Distribution of Modal Transition Systems

German E. Sibay[1], Sebastián Uchitel[1,2], Victor Braberman[2], and Jeff Kramer[1]

[1] Imperial College London, London, U.K.
[2] Universidad de Buenos Aires, FCEyN, Buenos Aires, Argentina

Abstract. In order to capture all permissible implementations, partial models of component based systems are given as at the system level. However, iterative refinement by engineers is often more convenient at the component level. In this paper, we address the problem of decomposing partial behaviour models from a single monolithic model to a component-wise model. Specifically, given a Modal Transition System (MTS) M and component interfaces (the set of actions each component can control/monitor), can MTSs $M_1, \ldots, M_n$ matching the component interfaces be produced such that independent refinement of each M_i will lead to a component Labelled Transition Systems (LTS) I_i such that composing the I_is result in a system LTS that is a refinement of M? We show that a sound and complete distribution can be built when the MTS to be distributed is deterministic, transition modalities are consistent and the LTS determined by its possible transitions is distributable.

Keywords: Modal Transition Systems, Distribution.

1 Introduction

Partial behaviour models such as Modal Transition Systems (MTS) [LT88] extend classical behaviour models by introducing transitions of two types: required or must transitions and possible or may transitions. Such extension supports interpreting them as sets of classical behaviour models. Thus, a partial behaviour model can be understood as describing the set of implementations which provide the behaviour described by the required transitions and in which any other additional implementation behaviour is possible in the partial behaviour model.

Partial behaviour model refinement can be defined as an implementation subset relation, thus naturally capturing the model elaboration process in which, as more information becomes available (e.g. may transitions are removed, required transitions are added), the set of acceptable implementations is reduced. Such notion is consistent with modern incremental development processes where fully described problem and solution domains are unavailable, undesirable or uneconomical.

The family of MTS formalisms has been shown to be useful as a modeling and analysis framework for component-based systems. Significant amount of work has been devoted to develop theory and algorithmic support in the

D. Giannakopoulou and D. Méry (Eds.): FM 2012, LNCS 7436, pp. 403–417, 2012.

context of MTS, MTS-variants, and software engineering applications. Developments include techniques for synthesising partial behaviour models from various specification languages (e.g. [FBD+11, SUB08, KBEM09]), algorithms for manipulating such partial behaviour models (e.g. [KBEM09, BKLS09b]), refinement checks [BKLS09a], composition operators including parallel composition and conjunction (e.g. [FBD+11]), model checking results(e.g. [GP11]), and tools (e.g. [Sto05, DFFU07]).

Up to now, an area that had been neglected is that of model decomposition or distribution. Distributed implementability and synthesis has been studied for LTS [Mor98, CMT99, Ste06, HS05] for different equivalences notion like isomorphism, language equivalence and bisimulation. On the other hand, work on MTSs has mostly assumed a monolithic system model which is iteratively refined until an implementation in the form of a LTS is reached.

Problems related to MTS distribution were studied by some authors [KBEM09, QG08, BKLS09b] and we compare their work to ours in Section 4. However the general problem of how to move from an MTS that plays the role of a monolithic partial behaviour model to component-wise partial behaviour model (set of MTSs) has not been studied. We study the distribution problem abstractly from the specification languages used to describe the MTS to be distributed. Those languages may allow description of behaviour that is not distributable [UKM04] and a distribution is not trivial. Furthermore we study the problem of finding all possible distributed implementations. Appropriate solutions to this problem would enable engineers to move from iterative refinement of a monolithic model to component-wise iterative refinement.

More specifically, we are interested in the following problem: given an MTS M and component interfaces (the set of actions each component can control/monitor), can MTSs $M_1, \ldots, M_n$ matching the component interfaces be produced such that independent refinement of each M_i will lead to a component LTS I_i such that composing the I_is result in a system LTS that is a refinement of M? We show that a sound and complete distribution can be built when the MTS to be distributed is deterministic, transition modalities are consistent and the LTS determined by its possible transitions is distributable.

We present various results that answer the above questions to some extent. The main result of the paper is an algorithm that, under well-defined conditions, produces component MTSs of a monolithic partial system behaviour model without loss of information. That is, the independent refinement of the component MTSs to LTSs and their parallel composition results in exactly the set of distributable implementations of the monolithic MTS.

2 Background

We start with the familiar concept of labelled transition systems (LTSs) which are widely used for modelling and analysing the behaviour of concurrent and distributed systems [MK99]. An LTS is a state transition system where transitions are labelled with actions. The set of actions of an LTS is called its *alphabet*

and constitutes the interactions that the modelled system can have with its environment. An example LTS is shown in Figure 5(a).

