A rule language for modelling and monitoring social expectations in multi-agent systems

Stephen Cranefield
Department of Information Science, University of Otago
PO Box 56, Dunedin, New Zealand
scranefield@infoscience.otago.ac.nz

Abstract

This paper proposes a rule language for defining social expectations based on a metric interval temporal logic with past and future modalities and a current-time binding operator. An algorithm for run-time monitoring compliance of rules in this language based on formula progression is also outlined.

1 Introduction

The study of *electronic institutions*—explicit declarative models of the rules governing particular open systems of autonomous agents—has gained much recent attention [Cortés, 2004]. There has been a significant amount of research on statically verifying properties of institutions as well as interpreting institutions to manage or guide agent interaction. However, there has been little attention paid to mechanisms for run-time compliance checking, i.e. monitoring events in a running agent system, determining the future expectations of agents' behaviour according to norms of the institution, and checking if these are fulfilled or violated. This paper focuses on this issue by presenting (i) a logic named hyMITL$^{\pm}$ that combines CTL$^{\pm}$ [Verdicchio and Colombetti, 2003] with Metric Interval Temporal Logic (MITL) [Alur *et al.*, 1996], as well as features of hybrid logics [Blackburn *et al.*, 2001], and (ii) a subset of hyMITL$^{\pm}$ that provides a rule language for defining social expectations. We have implemented an algorithm for monitoring social expectations encoded in this language, based on the technique of formula progression from the planning system TLPlan [Bacchus and Kabanza, 1998].

2 Syntax of hyMITL$^{\pm}$

The syntax of hyMITL$^{\pm}$ is defined as follows:

$$
\begin{aligned}
\phi \ &::= p \mid \neg\phi \mid \phi \wedge \phi \mid \forall x.\phi_x \mid \mathsf{X}^{+}\phi \mid \mathsf{X}^{-}\phi \mid \\
&\quad\ \phi\,\mathsf{U}^{+}_{I}\,\phi \mid \phi\,\mathsf{U}^{-}_{I}\,\phi \mid \mathsf{A}\,\phi \mid \mathsf{E}\,\phi \mid \ {\downarrow^{u}} x.\phi_x \mid I \\
I \ &::= (-\infty,+\infty) \mid [b,b] \mid [b,b) \mid (b,b] \mid (b,b) \\
b \ &::= a \mid +d \mid -d
\end{aligned}
$$

where

- p is an atomic formula from a first order language L.
- ϕ_x denotes a formula ϕ in which variable x is free (i.e. not bound by $\forall$ or $\downarrow$).
- a and d are terms, possibly containing variables, that denote (respectively) absolute points in time and durations (e.g. using the language proposed by Verdicchio and Colombetti [2004]).
- u is a unit selector on the $\downarrow$ binding operator, referring to the desired granularity of time (e.g. year or minute) for binding x to the current time (with default value now).

We constrain the use of variables within interval bounds b: any such variables must be bound by an enclosing $\downarrow$ operator.

In this logic, the temporal operators X^{+} and X^{-} mean "in the next state" and "in the previous state" respectively. $\phi\,\mathsf{U}^{+}_{I}\,\psi$ asserts that ϕ will remain true from the current state for some (possibly empty) sequence of consecutive future states, followed by a state that is within the time interval I and for which ψ holds. X^{-} and U^{-}_{I} are defined similarly, but in the past direction.

The bounds of intervals can be specified either relatively or absolutely—a prefix of "+" or "−" indicates a relative time value. When qualifying U^{-}, the interval bounds are written in the reverse order from usual, e.g. $[-2\,\mathsf{hours}, -3\,\mathsf{hours}]$.

A and E are temporal path quantifiers. They assert that the formula that follows the operator applies to all, or respectively at least one, of the possible sequences of states passing through the current state.

The $\downarrow$ operator is based on the "binder" operator used in hybrid logics [Blackburn *et al.*, 2001]. It binds a variable to a term denoting the current date/time, possibly 'rounded down' to a particular degree of precision, e.g. to the start of the current year, month or day, depending on the unit indicated by the superscript u.

We use the usual derived operators of first order predicate logic, as well as the standard abbreviations for existential and universal quantification over states in a path: $\mathsf{F}^{+}_{I}\phi \equiv \mathsf{true}\,\mathsf{U}^{+}_{I}\,\phi$ and $\mathsf{G}^{+}_{I}\phi \equiv \neg\mathsf{F}^{+}_{I}\neg\phi$, with similar definitions for F^{-}_{I} and G^{-}_{I}. We define future and past "weak until" operators in the following way:

$$\phi\,\mathsf{W}^{+}_{[l,u]}\,\psi \ \equiv {\downarrow} t.(\mathsf{G}^{+}_{[t,u]}\phi \vee \phi\,\mathsf{U}^{+}_{[l,u]}\,\psi)$$

with similar definitions for intervals with open bounds and for W^{-}.

