NoA - A Normative Agent Architecture

Martin J. Kollingbaum and Timothy J. Norman
Department of Computing Science
University of Aberdeen
Aberdeen, AB24 3UE, UK
mkolling@csd.abdn.ac.uk
tnorman@csd.abdn.ac.uk

Abstract

NoA is an agent architecture that supports the development of agents motivated by norms: obligations, permissions and prohibitions. Obligations motivate a normative agent to act: a motive to achieve a state of affairs or to perform some action. Prohibitions restrict an agent's behaviour, whereas permissions allow an agent to pursue certain activities. To test the architecture, NoA agents arc applied to automated business transaction scenarios where the correct execution of contracts is paramount to create a situation of trust.

1 Introduction

The NoA normative agent architecture is designed to support the development of agent societies where agents are motivated by norms: obligations, permissions and prohibitions. The development of the NoA agent architecture is driven by the particular needs of automated business transactions within electronic commerce environments. However, NoA is designed as an architecture that is generally applicable for agent system development. The specific concern of the NoA architecture is to provide means for the development of norm-motivated practical reasoning agents. This architecture is determined by two main elements: The NoA language for the specification of plans and norms and the NoA interpreter, which is capable of interpreting and executing plan and norm specifications formulated in the NoA language. The NoA architecture and the NoA specification language for norms and plans is influenced by systems such as Agentspeak(L) [Rao, 1996]. There are three principal distinctions, however, between these systems and NoA:

- *Multiple effects.* In common with planning domain specification languages such as PDDL [PDDL], but in contrast to languages such as JAM [Huber, 1999], the NoA plan specification language allows *all* the effects of a plan to be declared. Any of these effects can be the reason for the agent to select a plan. This provides greater flexibility in the specification of agent capabilities, and enables a NoA agent to reason about the side-effects of executing a plan.
- *Motivated by norms.* NoA agents arc motivated by norms rather than desires and intentions. These norms capture states of affairs or actions the agent is obliged or permitted to achieve or perform, or prohibited from achieving or performing.
- *Distinction between states and actions.* The norms governing the behaviour of a NoA agent refer to either actions that are obligatory, permitted, forbidden, or states of affairs that are obligatory, permitted or forbidden [Norman and Reed, 2001].

2 The NoA Language

The main elements that influence the behaviour of a NoA agent are (a) a set of beliefs, (b) a set of pre-specificd plans and (c) a set of norms. These concepts are reflected in the NoA specification language. The general form of a NoA plan carries information about when it would be allowed to select it for execution (preconditions), what kind of states of affairs it can achieve (effects) and the actual behaviour specification as the body of the plan.

```
plan stack ( X, Y )
  preconditions ( ontable ( X ),clear ( Y ) )
  effects ( on ( X, Y ),not ontable ( X ),
            not clear ( Y ) )
{
  achieve clear ( X ) ;
  primitive doMove ( X, Y ) ;
```

Figure 1. A NoA Plan

Figure 1 shows an example plan, usually applicable in a blocks world scenario for stacking blocks. NoA plans are chosen for achieving a state of affairs according to their effects (see the effects list in figure 1). For the performance of an action, plans arc chosen according to their signature (name and parameter list). The plan body in this example shows language constructs to establish a subgoal (achieve clear (X)) and to perform primitive actions.

```
obligation ( BlockMover,
             achieve ontable( a ),
             on ( a, b ),
             clear ( b ) and ontable( a )
)
```

Figure 2. A NoA norm specification

Figure 2 shows a norm specification: an obligation for agent BlockMover to achieve the placement of a block *a* on the table (again in an imagined blocks world). Each normative statements carries information under what circumstances such a norm is "active" and therefore relevant to an agent. Normative statements include activation and expiration conditions that determine when a norm is active and when it expires.

3 The NoA Interpreter

The behaviour of a NoA agent is determined by norms and plans. Norms arc relevant to an agent only if they are "active". Norms become active when the agent's beliefs reflect their specified conditions of activation. Activated norms influence the behaviour of an agent in two ways, (a) by motivating the generation of a goal that must be achieved or an action that has to be performed, these are externally motivated activities, and (b) by being used in a special "norm filtering" step within the execution cycle of the NoA architecture by restricting the options of the agent in fulfilling its responsibilities.
Plans arc instantiated when their preconditions hold. From this set of plan instances, a set of plan options has to be selected. This selection takes place according to one of their effects, whereas all other effects represent side-effects of this plan, which occur during its execution. All the effects of a plan must be taken into account when it is considered for execution, e.g. a plan selected for satisfaction of one obligation may have a side-effect that is contrasting other norms.

```
while state / action requested
    if achievement of state is requested
        then plan_options = choosc_plan_.instances (state)
        else plan_options - retrieve _j)lan (action)
    endif
    filtered..plans - norm_filter (plan .options)
    if filtered,plans not empty then
        plan - deliberate (filtered_plans)
        execute (plan)
    endif
endwhile
```

Figure 3. Execution cycle of the NoA interpreter

The execution cycle of the NoA interpreter is shown in figure 3. It is assumed that a change in the agent's beliefs activates / de-activates (a) obligations that motivate the achievement of a state of affairs or the performance of an action, (b) prohibitions and permissions relevant for the norm filtering step, and (c) instantiates plans. In case that a state of affairs must be achieved, a set of plan instances is chosen according to one of the effects they would produce if they would be executed (see figure 3). From the set of currently instantiated plans (because their preconditions hold) the set of plan options is chosen. In case that the direct performance of an action is required, one specific plan instance is chosen. Active norms such as prohibitions and permissions are applied to this set of plans to filter out those that would produce effects that are forbidden or arc directly acts that are forbidden. One of the plans must be chosen for execution. The execution of a plan can have various outcomes: performing a primitive action affecting the world, updating beliefs or establishing a subgoal or subsidiary actions.

5 Conclusion

In this paper, the NoA normative agent architecture has been presented as a framework that supports the development of agent societies where agents arc motivated by norms. This architecture introduces norms as external motivators for an agent to achieve states of affairs or perform actions. It provides agents with capabilities of decision-making driven by such norms expressing the obligations, permissions and prohibitions of an agent. The NoA normative agent architecture is based on three key concepts: (a) a clear distinction between an agent taking responsibility for the achievement of a state of affairs and the agent taking responsibility for the performance of an action, (b) agents understand normative statements within contracts and how to correctly execute them and (c) goal / action generation and plan selection is based on norms.

References

[Kollingbaum and Norman, 2002J Martin J. Kollingbaum and Timothy J. Norman. Supervised Interaction - creating a Web of Trust for Contracting Agents in Electronic Environments. In C. Castelfranchi & W. Johnson (eds), AAMAS 2002, ACM Press, New York, pages 272-279, 2002.

[Gcorgcff and Lansky, 1987] Michael P. Georgeff and Amy L. Lansky. Reactive Reasoning and Planning. In *Proceedings ofAAAI-87,* pp. 677682, 1987.

[Huber, 1999] Marcus Huber. JAM: A BDI-theoretic mobile agent architecture. Agcnts'99, pp. 236-243, (May 1999).

[Rao, 1996] Anand S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. LNAI 1038, pp. 42-55. Springer, (1996).

[Norman and Reed, 2001] Timothy J. Norman and Chris Reed. Delegation and Responsibility. In C. Castelfranchi, Y. Lespcrance (eds), Intelligent Agents VII: *Proceedings ATAL 2001,* LNAI 1986, pp. 136-149, 2001.

[PDDL]http://www.dur.ac.Uk/d.p.long/IPC/pddl.html