Università di Pisa

Dipartimento di Informatica

Technical Report : TR-99-23

A Logic with Modalities for Asynchronous Distributed Systems

Laura Semini, Carlo Montangero

December 21, 1999

ADDR: Corso Italia 40,56125 Pisa,Italy. TEL: +39-50-887111. FAX: +39-50-887226

A Logic with Modalities for Asynchronous Distributed Systems

Laura Semini[1] and Carlo Montangero[2]

[1] Dip. di Sistemi e Informatica, Università di Firenze, semini@dsi.unifi.it
[2] Dip. di Informatica, Università di Pisa, monta@di.unipi.it

Abstract. The purpose of this work is to establish some foundational ground for the logics that can be used to reason on distributed systems where asyncronous message passing is the only possible mechanism for communication, e.g. distributed workflow systems.
We define *adtl*, a logic with modalities for time and locality.
We introduce a complete and decidable axiom system, which characterizes the class of structures that reflect the nature of the causal relation when communications are asyncronous.

1 Introduction

The design of quality software for distributed systems is becoming more and more critical, due to the current impact of software on every technical accomplishment, and the fact that networks pervade any current application. The problem has two facets: the complexity of the systems under development, and the need of continuous update to keep the pace with moving requirements. We are convinced that a design process based on formal refinements, when centered on the system architecture and applied in the early phases of development, would mitigate the problems related to both complexity and change. To this end, we have been working on a refinement calculus for distributed systems that integrates in a natural way local refinements (i.e. within a single component) and coordination templates [13, 10, 11].

The essential element of the approach is the logic which is used to specify the components of the distributed systems and their interactions. Indeed, the approach can be effective only if it comes with simple, workable concepts and notations, well matched to computational paradigm which characterizes the class of systems we are interested in, namely distributed work–flow systems. This class of systems is becoming more and more attractive for applications in domains like geografically distributed manufacturing, electronic commerce, round–the–clock world–wide software development, etc. They essentally consist of loosely coupled asynchronous long running services.

In the context of this refinement–centered design approach, in [11] we introduced Oikos_adtl, an extention of Unity [2] to deal with components and events. As in Unity, in Oikos_adtl the emphasys is on high level operators and application oriented theorems, to be used in the everyday activity of formal design.

As it is fairly common in the specification of distributed systems, Oikos_adtl is a modal logic, with modalities for time and locality, which has been designed to support an essential facet of any design method: the composition of specifications. It fosters the expression of the global properties of a system in terms of the local properties and the interactions of the components.

Complementarily, the purpose of this work is to establish some foundational ground: we provide a characterization of the class of frames which have a structure that reflects the essential properties of the causal relation in distributed systems communicating asyncronously via message passing. In this paper we will refer to this class of systems as *asynchronous systems*. Before delving into the technicalities, in the rest of this introduction we discuss informally:

- the structure of the frames;
- its impact on the logic *adtl* presented here, especially with respect to the kind of properties of the computations that can be meaningfully expressed.

Asynchronous systems have been characterized in [1] by the following properties: there are no bounds on the relative speeds of processes as well as on message delays. Therefore, they rule out synchronized local clocks, and reasoning on global time: message passing is the only possible mechanism for communication. This restricts the ways in which events in the computation can be ordered. In an asynchronous system, two events are constrained to occur in a certain order only if the occurrence of the first may affect the outcome of the second, that is only if there is a flow of information from the first to the second: if the events occur in different processes, they must be linked by the exchange of a message. This establishes an asymmetry between the processes involved in the communication, and is the obvious root of the differencies with distributed systems based on synchronous communication (called in the following *synchronous systems*).

These differences between syncronous and asynchronous systems are discussed in details in Section 7, devoted to related work. We anticipate here the main point of the discussion, making use of diagrams, reminiscent of Lamport's space–time diagrams [6], to show the evolution of the states of the components, as the result of internal transitions or of communications.

The frames describing the computations of synchronous systems have the shape shown in Fig. 1. In these structures, and consequently in any logic defined over them [9, 14, 8, 12, 3], it is not possible to express the asymmetric nature of causality which is characteristic of asynchronous systems. Looking at Fig. 1, we can both assert that w_1 causes w_2'' and that w_2 causes w_1'', because of a single communication (the double arrow linking w_1' and w_2'). The same kind of causal dependency needs at least two communications, in an asynchrous system, as shown in Fig. 2, where w_1 causes w_2' and w_2 causes w_1'.

Also, due to the synchronization induced by the communication, in Fig. 1 w_1 can be seen as the cause of a global state made of w_1' and w_2'. On the contrary, the diagram in Fig. 2 shows that w_2' is in causal relation with w_1' in component 1 and with w_3 in 3 as well, without entailing a causal relation of w_2' with a sort

Fig. 1. A synchronous frame.

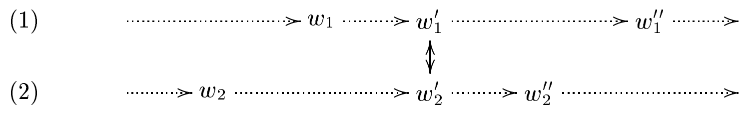
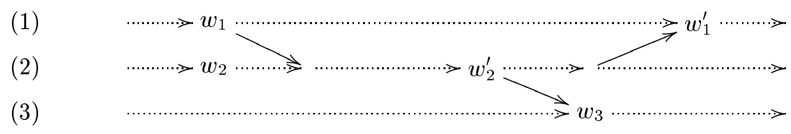


Fig. 2. An asynchronous frame.



of global state made of w_1' and w_3. Indeed, in asynchronous systems, no set of local states can be seen as a global one.

Since the global properties that may hold in a system are constrained by the underlying causal structure, any attempt to express properties that entail the syncronization of remote components must end up in a contradiction: no logical formula can be true, which requires some properties of states in different components to hold at the same time instant. This has an impact on the way specifications are dealt with, in a logic for asynchronous systems.

Normally, in temporal logic, specifications are sets of formulae $\{F_1, \ldots, F_n\}$ and are interpreted as the conjunction $F_1 \wedge \ldots \wedge F_n$, in a suitably chosen state. This traditional interpretation cannot be taken in a logic for asynchronous systems. Indeed, consider the perfectly legitimate specification which requires that p and q hold initially in components 1 and 2, respectively. The conjunction of the formulae expressing the two local properties would be false, since no state can belong to both components.

