Available online at www.sciencedirect.com
SCIENCE DIRECT®

The Journal of Systems and Software 76 (2005) 65–76

The Journal of Systems and Software

www.elsevier.com/locate/jss

Effects of introducing survival behaviours into automated negotiators specified in an environmental and behavioural framework

Peter Henderson, Stephen Crouch *, Robert John Walters, Qinglai Ni [1]

Declarative Systems and Software Engineering, School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

Received 15 December 2003; received in revised form 15 April 2004; accepted 15 June 2004
Available online 26 August 2004

Abstract

With the rise of distributed e-commerce in recent years, demand for automated negotiation has increased. In turn, this has engendered a demand for ever more complex algorithms to conduct these negotiations. As the complexity of these algorithms increases, our ability to reason about and predict their behaviour in an ever larger and more diverse negotiation environment decreases. In addition, with the proliferation of internet-based negotiation, any algorithm also has to contend with potential reliability issues in the underlying message-passing infrastructure. These factors can create problems for building these algorithms, which need to incorporate methods for survival as well as negotiation.

This paper proposes a simple yet effective framework for integrating survivability into negotiators, so they are better able to withstand imperfections in their environment. An overview of this framework is given, with two examples of how negotiation behaviour can be specified within this framework. Results of an experiment which is based on these negotiation algorithms are provided. These results show how the stability of a negotiation community is affected by incorporating an example survival behaviour into negotiators operating in an environment developed to support this framework.

Keywords: E-commerce; Automated negotiation; Negotiation framework; Pseudocode

1. Introduction

1.1. Background

The choice of algorithm used to carry out automated negotiation on behalf of a client is a significant problem in distributed e-commerce (Bichler et al., 1998; Burg, 2002; Cranor and Resnick, 1997; Farhoodi and Fingar, 1997; Fingar et al., 2000; Henderson, 2002). Furthermore, predicting how well a given algorithm will perform in a given environment is difficult.

The ability of an algorithm to succeed in an automated negotiation environment is dependent on its ability to survive in that environment. Automated negotiators, on their own terms, must be able to make sense of and conduct negotiation on the web in which there are no guarantees of the reliability of the underlying message-passing infrastructure. As the web increases in size and interconnectivity, this will become an even greater problem. In some cases, offers sent may not be received at all, but equally problematic is that offers received are out of date. Suppose a negotiator receives an offer that has spent an inordinate amount of time in transit. Despite replying promptly in sending an accept to this offer, the negotiator finds their acceptance rejected because the offer's sender has already sold their

* Corresponding author. Tel.: +44 23 8059 7684; fax: +44 23 8059 3045.

E-mail addresses: p.henderson@ecs.soton.ac.uk (P. Henderson), s.crouch@ecs.soton.ac.uk (S. Crouch), r.j.walters@ecs.soton.ac.uk (R.J. Walters), q.ni@reading.ac.uk (Q. Ni).

[1] Present address: School of Plant Sciences, University of Reading, Reading, Berkshire RG6 6AS, UK.

0164-1212/$ - see front matter
doi:10.1016/j.jss.2004.06.024

last stock to someone else. Agreement is not reached because of inconsistent views of negotiation, caused by inconsistent information. The ability to tolerate this information inconsistency, and being able to minimise its negative effects by taking corrective or compensating action, may reward the negotiator with greater success.

To compound these issues, a negotiator cannot be certain whether their experience of such problems with another negotiator is because of natural occurrence, or faulty or even malicious behaviour. It is also possible that the two negotiators are simply unable to reach agreement because they exhibit mutually incompatible negotiation strategies. To succeed, automated negotiators must be able to survive and progress despite such eventualities, without knowing the intent of other negotiators.

It is not uncommon for communities of automated negotiators to establish stable norms of behaviour. Over time, despite negotiators' different behavioural characteristics, initially erratic patterns of negotiation can eventually settle into predictable patterns of apparent co-operation (Fogel, 2000; Young, 1998). However, making successful predictions about how and in what form such stability will emerge can prove difficult (Axelrod, 1984, 1997). Even more difficult are attempts to predict how the community will react when potentially disruptive elements are introduced into the environment.

1.2. Previous work

Previously we have examined architectures for e-commerce systems (Henderson, 1998, 2002; Henderson and Walters, 2001), to investigate how federations of applications co-operate. We have also investigated the use of a fixed-length tournament-based approach to judge the fitness of negotiation algorithms against each other (Henderson et al., 2003). Certain patterns of negotiation were observed during the tournament between various algorithms, where certain pairs of algorithms did consistently better with each other than others. Examining their negotiation traces, we observed that stability would often emerge in their negotiations; their behaviour following a predictable path until negotiation was positively or negatively concluded. However, due to the nature of the experiment, negotiation between two participants did not affect other participants during a simulation. This meant that we were unable to investigate how stability would emerge at a *communal* level. Essentially, the environment was incapable of answering some interesting questions. Given a community of algorithms, would stability emerge? If so, in what form? Then, if a stability were to emerge in a community of negotiators, how would this stability be affected if we: introduce or extract an algorithm mid-experiment? Introduce an unreliable environment? Adapt algorithms to cope with this environment?

In this paper, we attempt to explore, and to some extent answer, the above questions. We extend the fixed-length tournament approach to encompass the concept of a continuously operational environment where negotiators may join and leave this community at any time. In our implementation of such an environment we are able to develop new algorithms using the framework described in this paper, then introduce, observe and evaluate them as they participate in negotiations with others. We are also able to introduce uncertainty into the message-passing infrastructure, and observe how this affects the participants. Of particular interest is how these changes affect the stability of negotiation communities.

2. Reactive and proactive negotiation

In essence, the negotiation process consists of a number of offers being exchanged between two participants until agreement is reached. This process consists of the following steps: wait until an offer is received, evaluate the new offer, then either reply with a counter-offer (going back to the first step), or send an acceptance or quit negotiations.

Notwithstanding the initial offer from either of the participants, this process is a reactive cycle. This process is depicted in Fig. 1. The dashed box at the top represents an initial action conducted by only one of the participants. This is the only proactive task in the process.

It is natural to assume that the structure of algorithms should follow this same rigid process. This idea is also easily extended to allow multiple negotiations with multiple participants.

In practice, adopting a purely reactive approach to the negotiation process is simply not sufficient. Developing negotiators in such a way does provide clarity of process, and simplicity of implementation. However, the success of the negotiator becomes ultimately dependent on the success of the negotiation process, which is itself dependent on the reliability of the operating environment. Notwithstanding 'bad' behaviour exhibited by negotiators, when this environment becomes unstable, the negotiation process is liable to collapse.

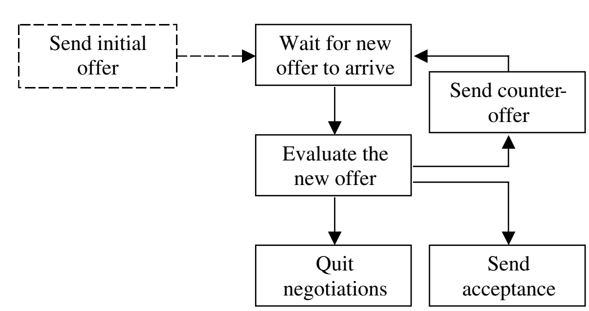

Fig. 1. Reactive negotiation process.