Distributed States Logic

Carlo Montangero Laura Semini
Dipartimento di Informatica, Università di Pisa, {monta,semini}@di.unipi.it

Abstract

We introduce a temporal logic to reason on global applications. First, we define a modal logic for localities that embeds the local theories of each component into a theory of the distributed states of the system. We provide the logic with a sound and complete axiomatization. Then, we extend the logic with a temporal operator. The contribution is that it is possible to reason about properties that involve several components in a natural way, even in the absence of a global clock, as required in an asynchronous setting.

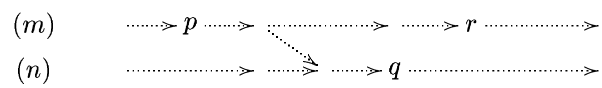
1. Introduction

The current trend towards global computing needs software that works in an open, concurrent, distributed, high–latency, security–sensitive environment. Besides, this software must be reliable, scalable, and "shipped today". In response to the challenges posed by so demanding requirements, there is an increasing interest in the seamless integration of asynchronous communication in programming, coordination, and specification languages. Indeed, message–passing, event–based programming, call–backs, continuations, dataflow models, workflow models etc. are ubiquitous in global computing. Notable examples in this direction are the delegate–based asynchronous calling model of the Microsoft .NET Common Language Runtime libraries, and *chords* in Polyphonic C#. As another example, Oikos_tl [14] deals with asynchronous communications in coordination and specification languages.

As a contribution to the response to the new challenges, we are developing YALL [10], an extension of temporal logic to deal with distributed systems. YALL has operators to name system components and to relate, causally, properties holding in distinguished components, in an asynchronous setting. A typical YALL formula might be:

$$\mathsf{m}\,p \;\text{LT}\; \mathsf{n}\,q \wedge \mathsf{m}\,r \qquad (1)$$

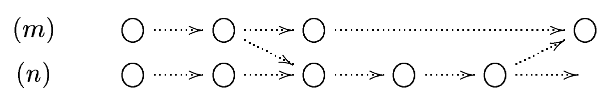
where operator LT is similar to Unity's $\mapsto$ (leads to) [2], and m and n express locality. Formula (1) says that a property p holding in component m, causes properties q and r to hold in future states of components n and m, respectively. An example is the computation below, where the oblique arrow denotes a communication.



We proceed in two steps. First, we define DSL (Distributed States Logic), a modal logic for localities that embeds the theories describing the local states of each component into a theory of the distributed states of the system. There is no notion of time or state transition at this stage. DSL has a sound and complete axiom system. Then, we define YALL by adding temporal operators. Since DSL carries over all meaningful propositional rules, like *and* simplification, in such a way that they can be exploited orthogonally to temporal operators, the exploitation of the local theories becomes smooth and robust, while proving distributed properties. The final contribution is that in YALL it is easy to reason about properties that involve several components, even in absence of a global clock, as required in an asynchronous setting.

The major problem with DSL is the frame structure. The usual choices to build a Kripke model for formulae like (1) are to consider the set of worlds W to be:

i) the set of the states of a computation, i.e. the union of all the states of the system components, like the circles in the following figure. This approach was taken in [11, 5].



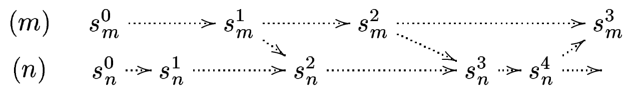
The problem of this choice is that it is not possible to reason on logical relations between formulae like the premises or the consequences of (1). In particular, a formula like

$$(\mathsf{n}\,q \wedge \mathsf{m}\,r) \rightarrow \mathsf{n}\,q \qquad (2)$$

which would permit to weaken the consequences of (1) would not be a legal formula, since no world can satisfy the conjunction $\mathsf{n}\,q \wedge \mathsf{m}\,r$.

ii) the set of global states, or snapshots, of the system, where each world is a set of states, one for each component. These sets must satisfy some constraints to be coherent with the communications between the subsystems.