However, the formulae in a specification have to be related, to reason on the global properties of a computation, and avoid illegitimate specifications. Consider for instance a specification made of the two local properties:

- component 1 asserts that p will eventually hold in component 2, and
- component 2 assumes $\neg p$ as an invariant.

We want to be able to derive that this specification is not satisfiable. In fact, the first formula is satisfied by a computation containing a path from a state in component 1 to a state in component 2 where p holds, while the second formula states that p is always false in the states of component 1.

The solution we present here for *adtl* is to have a two–tiered logic:

- *formulae* describe local properties, and are interpreted over states of space–time diagrams, in such a way that the conjunction ($\wedge$) of properties of states in different components is always false;
- *specifications* use a weaker form of conjunction (&) to define global properties combining formulae which describe local properties: a specification is interpreted over the set of initial states of a space–time diagram.

In sections 6 and 7 we discuss alternative solutions we have considered for the definition of *adtl*, and the differences with the logic introduced in [7], which is the only logic for asynchronous systems we are aware of. The organization of the rest paper is the following. Sections 2 and 3 present the syntax and the semantics of *adtl*, respectively. Sections 4 and 5 present the main contribution of the paper, that is a complete and decidable axiom system for *adtl*.

2 The logic *adtl*

The syntax of *adtl* formulae is given over a denumerable set $Prop$ of propositional symbols, and a finite, fixed set $I = \{1, \ldots, n\}$ of component names. We let p, q, ... range over $Prop$, i, j, ... range over I.

Definition 1. *An* ***adtl*** **formula** F *is defined by:*

$$F ::= \; p \;\big|\; \neg F \;\big|\; F \wedge F \;\big|\; [\mathrm{X}]F \;\big|\; \Box F \;\big|\; [i]F$$

Modality $[\mathrm{X}]$ means "in all states reachable in one step", and $\Box$ means "in all reachable states", as usual. The new modality $[i]$ is read "if in i then". Informally, $[i]F$ holds in a state w if either F holds in w, or if w is not a state of component i. Formal semantics is provided with Def. 7. Some dual modalities can be introduced:

$$\Diamond F = \neg \Box \neg F \qquad \langle \mathrm{X} \rangle F = \neg [\mathrm{X}] \neg F \qquad \langle i \rangle F = \neg [i] \neg F$$

They read "in some reachable state", "in some state reachable in one step", and "in i and", respectively.

Example 1.

$\Box[\langle 1 \rangle p \;\rightarrow\; \Diamond\langle 2 \rangle q]$**:** this is a typical meaningful formula. Its intended meaning is that whenever p holds in 1, q will hold in 2 (and there will be a communication in between, from 1 to 2).

$\Diamond[\langle 1 \rangle p \;\wedge\; \langle 2 \rangle q]$**:** conversely, this is a typical unsatisfiable formula. Formulae are interpreted in local states, and no state can belong to more than a component.

$\Box[i]F$ says that F is an invariant of component i.

Definition 2. *An* ***adtl*** **specification** *S is defined by:*

$$S ::= \langle i \rangle F \;\big|\; S \,\&\, S$$

Example 2.

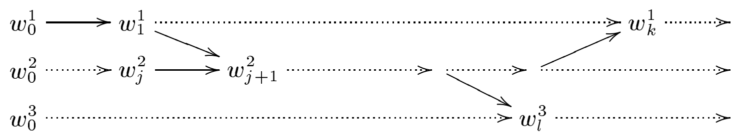
$\langle 1 \rangle p \,\&\, \langle 2 \rangle q$**:** is a meaningful specification, it requires p to hold in 1, and q in 2, initially.

$\langle 1 \rangle \Diamond \langle 2 \rangle p \,\&\, \langle 2 \rangle \Box [2] \neg p$: conversely, this is a typical unsatisfiable specification. Indeed, the assertions made by component 1 on the future of component 2 are not consistent with the assertions made by component 2 on its own future behaviour.

3 Semantics

In our setting, the interpretation domain for formulae and specifications (built over sets *Prop* and $I = \{1, \ldots, n\}$) is a structure shaping like the one in Fig. 3, composed of n denumerable chains , each representing the evolution of a single component, and an n-partite oriented graph, representing the asynchronous communications among the components. Since we are interested in distributed systems, we assume n to be strictly greater than 1: the case $n = 1$ only needs a little accommodation of a couple of definitions.

Fig. 3. A frame: plain arrows denote single steps, while dotted arrows denote paths composed of a number of steps.



Next, we define the frames for asynchronous systems, and list the properties satisfied by the accessibility relations. In Sect. 4, we will provide the axioms that characterize these properties. We first need to introduce some notation.

Definition 3. *In the paper:*

W	*is*	*a set of worlds;*
$w,\ w',\ \ldots$	*are*	*generic worlds in* W*;*
$w_0^1,\ \ldots,\ w_0^n$	*are*	*the initial worlds;*
$R_X \subseteq W \times W$	*is*	*the* "next state" *accessibility relation;*
$R_i \subseteq W \times W$	*is*	*the* "in i" *accessibility relation (for* $i = 1, \ldots, n$*);*
$w \in i$	*is*	*a shorthand for* $(w, w) \in R_i$*;*
R_X^*	*is*	*the reflexive and transitive closure of* R_X*;*
R_X^+	*is*	*the transitive closure of* R_X*;*
R_{X_i}	*is*	*the set* $\{(w,\ w') \mid (w,\ w') \in R_X$ and $w,\ w' \in i\}$.

Definition 4. *A* **frame** $\mathcal{F}$ *is a tuple* $\langle W,\ R_X,\ R_1,\ \ldots,\ R_n,\ w_0^1,\ \ldots,\ w_0^n \rangle$ *satisfying:*

$$(w, w') \in R_i \ \rightarrow\ w = w' \tag{1}$$

$$\forall\, w.\, (w, w) \in \textstyle\bigcup_{i \in I} R_i \tag{2}$$

$$R_i \cap R_j = \emptyset \ \ \textit{for } i \neq j \tag{3}$$

$$\forall\, i.\, w_0^i \in i \tag{4}$$

$$\forall\, (w, w') \in R_X,\ (w, w'') \in R_X.\, w', w'' \in i \ \Rightarrow\ w' = w'' \tag{5}$$

$$\forall\, w \in i.\, \exists\, w' :\ (w, w') \in R_{X_i} \tag{6}$$

$$(w_o^j, w) \in R_X^*,\ w \in i \ \Rightarrow\ (w_o^i, w) \in R_X^* \tag{7}$$

$$(w, w'),\ (w, w'') \in R_X^*,\ w',\ w'' \in i \ \Rightarrow\ (w', w'') \in R_X^* \vee (w'', w') \in R_X^* \tag{8}$$

