Instant Messaging Bots: Accountability and Peripheral Participation for Textual User Interfaces

Stephen Chan
SIMS
UC Berkeley
102 South Hall
Berkeley, CA 94720-4600
sychan@sims.berkeley.edu

Benjamin Hill
SIMS
UC Berkeley
102 South Hall
Berkeley, CA 94720-4600
bhill@sims.berkeley.edu

Sarita Yardi
SIMS
UC Berkeley
102 South Hall
Berkeley, CA 94720-4600
yardi@sims.berkeley.edu

ABSTRACT

Over the last several years, studies of instant messaging have observed its increasing role in the workplace[1] and in social situations[2]. We propose that modifying applications to interact with users over Instant Messaging (as IM bots) extends the collaborative benefits of IM into new areas. As IM Bots participating in group chatrooms, applications that had previously been restricted to a single user command line are able to engage in many to many interactions between users and applications. Current command line oriented user interfaces can be made into collaborative interfaces that exhibit (at a basic level) the ethnomethodological property of accountability as well as supporting legitimate peripheral participation.

Categories and Subject Descriptors

H.5.3 [**Group and Organization Interfaces**]: Collaborative Computing – *computer-supported cooperative work, synchronous interactions.*

General Terms

Design, Human Factors

Keywords

Instant Messaging, Ethnomethodology, Technomethodology, Legitimate Peripheral Participation, Software Agents, Learning, Collaboration

1. INTRODUCTION

Accountability is a principle from ethnomethodology[3] where the performance of an action simultaneously demonstrates the rationality of the person performing the action. The way in which an action is performed provides a commentary to observers, providing a form of transparency that assures people of the correctness of the actions as well as the skill of the actor.

Legitimate Peripheral Participation is a principle from theories of situated learning[2] that describes how people at the periphery of a community can observe and participate in practices as part of socialization and learning processes. For peripheral participation to occur, it is necessary for members at the periphery to observe the skilled performance of experienced practitioners, as well as observe the interactions between experienced members of the community. Legitimate peripheral participation is part of the process whereby communities create and recreate themselves.

These two principles are complementary: accountability in a work practice not only allows peers to verify rationality, it also allows novice observers to peripherally participate, which is essentially a learning process.

These two principles motivate our suggestion that instant messaging be investigated as a supplementary user interface for software. Instant messaging allows textual status information to be viewed collaboratively. In response to the status information, commands can be issued, which are also visible to all parties in the conversation. Logs of IM conversations can be saved and replayed for explanation, analysis and discussion – further promoting peripheral participation. Novice observers of the IM conversation can be thought of as participating in a form of apprenticeship – observing the activities of the experts. Expert participants in the IM conversation serve as peers, validating the expertise of the performer, as well as providing a form of distributed error correction. The use of IM also allows back channel commentary to occur, facilitating the learning and socialization process: commentary can come from fellow apprentices, as well as more experienced practitioners, promoting vertical as well as horizontal learning.

2. Instant Messaging Between Human Agents

Instant messaging has been studied extensively as a form of collaboration. This has generally been in the context of human agents participating in conversations with other people, in social and workplace settings. Nardi, Whittaker and Bradner[1] provide an ethnographic study of IM in the workplace as a distinct form of interaction. Hansen and Damm[4] study the integration of IM within a broader set of collaboration tools, moving along the path of IM becoming an integrated part of broader tools.

These studies provide valuable insight into the role of IM in the workplace, as a tool for collaboration. It is also possible for software to engage in IM conversations, as what are colloquially known as IM bots. Our research has not turned up much work on the potential of IM enabled software applications to support

GROUP'05, November 6–9, 2005, Sanibel Island, Florida, USA.
Copyright 2005 ACM 1-59593-223-2/05/0011...$5.00.

CSCW. This paper is an attempt to draw more attention to this particular application of IM.

3. Instant Messaging Bots

At the DIAC 2005 conference, Matthew Easterday demonstrated an IM bot interfaced to argument mapping software[5]. This software package could be controlled via commands from an IM conversation. This is an example of providing an IM interface into a non-IM related application, providing a collaborative interface that supports accountability and legitimate peripheral participation.

Instant messaging bots are not a new invention: simple bots controlled via IRC and similar chat systems have existed for many years, but we feel that if this form of user interface is understood in the proper theoretical context, it will motivate further development and adoption.

An application can have an instant messenger bot embedded within it that parses IM conversations for specially formatted commands. These commands would then be executed and the results sent back to participants in the conversation. Multiple users and applications can all be part of the conversation, providing a many to many interaction between bots and people that cannot be achieved with tools that merely share a command line (such as the Unix utility *screen*). Users can also engage in commentary and discussion, interspersed as necessary with commands to the bots. Bots could easily be designed to send information to each other, producing more complex workflows that can be observed, controlled and modified by human observers.

As examples of applying these principles to an IM based interface, we offer the following:

Collaborative Debugging: an instant messaging bot is tied to a debugger. This debugger allows a group of developers (and any observers) to execute programs, examine program state, set breakpoints, etc… This would allow a new and novel form of collaborative programming practice – supporting forms of apprenticeship, as well as expanding on ideas of paired debugging.

Software that provides an account of itself: This is comparable to the "debug mode" that some software provides: a software package describes its actions and state changes over an IM channel. In addition, this can be expanded to support non-programmer's asking for an explanation from a software package. Software could provide context sensitive help over an IM channel, or it could explain the internal logic that led to a recent action. Software that is engaged in long running, complex plan oriented actions could provide an account of its plan and its progress in the plan (for example: a database engine could describe the query execution plan being executed and its current location in the plan).

