delay event	dlye$_g$
acknowledgement event	acke$_g$

OCC Types	**Symbols**
Prospective	all
Unexpected	-
Desirable	ce$_g$
Undesirable	gfe$_g$, rfe$_g$, dce$_g$
Unconfirmed	dlye$_g$, acke$_g$
Confirming	ce$_g$
Disconfirming	dce$_g$

Table 2. Interface based classification of events.

Event Types	**Symbols**
User side time events: TE_g	gfe$_g$, rfe$_g$
Informing events: IE_g	ce$_g$, dce$_g$, dlye$_g$, acke$_g$

the event symbols. Table 1 also shows Ortony, Collins and Clore (OCC) [23] classification of these events. OCC have proposed a classification of events and their affects on (causes emotions) on communication participants.

As shown in Figure 1, for a directive goal g, there will be the time rt_g to inform of the acknowledgement of the acceptance of the goal to the user and the time gt_g to fulfill the goal before the user aware of undesirable events. However, when the user is not informed of the achievement of the goal within the goal achievement time gt_g, a goal failure time event gfe_g fires. When neither the achievement nor an acknowledgement is informed to the user within the response time rt_g, a response failure event rfe_g fires. As shown in Figure 1, for promised goals, rt_g is not necessary and only goal failure event gfe_g will occur.

Any user side time events $\{gfe_g, rfe_g\}$ can cause negative emotions on the user. Thus, the process must create appropriate informing messages to prevent the user side time events occurring. According to Figure 1, there are four possible types of informing events: confirming events ce_g, disconfirming events dce_g, delay events $dlye_g$, and acknowledgement events $acke_g$. Each of these events occurs when the service sends the corresponding notification messages. The following production rules (whose conclusions are evaluated when their conditions are satisfied) summaries the event firing policies for the user side time events:

$$r_1 : directive_g \wedge (rt_g \leq t) \wedge \neg(acke_g \wedge dlye_g \wedge ce_g \wedge dce_g) \rightarrow rfe_g$$
$$r_2 : (gt_g \leq t) \wedge \neg(ce_g \wedge dce_g) \rightarrow gfe_g$$
$$r_3 : delay_g(d) \rightarrow (gt_g = gt_g + d)$$

Rule r_1 says that if the response time has passed and there have been no responses at all for a directive goal g, a response failure time event rfe_g occurs. r_2 says that if the goal achievement time has passed and there have been neither a confirming message nor a disconfirming message, then a goal failure time event gfe_g occurs. r_3 says that a delay message resets the goal achievement time.

If a failure event occurs, a new promising goal g' should be created in order to compensate the failure. The promising goal can only be formulated if we know the affect of the failure on the user.

4.2 Estimating Emotional States

This section describes how emotional states can be deduced from the prospective events of Web services based on the work of the Ortony, Collins and Clore (OCC) [23] cognitive appraisal theory of emotion which is one of the most widely accepted emotion models. The OCC model defines twenty-two emotion types, but we only describe six of them in this paper: hope, satisfied, fear, fears-confirmed, disappointment, and reproach. These emotions are prospective-based emotions that are emotions in response to expected and suspected states and in response to the confirmation or disconfirmation of such states [23].

Although the events we have described provide significant information to estimate users' emotional states, there can be always many other sources that can affect the users. Thus, we cannot use strict rules to capture relations between the events and emotional states. We use a fragment of Defeasible Logic (DL) [22] which is a popular nonmonotonic logic that is simple and computationally efficient. In DL, A defeasible rule $L \Rightarrow c$ consists of its antecedent L which is a finite set of literals, an arrow, and its consequent c which is a literal. A literal is an atom p or its negation $\neg p$. A defeasible rule $a_1, ..., a_n \Rightarrow c$ can be expressed in the following logic program (without the monotonic kernel and the superiority relation of DL):

$$supported(c) \text{ :- } conclusion(a_1), ..., conclusion(a_n).$$
$$conclusion(c) \text{ :- } supported(c), \textit{not } supported(\sim c), \textit{not } strictrly(\sim c).$$

where $\sim c$ is the complement of literal c; $conclusion(c)$ denotes that c is defeasibly provable; $strictrly(c)$ denotes that c is strictly provable. Then, the following defeasible rules roughly capture the relations between the events and emotional states:

R0. $\Rightarrow \neg rfe_g, \quad \Rightarrow \neg gfe_g$
R1. $ce_g \Rightarrow \neg hope_g, \quad dce_g \Rightarrow \neg hope_g, \quad gfe_g \Rightarrow \neg hope_g$
R2. $ce_g \Rightarrow \neg fear_g, \quad dce_g \Rightarrow \neg fear_g, \quad gfe_g \Rightarrow \neg fear_g$
R3. $directive_g, \neg rfe_g \Rightarrow hope_g$
R4. $directive_g, rfe_g \Rightarrow fear_g$
R5. $directive_g, \neg rfe_g, \neg gfe_g, ce_g \Rightarrow satisfied_g$
R6. $commissive_g, \neg gfe_g, ce_g \Rightarrow satisfied_g$
R7. $directive_g, rfe_g, gfe_g \Rightarrow fearconfirm_g$
$directive_g, rfe_g, dce_g \Rightarrow fearconfirm_g$
R8. $directive_g, \neg rfe_g, gfe_g \Rightarrow disappoint_g$
$directive_g, \neg rfe_g, dce_g \Rightarrow disappoint_g$
R9. $commissive_g, gfe_g \Rightarrow disappoint_g$
$commissive_g, dce_g \Rightarrow disappoint_g$
R10. $directive_g, rfe_g, ce_g \Rightarrow relieved_g$

Rules in R0 are assumptions that response failure events and goal failure events are not occurred. Rules in R1 and R2 say that if a communication is ended, a user usually feels neither hope nor fear. R3 says that a user usually feels hope over a desired goal g if no fear prospect (rfe_g) is triggered. R4 says that a user usually feels fear if a desirable goal seem to be failing ($rfe(g)$). Rules in R5 and R6 say that a user usually feels satisfied if a desirable goal (*directive* or *commissive*) is fulfilled without a trouble. Rules in R7 say that a user usually feels fear-confirmed if a desirable goal ($directive(g)$) that seems to be failing is actually failed. Rules in R8 and R9 say that a user is usually disappointed if a desirable goal (*directive* or *commissive*) is failed. R10 says that a user is usually relieved if the user has had fear over a desirable goal (*directive*), but it is actually achieved.

The rules can be used to predict and estimate various affects that a Web service can cause on it users. With this information, a Web-service management system can prevent potential negative effects, compensate negative effects (e.g., sending an apology gift when a delivery is delayed), and even take advantage of positive affects (e.g., advertising when goods are successfully delivered without any troubles). This is a perfect technology that can tell when it is acceptable to send spam messages or to show pop-up advertisements.

5 Conclusion

Intelligent agent research communities have been working on various agent communication languages (e.g., KQML[13], FIPA-ACL[14]) [21,28] based on speech act theory. Speech act theory [27,8,9,10] has helped defining the types of messages based on the concept of illocutionary point, which constraints the semantics of the communication act itself [18, p.87]. It is also used as a basic ontology in organizational management systems [15] and in a conversation model for Web services [2].

However, Web service development communities have largely ignored the semantic issues of interactions. Currently, most works on Web service focus on design tools, infrastructure, and Web service composition. Thus, the standards developed for Web services (e.g., WDSL) mainly focus on the syntaxes of the description languages (e.g., WSDL), structural issues, or operational semantics (e.g., BPEL[1]) largely ignoring various service quality issues. Thus, there are currently no standard ways to represent necessary data for quality management; it is difficult to compose Web services that meets the minimum requirement for *well-behaved Web services* for human users.

In contrast to usability evaluation methods (e.g., [19]), we only focus on the affects on the Web services users rather than efficiency oriented issues such as cognitive workload, performance, easy of use, and easy to learn. We should also note that our work does not require any of direct measurements of users such as brain activity, facial expression unlike existing approaches of affective computing. All information required is provided by the illocutionary points of operations and the emotion generation rules because our work only account for affects related to the goals that the services promise to deliver. However, actually these affects are the main issues that must be addressed.

This paper proposed that we need to incorporate communication modes in the current standards of Web-service specifications in order to improve customer satisfaction. We showed that with the communication modes, we can define well-behavedness of Web-service interfaces and estimate various affects on customers during their interac-

[1] http://www.ibm.com/developerworks/webservices/library/ws-bpel/

tions with Web services. The result is important, since more businesses are relying on Web services as their primary contacts of customers.