$$\mathsf{AG}^+\ (\mathrm{Done}(c,\ \mathrm{make\_payment}(c, p, \mathit{amount}, \mathit{prod\_num}))\ \wedge\ [t,\ t{+}1\,\mathsf{week})\ \rightarrow$$
$$\downarrow^{\mathsf{week}} \mathrm{w}.((\neg\, \mathsf{F}^-_{[-0,\ \mathrm{w}]}\mathrm{Done}(p,\ \mathrm{send\_report}(c, \mathit{prod\_num}, \mathrm{w})) \rightarrow \mathsf{F}^+_{(+0,\ \mathrm{w}+1\,\mathsf{week})}\mathrm{Done}(p,\ \mathrm{send\_report}(c, \mathit{prod\_num}, \mathrm{w})))$$
$$\mathsf{W}^+_{[+0,\ \mathrm{w}+52\,\mathsf{weeks}]}$$
$$\mathrm{Done}(c,\ \mathrm{cancel\_order}(c, p, \mathit{prod\_num})))$$

Figure 1: A rule expressing the terms of service offered by agent p

3 The Rule Language

We identify a subset of hyMITL$^\pm$ that is suitable for encoding social expectations in a form that can be used in run-time compliance testing: the set of all formulae of the following form:

$$\mathsf{AG}^+\, \forall_{1 \le i \le n} x_i.(\phi \rightarrow \psi)$$

for $n \ge 0$, where ϕ and ψ are linear-time formulae (i.e. they do not contain A or E), the free variables of ϕ and ψ are precisely the quantified variables $\{x_1, \ldots, x_n\}$, ϕ and ψ do not contain any occurrences of $\forall$, and any occurrence of $\exists$ must be of a certain restricted form that is equivalent to TLPlan's bounded existential quantification [Bacchus and Kabanza, 1998].

Figure 1 shows an example rule encoding a service offer by an agent *p* that can provide weekly reports on a particular market for an annual fee. This rule states that once payment of the required amount is made by the potential client *c* within the validity period of the offer (one week), the service-providing agent is committed to sending a report to the client once a week for 52 weeks or until the client cancels the order.

4 The Compliance Monitoring Process

Rules in our language are intended to be used in the following compliance-testing process:

> Given a current state and the history of all prior states and their associated times, for each rule, match the left hand side against the current state and history, resulting in a set of instances of the right hand side. Add these instances to the set of current expectations, then check all expectations to see which are fulfilled or violated. Any expectations that cannot yet be evaluated because they involve future states are 'progressed' to the next state when it is created by an event observation.

The progression of unfulfilled expectations from one state to the next uses a modified version of the progression algorithm of Bacchus and Kabanza [1998]. This algorithm generates a formula expressing what needs to be true in the new state if the input expectation was required to be true in the previous state, but was not yet able to be evaluated there.

Once new expectations have been computed by matching the rules to the current state and history, each formula in the combined set of old and new expectations is partially evaluated, resulting in true or false if the truth of the formula can be determined yet, and otherwise returning a formula equivalent to the original one (given the facts in the history states), but modified where possible to make progression and future evaluation easier. This partial evaluation is separated out from the progression step as it is performed in the current state—before a new state is created by a new event occurrence, and therefore before formula progression can be done (this needs the time difference between the old and new state as an input).

5 Conclusion

This paper has defined a rule language for defining social expectations based on a metric interval temporal logic and has outlined an algorithm that can be used at run time in a multi-agent system to monitor when expectations are generated, fulfilled and violated. A prototype implementation of the compliance testing algorithm has been implemented using SWI Prolog. Most of the features described here have been implemented, although currently the system is not connected to an agent—it uses a static database of states and their facts—and integers are used to represent times.

The semantics for hyMITL$^\pm$, the algorithms used, a discussion of related work and future extensions to this work, and acknowledgements can be found in the full version of this paper [Cranefield, 2005].

References

[Alur *et al.*, 1996] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM*, 43(1):116–146, 1996.

[Bacchus and Kabanza, 1998] F. Bacchus and F. Kabanza. Planning for temporally extended goals. *Annals of Mathematics and Artificial Intelligence*, 22(1-2):5–27, 1998.

[Blackburn *et al.*, 2001] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, chapter 7, pages 436–447. Cambridge University Press, 2001.

[Cortés, 2004] U. Cortés. Electronic institutions and agents. *AgentLink News*, 15:14–15, September 2004.

[Cranefield, 2005] S. Cranefield. A rule language for modelling and monitoring social expectations in multi-agent systems. Discussion Paper 2005/01, Department of Information Science, University of Otago, 2005. http://www.otago.ac.nz/informationscience/pubs/papers/papers/dp2005-01.pdf.

[Verdicchio and Colombetti, 2003] M. Verdicchio and M. Colombetti. A logical model of social commitment for agent communication. In *Proc. of AAMAS 2003*, pages 528–535. ACM Press, 2003.

[Verdicchio and Colombetti, 2004] M. Verdicchio and M. Colombetti. Dealing with time in content language expressions. In *Proc. of the AAMAS 2004 Workshop on Agent Communication*, pages 90–104, 2004.