Proceedings of the Ninth International Symposium on Temporal Representation and Reasoning (TIME'02)
1530-1311/02 $17.00 © 2002 **IEEE**



Examples of worlds are $\{s_m^i, s_n^j\}_{0\leq j\leq 2}^{i=0,1}$, while $\{s_m^2, s_n^1\}$ would not be a legal world.

This choice is not well suited in the case of asynchronous communication. Think of the case of property p holding *only* in state s_m^1 and q holding *only* in states s_n^j, for $0 \leq j \leq 4$. The following formula would be valid in the model

$$\mathsf{m}\, p \rightarrow \mathsf{n}\, q \tag{3}$$

inferring a remote knowledge which is meaningless in an asynchronous setting. Moreover, it would be natural to say that world $\{s_m^2, s_n^3\}$ follows $\{s_m^1, s_n^2\}$. In this case, one could assert that $\mathsf{n}\, p$ LT $\mathsf{m}\, q$ holds, if p and q hold in s_n^2 and s_m^2, respectively, even though no causal relationship exists between these two states. Similar problems arise if we use most logics for distributed systems (see, for instance [8, 15, 13, 3]), where components communicate via some form of synchronization and, therefore, it is not possible to express the asymmetric nature of causality we are interested in.

As shown in the next sections, we can get the desired properties by using the power–set of the set of all system states as the semantic domain of DSL. This choice, together with an appropriate next–state relation, makes YALL a very expressive language, that fully meets the pragmatic expectations of a designer.

2. DSL

We assume a countable set of propositional letters $P = \{p, q, \ldots\}$. The DSL formulae over a finite set of components $\Sigma = \{m_1, m_2, \ldots, m_k\}$ are defined by:

$$F \quad ::= \quad p \mid \bot \mid \sim F \mid F \wedge F' \mid \mathsf{m_i} F$$

where $\bot$ is the propositional constant *false*, and $\mathsf{m_i}$ for $i = 1 \ldots k$ are unary location operators. With $\bar{\mathsf{m}}_\mathsf{i}$ we denote the dual of $\mathsf{m_i}$, i.e., $\bar{\mathsf{m}}_\mathsf{i} F \equiv \sim \mathsf{m_i} \sim F$. With $\top$ we denote *true*.

Semantics. A model $\mathcal{M}$ for DSL formulae is a tuple $(W, R_1, \ldots, R_k, V)$, with u, v, w ranging over W. The reachability relations R_i satisfy the following conditions:

$$(u, v) \in R_i \quad \rightarrow \quad (v, v) \in R_i \tag{4}$$

$$(u, v) \in R_i \quad \rightarrow \quad (v, w) \in R_i \rightarrow v = w \tag{5}$$

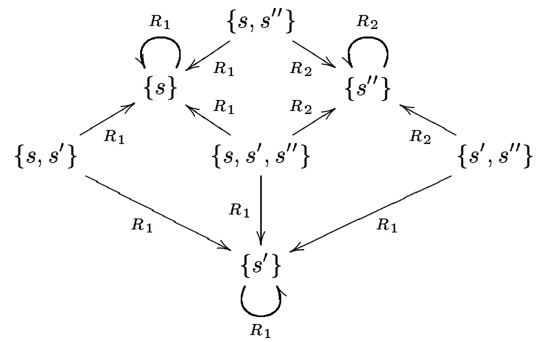
$$(u, v) \in R_i \quad \rightarrow \quad \nexists w.\, (v, w) \in R_j \text{ for } j \neq i \tag{6}$$

The semantics of DSL formulae is given by:

$(\mathcal{M}, u) \models \top$
$(\mathcal{M}, u) \models p$ iff $p \in V(u)$
$(\mathcal{M}, u) \models \sim F$ iff not $(\mathcal{M}, u) \models F$
$(\mathcal{M}, u) \models F \wedge F'$ iff $(\mathcal{M}, u) \models F$ and $(\mathcal{M}, u) \models F'$
$(\mathcal{M}, u) \models \mathsf{m_i} F$ iff $\exists v.\, (u, v) \in R_i$ and $(\mathcal{M}, v) \models F$