References

1. ACSI. American customer satisfaction index. *http://www.theacsi.org*, 2004.
2. L. Ardissono, A. Goy, and G. Petrone. Enabling conversations with Web services. In *Proc. AAMAS 2003*, pages 819–826, 2003.
3. Lesley Axelrod and Kate Hone. E-motional advantage: performance and satisfaction gains with affective computing. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1192–1195, New York, NY, USA, 2005. ACM Press.
4. Alberto Battocchi, Fabio Pianesi, and Dina Goren-Bar. A first evaluation study of a database of kinetic facial expressions (dafex). In *Proc. ICMI '05: the 7th international conference on Multimodal interfaces*, pages 214–221, New York, NY, USA, 2005. ACM Press.
5. Timothy W. Bickmore and Rosalind W. Picard. Establishing and maintaining long-term human-computer relationships. *ACM Trans. Comput.-Hum. Interact.*, 12(2):293–327, 2005.
6. Carlos Busso, Zhigang Deng, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann, and Shrikanth Narayanan. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proc. ICMI '04: the 6th international conference on Multimodal interfaces*, pages 205–211, New York, NY, USA, 2004. ACM Press.
7. Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web services description language (WSDL). *W3C: http://www.w3.org/TR/wsdl*, 2001.
8. Philip R. Cohen and Hector J. Levesque. Performatives in a rationally based speech act theory. In *Meeting of the Association for Computational Linguistics*, pages 79–88, 1990.
9. Philip R. Cohen and Hector J. Levesque. Rational interaction as the basis for communication. In *Intentions in Communication*, pages 221–255. MIT Press, Cambridge, Massachusetts, 1990.
10. Philip R. Cohen and Hector J. Levesque. Communicative actions for artificial agents. In *Proc. ICMAS '95*, pages 65–72, San Francisco, CA, USA, 1995. The MIT Press: Cambridge, MA, USA.
11. Erica Costantini, Fabio Pianesi, and Michela Prete. Recognising emotions in human and synthetic faces: the role of the upper and lower parts of the face. In *Proc. IUI '05*, pages 20–27, New York, NY, USA, 2005. ACM Press.
12. F.N. Egger. Affective design of e-commerce user interfaces: How to maximise perceived trustworthiness. In *Proc. CAHD2001: Conference on Affective Human Factors Design*, pages 317–324, 2001.
13. Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. Kqml as an agent communication language. In *Proc. CIKM '94: the third international conference on Information and knowledge management*, pages 456–463, New York, NY, USA, 1994. ACM Press.
14. FIPA [Fondation for Intelligent Physical Agents]. FIPA ACL message structure specification. *http://www.FIPA.org*, 2002.
15. Fernando Flores, Michael Graves, Brad Hartfield, and Terry Winograd. Computer systems and the design of organizational interaction. *ACM Trans. Inf. Syst.*, 6(2):153–172, 1988.
16. Kiel M Gilleade and Alan Dix. Using frustration in the design of adaptive videogames. In *Proc. ACE '04: the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 228–232, New York, NY, USA, 2004. ACM Press.
17. Lisa Gralewski, Neill Campbell, Barry Thomas, Colin Dalton, David Gibson, and University of Bristol. Statistical synthesis of facial expressions for the portrayal of emotion. In *Proc.*

GRAPHITE '04: the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, pages 190–198, New York, NY, USA, 2004. ACM Press.
18. Michael N. Huhns and Larry M. Stephens. Multiagent systems and societies of agents. In *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 2, pages 79–120. The MIT Press, Cambridge, MA, USA, 1999.
19. Melody Y. Ivory and Marti A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33(4):470–516, December 2001.
20. Jonathan Klein, Youngme Moon, and Rosalind W. Picard. This computer responds to user frustration. In *Proceedings of ACM CHI 99*, volume 2, pages 242–243, 1999.
21. Yannis Labrou, Tim Finin, and Yun Peng. Agent communication languages: The current landscape. *IEEE Intelligent Systems*, 14(2):45–52, March/April 1999.
22. M. J. Maher and G. Governatori. A semantic decomposition of defeasible logics. In *AAAI '99*, pages 299–305, 1999.
23. A. Ortony, G. L. Clore, and A. Collins. *The Congnitive Structure of Emotions*. Cambridge University, Cambridge, 1988.
24. Rosalind W. Picard. *Affective Computing*. MIT Press, 1997.
25. Rosalind W. Picard. Affective computing: challenges. *International Journal of Human-Computer Studies*, 59(1/2):55–64, 2003.
26. John Searle. *Expression and Meaning*. Cambridge Univesity Press, Cambridge, England, 1979.
27. Munindar P. Singh. A semantics for speech acts. In *Readings in Agents*, pages 458–470. Morgan Kaufmann, San Francisco, CA, USA, 1997. (Reprinted from *Annals of Mathematics and Artificial Intelligence, 1993*).
28. Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley Sons, 2001.