Ethics for Bots[1]

Miranda Mowbray
Applications Systems Department
HP Laboratories Bristol
HPL-2002-48(R.1)
February 28, 2005*

Bots, online communities, MOO

The rise of online communities has led to a phenomenon of real-time, multiperson interaction via online personas. Some online community technologies allow the creation of bots (personas that act according to a software programme rather than being directly controlled by a human user) in such a way that it is not always easy to tell a bot from a human within an online social space. In this paper I illustrate ethical issues in bot design by discussing some dilemmas and problems of policing bots, with real examples from Little Italy MOO, an online community, and elsewhere.

* Internal Accession Date Only

[1] 14th International Conference on System Research, Informatics and Cybernetics, Baden-Baden, July 29-Aug 3, 2002

Approved for External Publication

Ethics for Bots

Miranda Mowbray, Hewlett Packard Laboratories Bristol

In Proc. Inter-Symp ’02, 14th International Conference on System Research, Informatics and Cybernetics, Baden-Baden, July 29-Aug 3, 2002

Abstract

The rise of online communities has led to a phenomenon of real-time, multiperson interaction via online personas. Some online community technologies allow the creation of bots (personas that act according to a software programme rather than being directly controlled by a human user) in such a way that it is not always easy to tell a bot from a human within an online social space.

In this paper I illustrate ethical issues in bot design by discussing some dilemmas and problems of policing bots, with real examples from Little Italy MOO, an online community, and elsewhere.

Keywords

Bots, online communities, MOO

Introduction

The rise of online communities has led to a phenomenon of real-time, multiperson interaction via online personas. Some online community technologies allow the creation of bots (personas that act according to a software programme rather than being directly controlled by a human user) in such a way that it is not always easy to tell a bot from a human within an online social space. It is also possible for a persona to be partly controlled by a software programme and partly directly by a human. In some social spaces a single user can have multiple personas, perhaps with different characteristics and characters, some or all of which can be bots. On the other hand, a single bot can be programmed or influenced by multiple people.

This leads to theoretical and practical problems for ethical arguments (not to mention policing) in these spaces, since the usual one-to-one correspondence between actors and moral agents can be lost. In this paper I will ask what the ethical design of bots might mean.

I will illustrate ethical issues in bot design by discussing some dilemmas and problems of policing bots, with real examples from Little Italy MOO, an online community, and elsewhere. (To access Little Italy MOO, telnet to kame.usr.dsi.unimi.it 4444, and type "connect guest".)

Issue 1: Bot mistaken for human

A feature of bots that can raise particular problems is that it can sometimes be difficult to distinguish a bot from a human. This does not generally happen for other software or robots. It is not principally a feature of the sophistication of bot design, but of the low bandwidth communication of the online social spaces within which the bots act. It is much easier to convincingly simulate a human agent within an online community in which all communication is in text, for example, than in a system generating 3D video, although bots with visuals that are designed to mimic humans do exist – the newsreader Ananova ™ (Ananova Ltd, 2001) is an example.

A user who mistakes a bot for a human can suffer hurt feelings and wasted time. See (Foner, 1993) for a description of Julia, a bot easily mistaken for a human. Section 3_3 of Foner's paper includes a partial transcript of a user's unsuccessful attempts, over 13 days, to chat up Julia. A user in Little Italy MOO was told by his friends that a bot called Gnagna with limited conversational abilities was a girl who fancied him; he thought she had encrypted her conversation in some way and asked me whether I knew how to decode her utterances.

If a bot behaves badly and is thought to be human then this can lead to a waste of time, energy and eloquence by users who try to reason with it. Gnagna often steals users' possessions within the MOO, and has been sent pleading messages from users trying to regain them.

It's probably not possible to defend the sensibilities of the least astute users. However, perhaps designers should give indications that the bot is not human. For example, Gnagna replies to any remark far faster than a human typist could. Some bots bear a sign saying that they are not human. Others, like Julia, evade the question even if asked directly whether they're human. This may not be not a good idea.

Issue 2: Loss of Privacy

Cobot, a bot in LambdaMOO, collects statistical data about the behaviour of users. To access LambdaMOO, telnet to lambda.moo.mud.org 8888 and type "connect guest". See (Isbell Jr. et al, 2000) for information about Cobot. Some LambdaMOO users raised privacy issues about this data collection, and the Cobot programmers incorporated several features for protecting users' privacy. Questioners can generally only ask Cobot about themselves, not about others. Cobot does not share events verbatim. Users can also opt-out from their data being collected by Cobot. Some users still think this privacy protection does not go far enough. In Little Italy MOO, one user programmed a bot in order to spy on other users. This user was banned from the MOO after making antisocial use of private conversations that he overheard using his bot.

In general, programmers who use bots to collect data should do so with care. The collection of anonymized data, or statistical data about groups of users, is less problematic. However, if a bot collects information on individual users identifiable by their names within the online community, in my opinion it should aim to follow the principle of informed consent.

Issue 3: Spamming

Bots can send messages faster than humans can type. This means that they have the capability to fill a user's screen with a large number of unwanted messages. This behaviour is known as "spamming", in analogy with unwanted email. Some bots are programmed to carry out this antisocial behaviour – for example, some ICQ channel bots are deliberately designed to do this to encourage particular users to leave a channel. Others can be coerced into this behaviour by other users. Some LambdaMOO users made Cobot spam by repeatedly querying him aloud. To combat this, Cobot's programmers made it possible for a designated list of regular users to make Cobot ignore a particular individual. Cobot also has a "silence" verb that stops him speaking aloud for a random length of time.