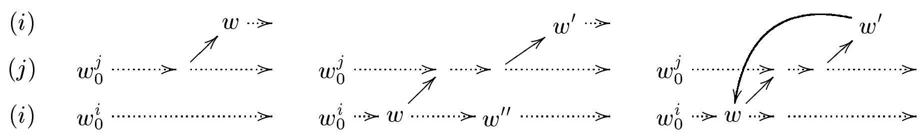
$$(w, w') \in R_X^+,\ w,\ w' \in i \ \Rightarrow\ \exists w'' \in i.\, (w, w'') \in R_X^* \wedge (w'', w') \in R_{X_i} \tag{9}$$

Worlds in W represent the states of the components, and the "in i" relations have enough structure to partition W. Indeed, property (1) says that R_i is a partial identity: the only accessible state from w according to R_i, if any, is w itself. Properties (3) and (2) entails, respectively that the R_i's are disjoint (each world in W is a state of at most one component), and that $\bigcup_{i \in I} R_i$ is reflexive (each world in W is a state of some component). Condition (4) asserts that each component has a (unique) initial state.

The accessibility relation "next state" is constrained to be linear in each component. Besides, access to other components is also linear: there is a unique state which is the recipient of a communication from a component to another one. That is, for any two pairs (w, w') and (w, w'') in R_X, either $w' = w''$ or w' and w'' belong to distinguished components. Formally, (5) expresses the required property (distributed linearity) of "next state".

It is useful to distinguish the local next states, i.e. those due to local transitions, from the remote ones, i.e. those due to the passing of a message from one component to another. For each i in I, we consider R_{X_i} as the *next i–local state accessibility* relation. R_{X_i} is exemplified in Fig. 3 by the pair of states $(w_j^i,\ w_{j+1}^i)$. Property (6) requires that R_{X_i} is total, meaning that each world has a successor in the same component.

Fig. 4. Wrong frames



The last three conditions restrain communications from j to i, from ending up in a world which is not in the linear evolution of component i. The first frame in Fig. 4 falsifies condition (7). However, this condition is not sufficient: indeed it is satisfied by the second wrong frame, and condition (8) is needed to prevent this situation. Finally, condition (9) is needed not to allow frames to shape as the third one of Fig. 4, which, on the contrary, satisfies both condition (7) and (8).

We now define the models for *adtl* formulae. As usual, a model is a frame enriched with a valuation function which defines the truth of propositional formulae on the worlds of the frame [5].

Definition 5. *A* **model** *on a frame* $\mathcal{F} = \langle W, R_x, R_1, \ldots, R_n, w_0^1, \ldots, w_0^n \rangle$ *is a tuple* $\mathcal{M} = \langle \mathcal{F}, V \rangle$ *where* V *is a mapping (valuation):* $V : W \rightarrow 2^{Prop}$

Definition 6. *A* **formula** F *is* **satisfied** *by a model* $\mathcal{M}$ *if and only if* F *is true in one of its initial worlds. A formula* F *is* **satisfiable** *if and only if it is satisfied by a model; it is* **valid** *if and only if it is true in all the initial states of all the models.*

We also say that $\mathcal{M}$ **satisfies** *a* **specification** $F_1 \,\&\, \ldots \,\&\, F_m$ *if and only for all* $i = 1, \ldots, m$, F_i *is satisfied by* $\mathcal{M}$*. A specification* S *is* **satisfiable** *if and only if it is satisfied by a model; it is* **valid** *if and only if it is satisfied by all the models.*

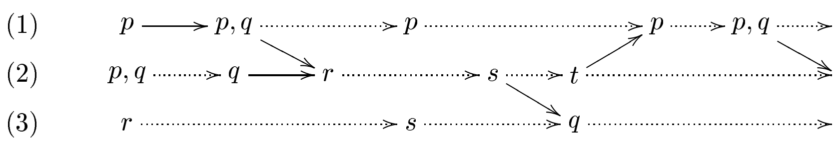
Note that if F is a valid formula, then $\langle i \rangle F$ is a valid specification, while no formula $\langle i \rangle F$ can be valid (it is false on the initial states of all components j, with $j \neq i$). This is true since $n > 1$.

The truth of a formula in a world of $\mathcal{M}$ is defined as follows.

Definition 7. *Given a model* $\mathcal{M}$*, and a world* w*:*

$$
\begin{array}{lll}
\langle \mathcal{M}, w \rangle \models & p & \text{iff } p \in V(w) \\
\langle \mathcal{M}, w \rangle \models & \neg F & \text{iff } \langle \mathcal{M}, w \rangle \not\models F \\
\langle \mathcal{M}, w \rangle \models & F \wedge F' & \text{iff } \langle \mathcal{M}, w \rangle \models F \text{ and } \langle \mathcal{M}, w \rangle \models F' \\
\langle \mathcal{M}, w \rangle \models & [\mathrm{x}]F & \text{iff } \forall\, w' : w R_x w' . \langle \mathcal{M}, w' \rangle \models F \\
\langle \mathcal{M}, w \rangle \models & \Box F & \text{iff } \forall\, w' : w R_x^* w' . \langle \mathcal{M}, w' \rangle \models F \\
\langle \mathcal{M}, w \rangle \models & [i]F & \text{iff } \forall\, w' : w R_i w' . \langle \mathcal{M}, w' \rangle \models F
\end{array}
$$

Example 3. As an example, the structure (each world is labelled with the propositions holding in it):



when suitably extended, is a model of the formulae:

$\langle 1\rangle\Box\,[1]\,p$: p holds in all states of component 1 ($\Box\langle 1\rangle p$ is false instead);

$\langle 1\rangle\,p$: p holds in the initial state of component 1;

$\Box\,[\langle 2\rangle\,q \;\rightarrow\; (\Diamond\,\langle 1\rangle\,p \;\wedge\; \Diamond\,\langle 3\rangle\,q)]$: Any state in 2 satisfying q is eventually followed by a state in 1 satisfying p, and by a state in 3 satisfying q;

$\Box\,(\langle 1\rangle\,q \;\rightarrow\; \langle \mathrm{x}\rangle\,\langle 2\rangle\,\mathit{true})$: A state of component 1 satisfying q is immediately followed by a state in 2, that is, every time q holds in 1, there is a communication from 1 to 2;

4 Axiom System

In this section we present the axiom system for *adtl*. F, F', F'' are *adtl* formulae built on the set $I = \{1, \ldots, n\}$, and i, $j \in I$. S, S' are specifications.