Distributed Computing becomes Collaborative Computing: Grid computing is an example of widely distributing computing systems. Although a Grid application is distributed, this does not ensure that its interface is similarly collaborative. An IM interface would allow the resources involved in a distributed computation to present their interfaces in a collaborative mode.

Distributed Command Lines: Command lines are often the interface of choice for expert users. However, command lines don't typically lend themselves to collaboration. Instant messaging could be used to create a form of distributed command line, allowing multiple people to both observe and enter commands. Collaborative forms of work such as Systems and Network Administration[8] that are heavily dependent on command lines, can benefit. An example is extending Intrusion Detection Systems to support collaboration via IM bots[9].

Many of the examples described already exist in some form, however they are typically accessed via relatively arcane and non-interactive methods, that are typically not collaborative and not distributed. IM would address these issues in a standard way, using tools that are popular, open and user friendly.

4. IM and Collaboration

As mentioned earlier, the key principles we want to emphasize are accountability and peripheral participation. By making the user interface interaction public over an IM conversation that offers a simultaneous channel for discussion and commentary, we fulfill the principle of accountability in work practice: observers are able to see what a practitioner is doing and how they do it. This is not a new form of interaction: people have been looking over the shoulders of their contemporaries from the earliest times up till the present. The role of Instant Messaging is to dramatically scale up the number and range of people that can simultaneously observe and/or participate, as well as make it possible for people in the future to observe via logged conversations. In activity theoretic terms[6], the use of IM as an interface allows both temporal and socio-spatial expansion of the activity.

Peripheral participation allows observing parties to learn, both at the social level and at the technical level. By making the user interface collaborative, we have expanded the range of potential peripheral participation, both in terms of the number of people that can observe, and also the types of activities that can be observed.

The accountability aspect of allowing peers to review actions not only supports social functions, it also has the potential to improve the usability of software. Software that provides an account of its behavior has important usability properties[3]. Accountability is not only between a practitioner and their peers, but also between the software and its users – IM has the potential to support both.

5. Issues for Further Exploration

There are issues that need further exploration in dealing with IM as a user interface. These topics are beyond the scope of a short, exploratory paper, but they merit mention:

Access Control and Security

These are general problems with computer use. However, the unique interactions possible in a "distributed command line" may present new and novel issues that require research.

Privacy

There seems to be a tension between accountability and privacy. Privacy is a very broad and thorny issue, beyond the scope of such a short paper. But it is clearly a topic that may need to be addressed at some point.

Etiquette

Every form of human interaction involves some form of social protocol. In a conversation that involves multiple parties, new forms of etiquette will need to be created to handle how people interact with each other, as well as the software agents on call.

There are doubtless many others, which will become apparent as IM based user interfaces are more fully explored.

6. Summary

Instant Messaging is a tool that has become pervasive, and is heavily studied. We feel that there are important applications of IM as a *user interface* that can promote collaboration. By applying the theoretical notions of accountability and legitimate peripheral participation, we can see how these new forms of interaction can be structured to promote learning and usability. We also stress that accountability is in terms of a practitioner being accountable to peers, as well as a software system being accountable to its users.

The side effect of promoting accountability is that peripheral participation also becomes easier to achieve – even for modes of interaction as obscure as a command line. We have offered some suggestions for possible uses and future issues for exploration. It is our hope that this will spur investigation and further research.

7. REFERENCES

[1] Nardi, B., Whittaker, S., Bradner E., Interaction and Outaction: Instant Messaging in Action, *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '00)* (Philadelphia, PA). ACM Press, New York, NY, 2000, 79-88

[2] Grinter, R.E., Palen, L., Instant Messaging in Teen Life, *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '02)* (New Orleans, LA). ACM Press, New York, NY 2002, 21-30

[3] Dourish, P., Button, G., Technomethodology: Paradoxes and Possibilities, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems:Common Ground* (Vancouver, British Columbia), ACM Press, New York, NY, 1996, 19-26

[4] Hanks, W. Foreward *Situated Learning: Legitimate Peripheral Participation*, J. Lane and E. Wenger, Cambridge University Press, Cambridge, England

[5] Hansen, K, Damm C,. Instant collaboration: Using Context Aware Instant Messaging for Session Management in Distributed Collaboration Tool*s, Proceedings of the second Nordic conference on Human Computer Interaction(NordiCHI 10/02)* (Arhus, Denmark) ACM Press, New York, NY 2002, 279-282

[6] EasterDay, M. Kanarek, J Harrell, M. Design Requirements of Argument Mapping Software for Teaching Deliberation, Presented at *Online Deliberation Conference 2005 (Stanford University, Palo Alto, CA)*, Online Deliberation 2005/DIAC-2005, 2005

[7] Engestrom, Y. Activity Theory as a framewortk for analyzing and redesigning work, *Ergonomics 2000* 43,7, 2000

[8] Barret, R., Kandogan, E., Maglio, P, Haber, E., Takayama, L., Prabaker, M., Field Studies of Computer Systems Administrators: Analysis of System Management Tools and Practices, *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '04)* (Chicago, IL), ACM Press, New York, NY, 2004, 388-395

[9] Campbell, S., Chan, S., Making Intrusion Detection Systems Interactive and Collaborative, Presented at the *14^{th} Usenix Security Symposium: Works in Progress* (Baltimore, MD) , 2005, http://www.usenix.org/events/sec05/wips.html