In the Brazilian online community, Cpdee MOO, I observed a user construct a new bot that was designed to coerce an existing bot into spamming. (To access Cpdee MOO, telnet to moo.cpdee.ufmg.br 7777 and type "connect guest".) The coerced bot, when asked a question containing a particular word A, would always answer aloud with a phrase containing word B. The newly constructed bot, when it heard word B, would ask the first bot a question containing word A twice. These two bots could stay within earshot of each other without any problems until some user asked a question containing A or B. The number of messages from the bots then escalated each round until the environment was unusable by humans. Although this was deliberate, it is clear that a similar effect could occur through the interaction of two bots whose designers had no such intention. A similar, though less drastic, loop sometimes appears involuntarily in email mailing lists if two list members have auto-reply bots wrongly configured in such a way that they each answer the other.

In addition to the safeguards that Cobot uses, two other strategies are for a bot not to speak if the noise level of the environment is already above a certain level, and to give the bot a “gag” verb allowing a user to specify that she will not see any more of the bot’s messages until further notice.

Issue 4: Socially destructive unfair advantage

Bots are often built to give their programmer or owner some advantage over other users. This is fair enough, but in some cases these bots have caused problems in the smooth running of the online environment. As mentioned above, some ICQ channel bots are designed to be obnoxious to particular users, and Gnagna has strong kleptomaniac tenencies, but a socially destructive unfair advantage can occur even when a bot is not socially deviant. Little Italy introduced an internal monetary system through which virtual objects were bought and sold; virtual money could be earned by programming socially useful objects, or by working in a MOO factory. Some users discovered how to make bots that could work 24 hours a day in the factory on their behalf. The earnings of these bots vastly increased the money supply, leading to hyperinflation, collapse of the monetary system, and a return to barter, before the MOO economists could work out what was going wrong.

Issue 5: Lack of transparency

Gnagna has been designed over several years, and does some automated learning from its experiences in Little Italy MOO. It reacts to its environment, sometimes surprisingly. Gnagna’s programmer has said that he does not always know why Gnagna acts a particular way. This is a general problem of complex self-learning programmes that interact with a complex environment. Moreover, in some online communities bots continue acting when their programmers are not present and difficult to contact. As a result, when a bot exhibits socially harmful behaviour it may not necessarily be easy to reprogramme it.

The examples in this paper show that a bot may cause harm to other users or to the community as a whole by the will of its programmers or other users, but that it also may cause harm through nobody’s fault because of a combination of circumstances involving some combination of its programming, the actions and mental/emotional states of human users who interact with it, behaviour of other bots and of the environment, and the social economy of the community. Although an ethically designed bot would not be designed to behave antisocially, and might have safeguards built in to make it difficult for users to coerce it into doing so, it is difficult if not impossible to design an interesting bot that will behave well in all possible circumstances in a complex environment. An ethically designed bot, therefore, should aim to be easy to reprogramme in case of emergencies.

As a first defence, the programmer (or some other responsible human) should be contactable through the bot by users who have complaints about its behaviour. Although it would be nice for a bot to be able to detect when it had made a faux pas and alter its behaviour accordingly, this is probably only possible for offences - such as spamming - whose detection does not require the bot to be able to tell that it has hurt a user’s feelings. Moreover, there is always the possibility of misprogramming and unexpected interactions that would incapacitate such automatic adjustment of behaviour.

The bot should be designed to make it reasonably clear to users that interact with it what the consequences of their interaction are likely to be. For instance, a bot should not offer users a verb whose name suggests that when it is invoked a pleasant message will be sent to another user, whereas in fact a mildly unpleasant one is sent. There is a programme in Little Italy that does this.

For easy reprogramming it is a good idea to make the bot design modular, and well documented, with the ability to disable specific actions. For the case where the programmer is unavailable, the bot should be reprogrammable by the staff of the online community, or other regularly present and responsible users. It may be helpful for the online community to have public rules of conduct for bots that make clear the circumstances under which a bot may be forcibly reprogrammed.

Issue 6: Subversion of the bot by online community staff

The consideration of the examples in this paper has led me to suggest that an ethically designed bot would be designed to avoid causing social harm, and would easily reprogrammed by responsible users and staff. As a reviewer of this paper has pointed out, the second aim could be in conflict with the first if the staff themselves wish to carry out harmful activities. For example, staff of an online community run by either a government body or a commercial company interested in obtaining marketing information might well be tempted to use bots to collect sensitive information about users without the users' consent.

A technical safeguard would be for responsible users and staff not to be given full reprogramming powers in an emergency, but only the power to disable groups of actions (or, more crudely, just the power to disable the bot entirely.) The modular design should be done carefully, with the aim of making it difficult to make the bot carry out harmful behaviour just by disabling some or all of its groups of actions.

It is sensible to have social and political safeguards as well as technical ones, especially if there is a possibility of a harmful bot being deliberately designed by a member of staff. One political safeguard would be to make the staff democratically accountable to the users for their actions and policies.

Acknowledgement

Grazie ai cittadini e wiz di Little Italy per loro aiuto e amichevolezza.

References

Ananova Ltd (2001); http://www.ananova.com/

Foner, L., (1993); What's an Agent, Anyway?: A Sociological Case Study; http://foner.www.media.mit.edu/people/foner/Julia/Julia.html

Isbell Jr., C.L., Kearns, M., Kormann, D., Singh, S. and P. Stone (2000); Cobot in LambdaMOO: A Social Statistics Agent; Proc. AAAI 2000, AAAI Press/The MIT Press (pp. 36-41); http://cobot.research.att.com/papers/cobot.pdf