We say $F \;\vdash_f\; F'$ if formula F' can be derived from F according to the axiom system of Sect. 4.1; we say $S \;\vdash_s\; S'$ if specification S' can be derived from S according to the axiom system of Sect. 4.2. When F (S) equal *true*, i.e., no assumption has to be made to derive F' (S'), we omit it and write $\vdash_f\; F'$ ($\vdash_s\; S'$).

4.1 Axioms for the Formulae

Axioms of the propositional calculus:

P1 $\quad F \rightarrow (F' \rightarrow F)$

P2 $\quad (F \rightarrow (F' \rightarrow F'')) \rightarrow ((F \rightarrow F') \rightarrow F \rightarrow F''))$

P3 $\quad ((\neg\, F \rightarrow \neg\, F') \rightarrow ((\neg\, F \rightarrow F') \rightarrow F))$

Axiom **K**, for all modalities:

K$\Box$ $\quad \Box(F \rightarrow F') \;\rightarrow\; (\Box F \rightarrow \Box F')$

K[x] $\quad [\mathrm{x}](F \rightarrow F') \;\rightarrow\; ([\mathrm{x}]F \rightarrow [\mathrm{x}]F')$

K[i] $\quad [i](F \rightarrow F') \;\rightarrow\; ([i]F \rightarrow [i]F')$

The following axioms are those normally used to define a temporal logic. Axioms **t1** and **t2** model the stepping of [x] through $\Box$. Axioms **4** and **T** define the transitive and reflexive properties of time, respectively, and axiom **D** states totality: a next state always exists.

t1	$\Box(F \to [\mathrm{x}]F) \to (F \to \Box F)$
t2	$\Box F \to [\mathrm{x}]\Box F$
4	$\Box F \ \to \ \Box\Box F$
T	$\Box F \ \to \ F$
D	$[\mathrm{x}]F \to \langle \mathrm{x} \rangle F$

The last set of axioms deals with the new modality $[i]$.

a[i]	$F \ \to \ [i]F$
b[i]	$[1]F \ \wedge \ \ldots \wedge [n]F \ \to \ F$
c[i]	$[i][j]F \qquad \text{with } i \neq j.$
d[i]	$\langle \mathrm{x} \rangle \langle i \rangle F \ \to \ [\mathrm{x}][i]F$
e[i]	$\langle i \rangle [\mathrm{x}]F \ \to \ \langle \mathrm{x} \rangle \langle i \rangle F$
f[i]	$(\Diamond\langle i \rangle F \ \wedge \ \Diamond\langle i \rangle F') \ \to \ \Diamond\langle i \rangle[(F \wedge \Diamond F') \vee (F' \wedge \Diamond F)]$
g[i]	$\langle i \rangle \Diamond \langle \mathrm{x} \rangle \langle i \rangle F \ \to \ \Diamond \langle i \rangle \langle \mathrm{x} \rangle F$

Finally, we have the inference rules:

$$\textbf{MP} \ \frac{F \quad F \to F'}{F'} \qquad \textbf{Nec} \ \frac{F}{\Box F}$$

Example 4. The following formulae can be derived:

1) $\langle i \rangle \langle i \rangle F \ \leftrightarrow \ \langle i \rangle F$
2) $[i]F \ \leftrightarrow \ [i][i]F$
3) $F \ \to \ [i]\langle j \rangle F$
4) $\langle i \rangle F \wedge \langle j \rangle F' \ \leftrightarrow \ \langle i \rangle \langle j \rangle (F \wedge F')$
5) $[i](F \vee F') \ \leftrightarrow \ [i]F \vee [i]F'$

4.2 Axioms for the Specifications

Init $\langle 1\rangle true\,\&\,\ldots\,\&\langle n\rangle true$

$\wedge$I $$\frac{\langle i\rangle F\;\&\;\langle i\rangle F'}{\langle i\rangle(F\wedge F')}$$

DC $$\frac{\langle j\rangle\Diamond\langle i\rangle F}{\langle i\rangle\Diamond F}$$

&I , &E $$\frac{S\quad S'}{S\;\&\;S'}\qquad\frac{S\;\&\;S'}{S}\qquad\frac{S\;\&\;S'}{S'}$$

LMP $$\frac{\langle i\rangle F\quad F\vdash_f F'}{\langle i\rangle F'}$$

Axiom **Init** says that an initial state exists, for each component $i\in I$; $\wedge$**I** says that each component has at most one initial state; **DC** (distributed coherence) says that the i-future assertions made by any component j are consistent with the future assertions made by the initial state of component i; &**I** and &**E** introduce or remove conjunction &, respectively; finally, **LMP** is a local modus ponens.

Example 5. The following rule is derived: **DC'** $\dfrac{\langle j\rangle\Diamond\langle i\rangle F}{\langle i\rangle\Diamond\langle i\rangle F}$

in fact:

$$\dfrac{\dfrac{\langle j\rangle\Diamond\langle i\rangle F\qquad\qquad\vdash_f\;\langle i\rangle F\;\leftrightarrow\;\langle i\rangle\langle i\rangle F\text{ (see 1 in Ex. 4)}}{\langle j\rangle\Diamond\langle i\rangle\langle i\rangle F}\,\text{LMP}}{\langle i\rangle\Diamond\langle i\rangle F}\,\text{DC}$$

5 Soundness, completeness and decidability

We say that an *adtl* formula F is consistent if and only if $F\not\vdash_f false$, and that an *adtl* specification S is consistent if and only if $S\not\vdash_s false$.

We consider, in this section, consistency with respect to axiom systems which are subsystems of those introduced in Sects. 4.1 and 4.2. If needed, we explicitly name the axiom system we refer to, and say “a formula consistent in ...”.

Example 6. The specification $\langle 1\rangle\Diamond\langle 2\rangle p \,\&\, \langle 2\rangle\Box\neg p$ is not consistent. In fact:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\langle 1\rangle\Diamond\langle 2\rangle p \,\&\, \langle 2\rangle\Box\neg p}{\langle 1\rangle\Diamond\langle 2\rangle p}\,\&\mathrm{E}}{\langle 2\rangle\Diamond p}\,\mathrm{DC} \qquad \dfrac{\langle 1\rangle\Diamond\langle 2\rangle p \,\&\, \langle 2\rangle\Box\neg p}{\langle 2\rangle\Box\neg p}\,\&\mathrm{E}}{\langle 2\rangle[\Diamond p \wedge \Box\neg p]}\,\wedge\mathrm{I} \qquad \Diamond p\wedge\Box\neg p \vdash_f \mathit{false}}{\mathit{false}}\,\mathrm{LMP}
$$

Note also that, while $\langle 1\rangle p \wedge \langle 2\rangle q$ is not a consistent formula, $\langle 1\rangle p \,\&\, \langle 2\rangle q$ is a consistent specification.