Let S_i be the set of states of component m_i, with $S_i \cap S_j = \emptyset$ for $i \neq j$, $S = \bigcup_k^{i=1}, S_i$, $DS = 2^S$, and ds, $ds' \in DS$. The frame $(DS, R_1, \ldots, R_k)$, where $(ds, ds') \in R_i$ if and only if ds' is a singleton set $\{s\}$, with $s \in S_i \cap ds$, satisfies the conditions (4)–(6) above. We call these frames, frames on DS, and call DS the set of *distributed states*, from which the name of the logic DSL. The frames on DS play a central role in the paper, since they are used to build the models for YALL formulae. Some examples follow.

Examples. Let the set DS be built on $S_1 = \{s, s'\}$ and $S_2 = \{s''\}$, then the frame on DS can be represented as:



For the sake of readability, we often let m, n range over Σ, with $m \neq n$, and use S_m, S_n, m and n .

If we take $s \in S_m$, $s' \in S_n$, $V(\{s\}) = \{p\}$, $V(\{s'\}) = \{q\}$, then the distributed state $\{s, s'\}$ satisfies $\mathsf{m}p \wedge \mathsf{n}q$.

The implication $\mathsf{m}\, F \wedge F' \rightarrow \mathsf{m}F \wedge \mathsf{m}F'$ holds, while the converse does not. Indeed, for $ds = \{s, s'\} \subseteq S_m$, and $V(\{s\}) = \{p\}$, $V(\{s'\}) = \{q\}$, we have $ds \models \mathsf{m}p \wedge \mathsf{m}q$, but not $ds \models \mathsf{m}\,(p \wedge q)$. In YALL, this non–equivalence is useful to specify that an event can have different future effects in a component, without constraining them to occur in the same state. Finally, $\mathsf{m}F \vee F' \leftrightarrow \mathsf{m}F \vee \mathsf{m}F'$.

The formula $\mathsf{mn}F$ is false. In fact, $ds \models \mathsf{mn}F$ if and only if there exists an $s \in S_n \cap S_m \cap ds$ such that $\{s\} \models F$, but S_m and S_n are disjoint. Conversely, $\mathsf{mm}F$ is satisfiable, and it is equivalent to $\mathsf{m}F$. The formula $\mathsf{m}\top$ is satisfied by all the distributed states ds such that $ds \cap S_m \neq \emptyset$.

Axiom system. DSL has the following axiomatization.

PC	axioms of the propositional calculus
K	$\bar{\mathsf{m}}(F \rightarrow F') \rightarrow (\bar{\mathsf{m}}F \rightarrow \bar{\mathsf{m}}F')$
DSL1	$\bar{\mathsf{m}}(\bar{\mathsf{m}}F \leftrightarrow F)$
DSL2	$\bar{\mathsf{m}}\bar{\mathsf{n}}\bot$
MP	$\dfrac{F \quad F \rightarrow F'}{F'}$ **Nec** $\dfrac{F}{\bar{\mathsf{m}}F}$

Example. Some examples of formulae that can be derived from the axioms follow: $\bar{\mathsf{m}}F \rightarrow \bar{\mathsf{m}}\bar{\mathsf{m}}F$ (axiom 4), $\mathsf{mm}F \leftrightarrow \mathsf{m}F$, $\mathsf{m}(F \wedge F') \rightarrow (\mathsf{m}F \wedge \mathsf{m}F')$, $\bar{\mathsf{m}}(\mathsf{m}F \leftrightarrow F)$.

Soundness is easy to see. We prove completeness.