Definition 1. (Labelled Transition System) *Let States be a universal set of states, and Act be the universal set of action labels. An* LTS *is a tuple* $I = \langle S, s^0, \Sigma, \Delta \rangle$*, where* $S \subseteq$ *States is a finite set of states,* $\Sigma \subseteq$ *Act is the set of labels,* $\Delta \subseteq (S \times \Sigma \times S)$ *is a transition relation, and* $s^0 \in S$ *is the initial state.*

Definition 2. (Bisimilarity) [Mil89] *Let LTSs* I *and* J *such that* $\alpha I = \alpha J$*.* I *and* J *are bisimilar, written* $I \sim J$*, if* (I, J) *is contained in some* bisimilarity relation B*, for which the following holds for all* $\ell \in Act$ *and for all* $(I', J') \in B$*:*

1. $\forall \ell \cdot \forall I'' \cdot (I' \xrightarrow{\ell} I'' \implies \exists J'' \cdot J' \xrightarrow{\ell} J'' \ \wedge \ (I'', J'') \in B)$.
2. $\forall \ell \cdot \forall J'' \cdot (J' \xrightarrow{\ell} J'' \implies \exists I'' \cdot I' \xrightarrow{\ell} I'' \ \wedge \ (I'', J'') \in B)$.

Definition 3 (Modal Transition System). $M = \langle S, s^0, \Sigma, \Delta^r, \Delta^p \rangle$ *is an* MTS *where* $\Delta^r \subseteq \Delta^p$*,* $\langle S, s^0, \Sigma, \Delta^r \rangle$ *is an LTS representing* required *behaviour of the system and* $\langle S, s^0, \Sigma, \Delta^p \rangle$ *is an LTS representing* possible *(but not necessarily required) behaviour.*

Every LTS $\langle S, s^0, \Sigma, \Delta \rangle$ can be embedded into an MTS $\langle S, s^0, \Sigma, \Delta, \Delta \rangle$. Hence we sometimes refer to MTS with the same set of required and possible transitions as LTS. We refer to transitions in $\Delta^p \setminus \Delta^r$ as *maybe* transitions, depict them with a question mark following the label. An example MTS is shown in Figure 2(a). We use $\alpha M = \Sigma$ to denote the communicating alphabet of an MTS M.

Given an MTS $M = \langle S, s^0, \Sigma, \Delta^r, \Delta^p \rangle$ we say M becomes M' via a required (possible) transition labelled by ℓ, denoted $M \xrightarrow{\ell}_{\mathrm{r}} M'$ ($M \xrightarrow{\ell}_{\mathrm{p}} M'$), if $M' = \langle S, s', \Sigma, \Delta^r, \Delta^p \rangle$ and $(s^0, \ell, s') \in \Delta^r$ ($(s^0, \ell, s') \in \Delta^p$). If (s^0, ℓ, s') is a maybe transition, i.e. $(s^0, \ell, s') \in \Delta^p \setminus \Delta^r$, we write $M \xrightarrow{\ell}_{\mathrm{m}} M'$.

Let $w = w_1 \ldots w_k$ be a word over Σ. Then $M \xrightarrow{w}_{\mathrm{p}} M'$ means that there exist $M_0, \ldots, M_k$ such that $M = M_0$, $M' = M_k$, and $M_i \xrightarrow{w_{i+1}}_{\mathrm{p}} M_{i+1}$ for $0 \leq i < k$. We write $M \xrightarrow{w}_{\mathrm{p}}$ to mean $\exists M' \cdot M \xrightarrow{w}_{\mathrm{p}} M'$. The language of an MTS M is defined as $\mathcal{L}(M) = \{w \in \alpha M \mid M \xrightarrow{w}_{\mathrm{p}}\}$. Finally we call optimistic implementation of M (M^+) the LTS obtained by making all possible transitions of M required.

Definition 4 (Parallel Composition). *Let* $M = \langle S_M, s^0_M, \Sigma, \Delta^r_M, \Delta^p_M \rangle$ *and* $N = \langle S_N, s^0_N, \Sigma, \Delta^r_N, \Delta^p_N \rangle$ *be MTSs.* Parallel composition *(*$\|$*) is a symmetric operator and* $M \| N$ *is the MTS* $\langle S_M \times S_N, (s^0_M, s^0_N), \Sigma, \Delta^r, \Delta^p \rangle$ *where* Δ^r *and* Δ^p *are the smallest relations that satisfy the rules in Figure 1.*

Parallel composition for MTSs with all transitions required (i.e. an LTS) is the same that parallel composition for LTSs [Mil89].