Proposition 1. *The* adtl *axiom systems are sound, i.e., if an* adtl *formula F can be derived ($\vdash_f F$), then F is valid; if and* adtl *specification S can be derived ($\vdash_s S$), then S is valid.*

We omit the proof, which follows standard reasoning: we prove formulae in the axiom system and the **Init** specification to be valid, and prove, for all rules in the axiom system that if the premises are valid, then so are the consequences. Instead, we dwell on completeness and decidability, which are proved in three steps:

1. In Prop. 2 we consider the axiom system Σ = **P1**, **P2**, **P3**, **K**$\Box$, **K**[x], **K**$[i]$, **t1**, **t2**, **4**, **T**, **D**, and show that for any formula F consistent in Σ, a finite model exists. The model built is not guaranteed to satisfy the frame properties of Def. 4.
2. In Prop. 3 we show that if we consider a consistent *adtl* formula F, we can exploit the characteristic formula technique to build a finite model for F satisfying properties (1), (2), (3), (5), (6), (8), and (9) introduced in Def. 4.
3. In Prop. 4 we consider specifications and show that any consistent specification S has a finite model satisfying also properties (4) and (7).

Proposition 2. *Any formula F consistent in Σ has a finite model.*

Proof. Axiom system Σ is exactly the axiom system of propositional dynamic logic (PDL) [4], where the set of actions is $I \cup \{\mathrm{x}\}$. We can thus apply the completeness and decidability results obtained for PDL. $\Box$

The finite models in PDL are built by filtration. We recall how this is done.

Definition 8. *Given a consistent formula F, the (Fischer–Ladner) closure of F is the set $CL(F) = CL'(F) \cup \{\neg G \,|\, G \in CL'(F)\}$ where $CL'(F)$ is the smallest set of formulae containing F, and satisfying:*

$if\ \neg G \in CL'(F)\ then\ G \in CL'(F)$
$if\ G \wedge H \in CL'(F)\ then\ \{G, H\} \subseteq CL'(F)$
$if\ [i]G \in CL'(F)\ then\ \{G\} \subseteq CL'(F)$
$if\ \Box G \in CL'(F)\ then\ \{G, [i]\Box G\}_{i=0,\ldots,m} \subseteq CL'(F)$

Definition 9. *Given a consistent formula F, we build the model $((AT, \Rightarrow, L), V)$, where:*

- *AT is the set of the maximal consistent subsets of $CL(F)$; w, w' range over AT. We denote with $\hat{w}$ the conjunction of all formulae in w. Note that $|AT|$ is linear on the size of the formula F, and so it is finite.*
- *L is a set of labels ranging over the set of actions $I \cup \{\mathrm{X}\}$, and the (labelled) accessibility relation $\stackrel{a}{\Rightarrow}$ is defined as follows:*

$$w \stackrel{a}{\Rightarrow} w' \quad \textit{iff} \quad \hat{w} \wedge \langle a \rangle \hat{w}' \ \textit{is consistent}$$

Proposition 3. *The axiom system of* adtl *formulae provides a complete axiomatization of formulae valid over the class of frames satisfying conditions: (1), (2), (3), (5), (6), (8), and (9) of Def. 4.*

Proof. Given a consistent formula $\bar{F}$, we build a model for it according to Def. 9 (here consistency is with respect to the axiom system for *adtl* formulae). We show that the model built satisfies the stated frame condition.

Let w be any maximal consistent set: and F a formula in w:

(1: R_i is a partial identity.) $F \in w$ implies ($\mathbf{a[i]}$) $[i]F \in w$. Thus, let $w \stackrel{i}{\Rightarrow} w'$, we have $F \in w'$ and consequently $w = w'$, for the maximality of w.

(2: $\bigcup_{i \in I} R_i$ is reflexive.) Assume that for no i $w \stackrel{i}{\Rightarrow} w$, i.e. that $\hat{w} \wedge \langle i \rangle \hat{w}$ is inconsistent for all i. This means that $\hat{w} \rightarrow \neg\langle 1 \rangle \hat{w} \wedge \ldots \wedge \neg\langle n \rangle \hat{w}$, i.e. $\hat{w} \rightarrow [1]\neg\hat{w} \wedge \ldots \wedge [n]\neg\hat{w}$, and consequently ($\mathbf{b[i]}$) $\hat{w} \rightarrow \neg\hat{w}$ which is a contradiction.

(3: the R_i's are disjoint.) Axiom $\mathbf{c[i]}$ says that $[i][j]F \in w$ for all F. In particular, $[i][j]false \in w$. Assume $w \stackrel{i}{\Rightarrow} w'$ and $w \stackrel{j}{\Rightarrow} w'$, i.e. (partial identity of R_i and R_j) $w \stackrel{i}{\Rightarrow} w$ and $w \stackrel{j}{\Rightarrow} w$. Then, $[i][j]false \in w$ implies $[j]false \in w$, and this implies $false \in w$ which is a contradiction.

(5: distributed linearity of "next state".) Let $w \stackrel{\mathrm{X}}{\Rightarrow} w'$, $w' \stackrel{i}{\Rightarrow} w'$, $w \stackrel{\mathrm{X}}{\Rightarrow} w''$, and $w'' \stackrel{i}{\Rightarrow} w''$. If we assume $w' \neq w''$, we can find a formula F, with $F \in w'$ and $F \notin w''$. $F \in w'$ implies that $\hat{w} \wedge \langle \mathrm{X} \rangle \langle i \rangle F$ is consistent. Then, for the maximality of w, $\langle \mathrm{X} \rangle \langle i \rangle F \in w$. Applying $\mathbf{d[i]}$, $[\mathrm{X}][i]F \in w$ which implies $[i]F \in w''$ and hence $F \in w''$, which a is contradiction.

(6: totality of R_{x_i}.) Let $w \stackrel{i}{\Rightarrow} w$. We know that if $F \in w$ then $\langle i \rangle F \in w$. Since $[\mathrm{X}]true$ is a valid formula, we have $[\mathrm{X}]true \in w$ and thus $\langle i \rangle [\mathrm{X}]true \in w$. Then, applying $\mathbf{e[i]}$, $\langle \mathrm{X} \rangle \langle i \rangle true \in w$. This implies that $\exists w'$ s.t. $w \stackrel{\mathrm{X}}{\Rightarrow} w'$, with $\langle i \rangle true \in w'$, which, in turn implies that $w' \stackrel{i}{\Rightarrow} w'$.

