

ENHANCING COMMITMENT MACHINES

W. MANSFIELD

Supervisor: H. JACOBSON

Honours Thesis

School of Computer Science and Information Technology
RMIT University
Melbourne, AUSTRALIA

July, 2004

Abstract

Agent interaction protocols are usually specified in terms of permissible sequences of messages. This representation is, unfortunately, brittle and does not allow for flexibility and robustness. The *commitment machines* framework of Yolum and Singh aims to provide more flexibility and robustness by defining interactions in terms of the commitments of agents. In this thesis we identify a number of areas where the commitment machines framework needs improvement and propose an improved version. In particular we improve the way in which commitments are discharged and the way in which pre-conditions are specified.

Contents

1	Introduction	3
2	Citation commands	3

1 Introduction

Communications between software agents are typically regulated by interaction protocols. These include general communication protocols, such as the auction protocol and the contract net protocol, as well as more specific protocols such as the NetBill payment protocol (Ye, 2003; Xiao and Dunham, 2001). Traditional protocol representations such as Finite State Machines (FSM), Petri-Nets (Pei et al., 2001) and AUML sequence diagrams often specify protocols in terms of legal message sequences. Under such protocol specifications, agent interactions are pre-defined and predictable. The inevitable rigidity resulting from this prevents agents from taking opportunities and handling exceptions in a highly dynamic and uncertain multi-agent environment.

Commitment Machines proposed by Xiao and Dunham (2001) define an interaction protocol in terms of actions that change the state of the system, which consists of the state of the world as well as the *commitments* that agents have made to each other. It is a commitment made to an interaction partner which makes an agent perform its next action. In other words, an agent acts because it wants to comply with the protocol and provide the promised outcomes for another party. Actions not only change the values of state variables, but also may initiate new commitments and/or discharge existing commitments. In traditional protocol representations, agents are constrained to perform a pre-defined sequence of actions, whereas in CMs, an agent is able to reason about what action should be taken next in accordance with the dynamics of the environment and the management of its commitments in that environment. This fundamentally changes the process of specifying a protocol from a procedural approach (i.e. prescribing *how* an interaction is to be executed) to a declarative one (i.e. describing *what* interaction is to take place).

Another advantage of the CM approach is that it provides a natural means of managing multi-agent interactions. Agent programming concepts are often discussed in the context of a single agent situated in an environment, discussing properties such as autonomy, pro-activeness, reactivity and social awareness. The CM approach enables pro-activeness and reactivity to be discussed in a multi-agent context.

CMs thus allow interactions between agents to be organized in a manner which is more flexible and robust than an approach based on pre-defined sequences. For example, in the NetBill protocol, a customer may wish to order goods without first receiving a quotation, or a merchant may be happy to send goods to a known reliable customer with less rigorous checking than normal.

In this thesis we identify a number of areas where the Commitment Machine framework can be improved. Specifically, we show how the identification of undesirable states (such as omitting to provide a receipt, or receiving the goods before payment has been confirmed) can be incorporated into the design process in order to achieve acceptable outcomes for a wider variety of circumstances than is done in (Ye, 2003; Xiao and Dunham, 2001). We also show how certain anomalies in discharging commitments and in handling pre-conditions can be remedied.

2 Citation commands

Citations in different styles. With the `natbib` package, you can use different citation commands. The `citep` command gives author and year (Ye, 2003), The `cite` command gives author and year for use in text, for example Ye (2003) wrote the data mining handbook.

References

- Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., and Yang, D. (2001). H-mine: Hyper-structure mining of frequent patterns in large databases. In *Proc. of ICDM 2001*.
- Xiao, Y. and Dunham, M. H. (2001). Efficient mining of traversal patterns. *Data and Knowledge Engineering*, 39(2):191–214.
- Ye, H. (2003). *Handbook of data mining*. Lawrence Erlbaum Associates.