Completeness. Let $(W^{DSL}, R_1^{DSL}, \ldots, R_k^{DSL}, V^{DSL})$ be the canonical model for DSL. We recall that worlds in

W^{DSL} are maximal consistent sets of DSL formulae (DSL–MCS in the following), and that $(u, v) \in R_i^{DSL}$ if and only if $\bar{\mathsf{m}}_\mathsf{i} F \in u \;\; \rightarrow \;\; F \in v$. We need to show that, for all i, R_i^{DSL} satisfies conditions (4)–(6).

Cond (4). We prove: $(u, v) \in R_i^{DSL} \rightarrow (v, v) \in R_i^{DSL}$ Suppose $\bar{\mathsf{m}}_\mathsf{i} F \in v$. u is a DSL–MCS and hence (see DSL1) $\bar{\mathsf{m}}_\mathsf{i}(\bar{\mathsf{m}}_\mathsf{i} F \rightarrow F) \in u$. But $(u, v) \in R_i^{DSL}$, hence $(\bar{\mathsf{m}}_\mathsf{i} F \rightarrow F) \in v$. Thus, by modus ponens, $F \in v$.

Cond (5). We prove that $(u, v) \in R_i^{DSL}$ and $(v, w) \in R_i^{DSL}$ imply $v = w$.
It is sufficient to prove that $v \subseteq w$. In fact, v and w are DSL–MCS and it is not the case that $v \subset w$, thus $v = w$. Let $F \in v$. u is a DSL–MCS and hence (see DSL1) it includes $\bar{\mathsf{m}}_\mathsf{i}(F \rightarrow \bar{\mathsf{m}}_\mathsf{i} F)$. But $(u, v) \in R_i^{DSL}$, hence $F \rightarrow \bar{\mathsf{m}}_\mathsf{i} F \in v$. Thus, by modus ponens, $\bar{\mathsf{m}}_\mathsf{i} F \in v$. As $(v, w) \in R_i^{DSL}$, we conclude that $F \in w$.

Cond (6). We prove that $(u, v) \in R_i^{DSL}$ implies $\not\exists w.\ (v, w) \in R_j^{DSL}$, for $j \neq i$.
Assume $(v, w) \in R_j^{DSL}$. As u is a DSL–MCS, it includes $\bar{\mathsf{m}}_\mathsf{i}\bar{\mathsf{m}}_\mathsf{j} \bot$ (DSL2). As $(u, v) \in R_i^{DSL}$, then $\bar{\mathsf{m}}_\mathsf{j} \bot \in v$. As $(v, w) \in R_j^{DSL}$, then $\bot \in w$, which is an absurd.

3. Adding time: a fragment of YALL

YALL extends DSL adding temporal operators. We consider here only operator LT, that expresses a liveness condition, and is similar to Unity's $\mapsto$ (leads to).

$$\phi \quad ::= \quad F \mid F \ \mathrm{LT}\ F' \qquad\qquad \text{where } F,\, F' \in DSL.$$

Semantics. YALL models are built on structures like the one at point ii) above. The arrows between states denote transitions and communications, and define a causal dependency relationship. We introduce a partial order relation R^*, where $(s, s') \in R^*$ if and only if s' causally depends on s. For example, in the named structure, (s_m^0, s_m^1), (s_m^0, s_n^3), $(s_m^1, s_n^3) \in R^*$.

A model $\mathcal{M}$ is a tuple $(DS, R_1, \ldots, R_k, \leq, V)$, where

$$ds \leq ds' \quad \text{iff} \quad \begin{cases} \forall s \in ds,\ \exists s' \in ds'.\ (s, s') \in R^* \\ \forall s' \in ds',\ \exists s \in ds.\ (s, s') \in R^* \end{cases}$$

Let $\mathcal{M}$ be a model, and ds_0 the set of its initial states:

$$\begin{array}{l} \mathcal{M} \models_Y F \quad \text{iff} \quad \forall ds \geq ds_0.\ ds \models F \\ \mathcal{M} \models_Y F \ \mathrm{LT}\ F' \quad \text{iff} \quad \forall ds \geq ds_0. \\ \qquad\qquad\qquad\quad ds \models F \text{ implies } \exists\, ds' \geq ds.\ ds' \models F' \end{array}$$

where $\models$ is the DSL satisfiability relation.