(8: confluence of R_x.) Assume $(w,w'),\,(w,w'') \in R_{\mathrm{X}}^*$ with $w',\,w'' \in i$. This implies that $\Diamond\langle i\rangle\hat{w'} \wedge \Diamond\langle i\rangle\hat{w''} \in w$. For the consistency of w, applying $\mathbf{f[i]}$: $\Diamond\langle i\rangle(\hat{w'} \wedge \Diamond\hat{w''} \,\vee\, \hat{w''} \wedge \Diamond\hat{w'}) \in w$. For the maximality of w' and w'' we thus have that $\Diamond\hat{w''} \in w'$ or $\Diamond\hat{w'} \in w''$, i.e. $(w',w'') \in R_{\mathrm{X}}^* \,\vee\, (w'',w') \in R_{\mathrm{X}}^*$.

(9.) Assume $(w,w') \in R_{\mathrm{X}}^+$ with $w,\,w' \in i$. This implies $\langle i\rangle\Diamond\langle \mathrm{X}\rangle\langle i\rangle\hat{w'} \in w$. For the consistency of w we need to have (see axiom $\mathbf{g[i]}$) $\Diamond\langle i\rangle\langle \mathrm{X}\rangle\hat{w'} \in w$. This implies the existence of w'' with $\Diamond\hat{w''} \in w$ and $\langle i\rangle\langle \mathrm{X}\rangle\hat{w'} \in w''$. So, we can conclude that $\exists w'' \in i.\,(w,w'') \in R_{\mathrm{X}}^* \,\wedge\, (w'',w') \in R_{\mathrm{X}_i}$.

□

Proposition 4. *Each consistent specification has a finite model.*

Proof. Given a consistent specification S, we bring it to the canonical form $S' = \langle 1\rangle F_1 \,\&\ldots\&\langle n\rangle F_n$ by repeatedly applying $\wedge$I to $S\&\langle 1\rangle \mathit{true}\&\ldots\&\langle n\rangle \mathit{true}$. S' is a consistent specification, and a model for S' is also a model for S.

The consistency of S' as a specification implies (rule **LMP**) the consistency of the formulae $\langle i\rangle F_i$ for all $i = i,\ldots,n$ (here consistency is for the axiom system of *adtl* formulae). We can apply Prop. 3 and build a model for each $\langle i\rangle F_i$.

We prove that we can merge the models for the $\langle i\rangle F_i$ to obtain a model for S.

Without loss of generality, we restrict to the case $n = 2$, i.e., to the specification $\langle 1\rangle F_1 \,\&\, \langle 2\rangle F_2$. Let w_a^2 be the first world of the model $\mathcal{M}_1$ of $\langle 1\rangle F_1$ belonging to component 2 and accessible from w_0^1, and w_b^1 the first world of the model $\mathcal{M}_2$ of $\langle 2\rangle F_2$ belonging to component 1 and accessible from w_0^2, as shown in the following pictures (only the relevant fragment of these models is shown):

$$\begin{tikzcd}
(1) & w_0^1 \arrow[r, dotted] & \cdot \arrow[r, dotted, -|] \arrow[dr] & {} \\
(2) & & & w_a^2 \arrow[r, dotted, -|] & {}
\end{tikzcd}
\qquad
\begin{tikzcd}
(1) & & & w_b \arrow[r, dotted, -|] & {} \\
(2) & w_0^2 \arrow[r, dotted] & \cdot \arrow[ur] \arrow[r, dotted, -|] & {}
\end{tikzcd}$$

If we prove that we can find in $\mathcal{M}_1$ a world $\bar{w}_b^1$ consistent with w_b^1, so that $\mathcal{M}_1$ shapes like:

$$\begin{tikzcd}
(1) & w_0^1 \arrow[r, dotted] & \bar{w}_b^1 \arrow[r, dotted] & \cdot \arrow[r, dotted, -|] \arrow[dr] & {} \\
(2) & & & & w_a^2 \arrow[r, dotted, -|] & {}
\end{tikzcd}
\quad OR \quad
\begin{tikzcd}
(1) & w_0^1 \arrow[r, dotted] & \cdot \arrow[r, dotted] & \cdot \arrow[r, dotted] \arrow[dr] & \bar{w}_b^1 \arrow[r, dotted, -|] & {} \\
(2) & & & & w_a^2 \arrow[r, dotted, -|] & {}
\end{tikzcd}$$

and that we can find in $\mathcal{M}_2$ a world $\bar{w}_a^2$ consistent with w_a^2, so that $\mathcal{M}_2$ shapes like:

$$\begin{tikzcd}
(1) & & & & w_b^1 \arrow[r, dotted, -|] & {} \\
(2) & w_0^2 \arrow[r, dotted] & \bar{w}_a^2 \arrow[r, dotted] & \cdot \arrow[ur] \arrow[r, dotted, -|] & {}
\end{tikzcd}
\quad OR \quad
\begin{tikzcd}
(1) & & & w_b^1 \arrow[r, dotted, -|] & {} \\
(2) & w_0^2 \arrow[r, dotted] & \cdot \arrow[ur] \arrow[r, dotted] & \bar{w}_a^2 \arrow[r, dotted, -|] & {}
\end{tikzcd}$$

we can build a model shaping:

$$\begin{tikzcd}
(1) & w_0^1 \cup \Diamond w_b^1 \cup \Diamond \bar{w}_a^2 \arrow[rr, dotted] & & w_b^1 \cup \bar{w}_b^1 \arrow[r, dotted] & \cdot \arrow[r, dotted, -|] \arrow[dr] & {} \\
(2) & w_0^2 \cup \Diamond w_a^2 \cup \Diamond \bar{w}_b^1 \arrow[r, dotted] & \cdot \arrow[ur] \arrow[rrr, dotted] & & & w_a^2 \cup \bar{w}_a^2 \arrow[r, dotted, -|] & {}
\end{tikzcd}$$

or:

$$\begin{tikzcd}[column sep=small, row sep=small]
(1) & w_0^1 \cup \Diamond w_b^1 \cup \Diamond \bar{w}_a^2 \arrow[r, dotted] & {} & {} \arrow[r, dotted] & w_b^1 \cup \bar{w}_b^1 \arrow[r, dotted] & {} \arrow[dlll] & {} \arrow[r, dotted, dash] & {} \\
(2) & w_0^2 \cup \Diamond w_a^2 \cup \Diamond \bar{w}_b^1 \arrow[r, dotted] & {} & {} \arrow[r, dotted] & w_a^2 \cup \bar{w}_a^2 \arrow[r, dotted] & {} \arrow[ulll] & {} \arrow[r, dotted, dash] & {}
\end{tikzcd}$$

or as the merge of the other two possible combinations of $\mathcal{M}_1$ and $\mathcal{M}_2$ above. By construction, all these structures satisfy the frame properties, including (4) and (7), and thus are (finite) models.

