**Title and Source**

"NONMONOTONIC TEMPORAL REASONING" (U)

**Author(s)**

Dr. Yoav Shoham

**Performing Organization Name(s) and Address(es)**

Stanford University
Dept of Computer Science
Stanford CA 94305

**Sponsoring/Monitoring Agency Name(s) and Address(es)**

AFOSR/NM
Bldg 410
Bolling AFB DC 20332-6448

**Abstract**

In research carried out under this grant, they investigated a number of issues, semantical and algorithmic, in the design of agents in a multi-agent environment. The issues that were investigated included the structure of agents (which they called "mental state"), the flow of control of agents' activities over time, a particular programming language geared towards controlling agents, and a number of subsidiary computational problems. The researchers have developed a computational framework called agent oriented programming. AOP can be viewed as a specialization of object oriented programming (OOP). The state of an agent consists of components called beliefs, choices, capabilities, commitments, and possibly others; for this reason the state of an agent is called its mental state. The mental state of agents is captured formally in an extension of standard epistemic logics: besides temporalizing the knowledge and belief operators, AOP introduces operators for commitment, choice and capability. Agents are controlled by agent programs, which include primitives for communicating with other agents. In the spirit of speech-act theory, each communication primitive is of a certain type: informing, requesting, offering, and so on.
Report on results from research carried out under AFOSR grant AFOSR-89-0326 on

Nonmonotonic Temporal Reasoning

by

Yoav Shoham
Computer Science Department
Stanford University
Stanford, CA 94305
March 20, 1992

1 Overview

In research carried out to this date under this grant we investigated a number of issues, semantical and algorithmic, in the design of agents in a multi-agent environment. The issues we investigated included the structure of agents’ state (which we called ‘mental state’), the flow of control of agents’ activities over time, a particular programming language geared towards controlling agents, and a number of subsidiary computational problems.

2 Summary of previous results

We have developed a computational framework called agent oriented programming. AOP can be viewed as an specialization of object oriented programming (OOP). The state of an agent consists of components called beliefs, choices, capabilities, commitments, and possibly others; for this reason the state of an agent is called its mental state. The mental state of agents is captured formally in an extension of standard epistemic logics: beside temporalizing the knowledge and belief operators, AOP introduces operators for commitment, choice and capability. Agents are controlled by agent programs, which include primitives for communicating with other agents. In the spirit of speech-act theory, each communication primitives is of a certain type: informing, requesting, offering, and so on.
The relationship between AOP and OOP can be summarized in the following table:

<table>
<thead>
<tr>
<th>Framework:</th>
<th>OOP</th>
<th>AOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic unit:</td>
<td>object</td>
<td>agent</td>
</tr>
<tr>
<td>Parameters defining state of basic unit:</td>
<td>unconstrained</td>
<td>beliefs, commitments, capabilities, choices, ...</td>
</tr>
<tr>
<td>Process of computation:</td>
<td>message passing and response methods</td>
<td>message passing and response methods</td>
</tr>
<tr>
<td>Types of message:</td>
<td>unconstrained</td>
<td>inform, request, offer, promise, decline, ...</td>
</tr>
<tr>
<td>Constraints on methods:</td>
<td>none</td>
<td>honesty, consistency, ...</td>
</tr>
</tbody>
</table>

The design of the generic agent interpreter may be depicted graphically as follows:

A detailed discussion of AOP appears in [7]; this article has been submitted for publication. We have implemented an agent interpreter; it is documented in [13], and also described in [8]. Ongoing collaboration with the Hewlett Packard corporation is aimed at incorporating features of AOP in the New Wave™ architecture.

Preliminary ideas on the logic of mental state appear in [12]; a concrete proposal is made in [1]. This latter work addresses the properties of mental state – beliefs, commitments and capabilities – at a given moment. Other publications address dynamic aspects of mental state. A logic for perfect memory and justified learning is discussed in [5]. [1] addresses the logic of belief revision; specifically, the postulates of belief update, which have been mentioned in the database and AI literature, are shown to be derivable from a formal theory of action, rather than arbitrarily stated. The theory used there is the ‘provably correct’ theory presented in [3],
which was later generalized to a framework admitting concurrent action [4].

In parallel to the logical aspects of action and mental state, we have investigated algorithmic questions. We have proposed a specific mechanism for tracking how beliefs change over time, called temporal belief maps [2]. This mechanism generalizes the functionality of so-called time maps. The following figure depicts two simple 2-dimensional temporal belief maps; the horizontal axis is the time of belief, and the vertical axis the time to which the belief refers.

![Temporal belief maps](image)

Figure 1: Consistent default regions ($t_1 \neq t_2$, $t_1 \neq t_2$)

We have also begun to investigate ways in which multiple agents can function usefully in the presence of other agents. In [6] we propose the mechanism of protograms to balance conflicting influences of different agents. We are also interested in minimizing such conflicts in the first place, and have been investigating the computational utility of social law. In [10] we study the special case of traffic laws in a restricted robot environment; in [9] we propose a general framework for representing social laws within a theory of action, and investigate the computational complexity of automatically synthesizing useful social laws.

References