Rules. We present the most useful rules of the logic. In the first rule (necessitation) we use $\vdash_{DSL}$ for the sake of comprehension.

$$\frac{(\vdash_{DSL})\ F}{F} \qquad \frac{F \rightarrow G}{F \ \mathrm{LT}\ G} \qquad \frac{G \rightarrow F \quad F \ \mathrm{LT}\ F' \quad F' \rightarrow G'}{G \ \mathrm{LT}\ G'}$$

$$\frac{F \ \mathrm{LT}\ G \quad G \ \mathrm{LT}\ H}{F \ \mathrm{LT}\ H} \qquad \frac{F \ \mathrm{LT}\ G \quad H \ \mathrm{LT}\ G}{F \vee H \ \ \mathrm{LT}\ G} \qquad \frac{F \ \mathrm{LT}\ G \quad F \ \mathrm{LT}\ H}{F \ \mathrm{LT}\ G \wedge H}$$

Discussion, examples, and comparison with related work (e.g. [1, 4, 6, 7, 9, 12]) can be found in [10].

Acknowledgments We gratefully thank Massimo Franceschet, Angelo Montanari, and Francesca Scozzari for interesting discussions on a draft of the paper. The work was supported by Projects Sahara and Degas.

References

[1] C. Areces. *Logic Engineering. The Case of Description and Hybrid Logics*. PhD thesis, Univ. of Amsterdam, 2000.

[2] K. Chandy and J. Misra. *Parallel Program Design: A Foundation*. Addison-Wesley, Reading Mass., 1988.

[3] H.-D. Ehrich, C. Caleiro, A. Sernadas, and G. Denker. Logics for specifying concurrent information systems. In *Logic for Databases and Information Systems*, pages 167–198. Kluver Academic Publishers, 1998.

[4] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[5] G. Ferrari, C. Montangero, L. Semini, and S. Semprini. Mark, a reasoning kit for mobility. *Automated Software Engineering*, 9(2):137–150, Apr 2002.

[6] S. Katz and D. Peled. Interleaving set temporal logic. *Theoretical Computer Science*, 75(3):263–287, 1990.

[7] Lodaya, Ramanujam, and Thiagarajan. Temporal logics for communicating sequential agents: I. *Int. Journal of Found. of Computer Science*, 3(2):117–159, '92.

[8] A. Masini and A. Maggiolo-Schettini. TTL: A formalism to describe local and global properties of distributed systems. *Theor. Informatics and Applic.*, 26(2):115–149, 1992.

[9] A. Montanari. *Metric and Layered Temporal Logic for Time Granularity*. PhD thesis, University of Amsterdam, 1996.

[10] C. Montangero and L. Semini. Distributed states logic. Technical Report TR-02-05, Dipartimento di Informatica, 2002. At www.di.unipi.it/ricerca/TR/tr.html".

[11] C. Montangero and L. Semini. Composing Specifications for Coordination. In *Proc. COORDINATION 99*, *LNCS* 1594, pages 118–133, 1999.

[12] Pinter and Wolper. A temporal logic for reasoning about partially ordered specifications. In *Proc. 3^{th} ACM Principles of Distributed Computing*, pages 28–37, 1984.

[13] R. Ramanujam. Locally linear time temporal logic. In *Proc. 11^{th} IEEE Symp. on Logic In Computer Science*, pages 118–127. IEEE Computer Society, 1996.

[14] L. Semini and C. Montangero. A Refinement Calculus for Tuple Spaces. *Science of Computer Programming*, 34:79–140, 1999.

[15] P. S. Thiagarajan and J. G. Henriksen. Distributed versions of linear time temporal logic: A trace perspective. In *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets*, *LNCS* 1491, pages 643–681, 1998.