Note that a model like the last one, including backwards communications, can be unfold to the equivalent[1]:

$$\begin{tikzcd}[column sep=small, row sep=small]
(1) & w_0^1 \cup \ldots \arrow[rrr, dotted] & & & {} \arrow[r, dotted] & w_b^1 \cup \bar{w}_b^1 \arrow[r, dotted] & {} \arrow[dr] \arrow[r, dotted] & {} \\
(2) & w_0^2 \cup \ldots \arrow[r, dotted] & w_a^2 \cup \bar{w}_a^2 \arrow[r, dotted] & {} & {} \arrow[ul] \arrow[rrr, dotted] & & & {} \arrow[r, dotted] & {}
\end{tikzcd}$$

We now prove that worlds $\bar{w}_b^1$, $\bar{w}_{a'}^2$ as above can be found. Assume that this is not the case. This means that there exist a pair of formulae F_a, F_b such that for all pair of models $\mathcal{M}_1$, $\mathcal{M}_2$ of $\langle 1\rangle F_1$, $\langle 2\rangle F_2$:

$$\begin{array}{l}
\langle \mathcal{M}_1, w_0^1\rangle \models \Diamond\langle 2\rangle F_a \text{ and } \langle \mathcal{M}_2, w_0^2\rangle \models \neg\Diamond\langle 2\rangle F_a \\
\text{or} \\
\langle \mathcal{M}_1, w_0^1\rangle \models \neg\Diamond\langle 1\rangle F_b \quad \text{and} \quad \langle \mathcal{M}_2, w_0^2\rangle \models \Diamond\langle 1\rangle F_b
\end{array}$$

It is easy to see that this is equivalent to say that:

$$\begin{array}{l}
\forall\mathcal{M}_1.\, \langle \mathcal{M}_1, w_0^1\rangle \models \Diamond\langle 2\rangle F_a \quad \text{and} \quad \forall\mathcal{M}_2.\, \langle \mathcal{M}_2, w_0^2\rangle \models \neg\Diamond\langle 2\rangle F_a \\
\text{or} \\
\forall\mathcal{M}_1.\, \langle \mathcal{M}_1, w_0^1\rangle \models \neg\Diamond\langle 1\rangle F_b \quad \text{and} \quad \forall\mathcal{M}_2.\, \langle \mathcal{M}_2, w_0^2\rangle \models \Diamond\langle 1\rangle F_b
\end{array}$$

Let (the other case is symmetric):

$$\forall\mathcal{M}_1.\, \langle \mathcal{M}_1, w_0^1\rangle \models \Diamond\langle 2\rangle F_a \quad \text{and} \quad \forall\mathcal{M}_2.\, \langle \mathcal{M}_2, w_0^2\rangle \models \neg\Diamond\langle 2\rangle F_a$$

We can apply the completeness result for the *adtl* formulae axiom system and conclude that:

$$\langle 1\rangle F_1 \vdash_f \langle 1\rangle\Diamond\langle 2\rangle F_a \quad \text{and} \quad \langle 2\rangle F_2 \vdash_f \neg\Diamond\langle 2\rangle F_a$$

[1] Since it is not possible to express the asymmetry of a reachability relation, the logic is not able to distinguish them, as usual.

We can now conclude deriving an inconsistent specification from the (consistent) one $\langle 1\rangle F_1 \;\&\; \langle 2\rangle F_2$:

$$\dfrac{\dfrac{\langle 1\rangle F_1 \;\&\; \langle 2\rangle F_2}{\langle 1\rangle F_1 \;\&\; \langle 2\rangle(F_2 \wedge \Diamond\langle 2\rangle F_a \wedge \neg\Diamond\langle 2\rangle F_a)}\ \&\mathrm{E}, \mathrm{LMP}, \mathrm{DC}'(\text{see Ex. } 5), \wedge\mathrm{I}, \&\mathrm{I}}{\langle 1\rangle F_1 \;\&\; \mathit{false}}\ \&\mathrm{I},\ \mathrm{LMP}$$

□

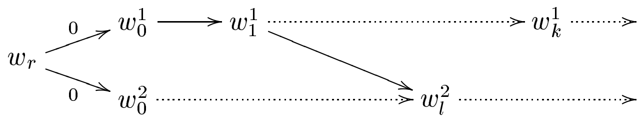
6 Discussion

Before defining *adtl* using a two–tiered syntax, we explored other solutions, we briefly discuss: why they have been discarded.

A first idea was to require a formula F to be true in a model $\mathcal{M}$ if and only if F is true in all the initial worlds of $\mathcal{M}$. This way it was possible to express with a formula $([1]\,p \;\wedge\; [2]\,q)$, i.e., without using specifications, the property of a system: p holds in the initial state of component 1, and q holds in the initial state of component 2.

However this notion of satisfiability causes some troubles. In particular, it is no longer the case that the negation of a non-satisfiable formula is valid.

A second alternative is to enrich the space–time diagrams with a kind of *root* state, w_r in the figure, accessing the initial states of the actual diagram, and adding label 0 to denote the new accessibility relation (call it R_0).



As a consequence of the extension, we can say that a model $\mathcal{M}$ satisfies a set of formulae $\{F_1, \ldots, F_n\}$ if and only $\langle 0\rangle F_1 \wedge \ldots \wedge \langle 0\rangle F_n$ is true in w_r, where $[0]$ is the modality corresponding to R_0.

For instance, the property "p holds in the initial state of component 1, and q holds in the initial state of component 2", can be expressed by the set of formulae $\{\langle 1\rangle p,\ \langle 2\rangle q\}$. In fact, the property holds in a model if and only if its root state w_r satisfies: $\langle 0\rangle\langle 1\rangle p \;\wedge\; \langle 0\rangle\langle 2\rangle q$.

Then, either we allow to use modality $[0]$ in the formulae or not. In the first case the problem is to define the meaning of a formula like $\langle 0\rangle\langle 1\rangle p$: w_r should satisfy $\langle 0\rangle\langle 0\rangle\langle 1\rangle p$ and relation R_0 be reflexive, but this would make the axiom system much more complex. In the second case, we cannot define an axiom corresponding to **DC**, which however is needed to relate i-future assertions made by any component j with the future assertions made by the initial state of component i.

7 Related Work

Various extensions of temporal logic to deal with distributed systems have been defined in the literature. We present and discuss the proposals which are most closely related to ours.

In the logic TTL [9], for each local state of the system, a *visibility* function specifies which information of remote components is accessible. The visibility function is defined on the basis of a relation among states which is symmetric in the case of states belonging to distinguished components.

A trace based extension of linear time temporal logic, called *TrPTL*, has been defined in [14] (see also [15]). The logic has been designed to be interpreted over infinite traces, i.e., labelled partial orders of actions, which respect some dependence relations associated to the alphabet of actions.

In [8], a temporal logic, StepTL, is defined and interpreted over multistep transition systems. These are a well known extension of transition systems, allowing to describe as concurrent the steps of computation that can actually be executed in parallel. A multistep transition system thus contains transitions of the form $s\,A\,s'$, where A is a set of actions, instead of a single one.

Three distinguished logics are presented in [12] to describe systems composed of sets of communicating *agents*. The logics differentiate on the amount of information each agent can have on the other agents running on the system, but share a common setting: agents communicate via common actions. The models for these logics are runs of networks of synchronizing automata. The logics $\mathbf{D}_0$ and $\mathbf{D}_1$ presented in [3] are based on a similar approach.

Anyway, in all these proposals, components communicate via some form of synchronization, and logic formulae are interpreted on models shaping:

$$\begin{tikzcd}
1: & {} \arrow[rr, dotted] & & a \arrow[r, dotted] & b \arrow[rr, dotted] \arrow[d, leftrightarrow] & & c \arrow[r, dotted] & {} \\
2: & {} \arrow[r, dotted] & d \arrow[rr, dotted] & & e \arrow[r, dotted] & f \arrow[rr, dotted] & & {}
\end{tikzcd}$$

In these models, and consequently in any logic defined over them, it is not possible to express the asymmetric nature of causality as required when modelling the behaviour of agents communicating asynchronously by message passing. Indeed, in the previous model we can both assert that a causes f and that d causes c.

Moreover, the model above satisfies a formula like $\Diamond\,(\langle 1\rangle b \wedge \langle 2\rangle e)$, implicitly talking about global states. Indeed, if we interpret a logic in these models, the only way to prevent formulae on compound states from being true in the same time instant is to restrict the formulae syntax in some (often unnatural) way.

A logic closer to *adtl* is proposed by Lodaya, Ramanujam, and Thiagarajan in [7], where a branching time temporal logic for asynchronously communicating sequential agents (ACSAs) is defined. ACSAs communicate asynchronously via message passing. The logic contains temporal modalities indexed with a local point of view of one agent and allows an agent “i” to refer to local properties of another agent "j" according to the latest message received. In other words, an agent can gain information about another agent through the reception of messages but not by sending them. On the contrary, we allow agents to make remote future assertions: therefore it is easier to express global liveness properties.

Conclusions

We have introduced *adtl*, a logic to reason on asynchronous distributed systems communicating via message passing. Models for *adtl* are based on space–time diagrams, which reflect the asymmetric nature of the causal relation in these systems. Further, models built on space–time diagrams falsify any formula predicating about global states. Finally, we have introduced an axiom system for *adtl* and proved its soundness, completeness, and decidability.

Since Oikos_adtl can be defined in terms of *adtl*, the results presented here will find a natural application in the refinement calculus we are developing for asynchronous systems.

Acknowledgements

We gratefully thank R. Ramanujam for discussions and comments.

This work was partly supported by the ESPRIT W.G. 24514 COORDINA, and the MURST co–funded project SALADIN. The first author's postdoctoral fellowship was offered by the Cassa di Risparmio di Firenze.

References

1. Ozalp Babaoglu and Keith Marzullo. Consistent global states of distributed systems: Fundamental concepts and mechanisms. In S. Mullender, editor, *Distributed Systems*, pages 55–96. Addison-Wesley, 1993.
2. K.M. 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 J. Chomicki and G. Saake, editors, *Logic for Databases and Information Syustems*, pages 167–198. Kluver Academic Publishers, 1998.
4. D. Harel. Dynamic Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, pages 497–604. Reidel Publishing Company, 1984.
5. S.A. Kripke. A completeness theorem in modal logic. *The Journal of Symbolic logic*, 24:1–14, 1959.
6. Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, July 1978.
7. Lodaya, Ramanujam, and Thiagarajan. Temporal logics for communicating sequential agents: I. *IJFCS: International Journal of Foundations of Computer Science*, 3(2):117–159, 1992.
8. K. Lodaya, R. Parikh, R. Ramanujam, and P.S. Thiagarajan. A Logical Study of Distributed Transition Systems. *Information and Computation*, 119(1):91–118, 1995.

9. A. Masini and A. Maggiolo-Schettini. TTL: A formalism to describe local and global properties of distributed systems. *Informatique théorique et Applications/Theoretical Informatics and Applications*, 26(2), 1992.
10. C. Montangero and L. Semini. Refining by Architectural Styles or Architecting by Refinements. In L. Vidal, A. Finkelstein, G. Spanoudakis, and A.L. Wolf, editors, *2nd International Software Architecture Workshop, Proceedings of the SIGSOFT '96 Workshops, Part 1*, pages 76–79, San Francisco, CA, Oct 1996. ACM Press.
11. C. Montangero and L. Semini. Composing specifications for coordination. In P. Ciancarini and A. Wolf, editors, *Proc. 3nd Int. Conf. on Coordination Models and Languages*, volume 1594 of *Lecture Notes in Computer Science*, pages 118–133, Amsterdam, April 1999. Springer-Verlag.
12. 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.
13. L. Semini and C. Montangero. A Refinement Calculus for Tuple Spaces. *Science of Computer Programming*, 34:79–140, 1999.
14. P. S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proceedings, Ninth Annual IEEE Symposium on Logic in Computer Science*, pages 438–447, Paris, Jul 1994. IEEE Computer Society Press.
15. P. S. Thiagarajan and J. G. Henriksen. Distributed versions of linear time temporal logic: A trace perspective. *Lecture Notes in Computer Science*, 1491, 1998.