Strong refinement, or simply *refinement*[LT88], of MTSs captures the notion of elaboration of a partial description into a more comprehensive one, in which some knowledge of the maybe behaviour has been gained. It can be seen as being a "more defined than" relation between two partial models. An MTS N refines

$$\frac{M \xrightarrow{\ell}_{\mathrm{m}} M',\ N \xrightarrow{\ell}_{\mathrm{m}} N'}{M \| N \xrightarrow{\ell}_{\mathrm{m}} M' \| N'} \quad \frac{M \xrightarrow{\ell}_{\mathrm{m}} M',\ N \xrightarrow{\ell}_{\mathrm{r}} N'}{M \| N \xrightarrow{\ell}_{\mathrm{m}} M' \| N'} \quad \frac{M \xrightarrow{\ell}_{\mathrm{r}} M',\ N \xrightarrow{\ell}_{\mathrm{r}} N'}{M \| N \xrightarrow{\ell}_{\mathrm{r}} M' \| N'}$$

$$\frac{M \xrightarrow{\ell}_{\gamma} M',\ \ell \notin \alpha N,\ \gamma \in \{p,r\}}{M \| N \xrightarrow{\ell}_{\gamma} M' \| N} \quad \frac{\ell \notin \alpha M,\ N \xrightarrow{\ell}_{\gamma} N',\ \gamma \in \{p,r\}}{M \| N \xrightarrow{\ell}_{\gamma} M \| N'}$$

Fig. 1. Rules for parallel composition

M if N preserves all of the required and all of the proscribed behaviours of M. Alternatively, an MTS N refines M if N can simulate the required behaviour of M, and M can simulate the possible behaviour of N.

Definition 5. (Refinement) *Let MTSs N and M such that $\alpha M = \alpha N = \Sigma$. N is a* strong refinement *of M, written $M \preceq N$, if (M, N) is contained in some* strong refinement relation *R, for which the following holds for all $\ell \in Act$ and for all $(M', N') \in R$:*

1. $\forall \ell \in \Sigma, \forall M'' \cdot (M' \xrightarrow{\ell}_{\mathrm{r}} M'' \implies \exists N'' \cdot N' \xrightarrow{\ell}_{\mathrm{r}} N'' \ \wedge\ (M'', N'') \in R)$.
2. $\forall \ell \in \Sigma, \forall N'' \cdot (N' \xrightarrow{\ell}_{\mathrm{p}} N'' \implies \exists M'' \cdot M' \xrightarrow{\ell}_{\mathrm{p}} M'' \ \wedge\ (M'', N'') \in R)$.

Property 1. Refinement is a precongruence with regards to $\|$ meaning that if $M_i \preceq I_i$ for $i \in [n]$ then $\|_{i\in[n]} M_i \preceq \|_{i\in[n]} I_i$ where $[n] = \{1, \ldots, n\}$.

LTSs that refine an MTS M are complete descriptions of the system behaviour up to the alphabet of M. We refer to them as the *implementations* of M.

Definition 6. (Implementation) *We say that an LTS $I = \langle S_I, i^0, \Sigma, \Delta_I \rangle$ is an implementation of an MTS M, written $M \preceq I$, if $M \preceq M_I$ with $M_I = \langle S_I, i^0, \Sigma, \Delta_I, \Delta_I \rangle$. We also define the set of implementations of M as $\mathcal{I}[M] = \{LTS\ I \mid M \preceq I\}$.*

An MTS can be thought of as a model that represents the set of LTSs that implement it. The diversity of the set results from making different choices on the maybe behaviour of the MTS. As expected, refinement preserves implementations: $M \preceq M'$ then $\mathcal{I}[M] \supseteq \mathcal{I}[M']$.

Given a word $w \in \Sigma^*$ the projection of w onto $\Sigma_i \subseteq \Sigma$ ($w|_{\Sigma_i}$) is obtained by removing from w the actions not in Σ_i.

Let $A \subseteq \Sigma$, $M = \langle S, s^0, \Sigma, \Delta^p, \Delta^r \rangle$ and $s \in S$ then the closure of the state s over A is the set of states reachable from s using only transitions labelled by an action in A. Formally:

$$\mathcal{C}_A(s) = \{s' \mid s \xrightarrow{w}_{\mathrm{p}} s' \ \wedge\ w \in A^*\}$$

The projection of an MTS M over an alphabet Σ is an MTS $M|_\Sigma$ obtained from M by replacing the labels in M that are not in Σ by the internal action τ (written tau in the graphic representation of the MTS). Note that for any alphabet Σ in this paper holds that $\tau \notin \Sigma$.

We now discuss distribution of LTS models. Distribution of an LTS is with respect to a specification of component interfaces (the actions each component controls and monitors). Such specification is given by an alphabet distribution.

Given an alphabet Σ we say that $\Gamma = \langle \Sigma_1, \ldots, \Sigma_n \rangle$ is an alphabet distribution over Σ iff $\Sigma = \cup_{i \in [n]} \Sigma_i$ were each Σ_i is the (non-empty) alphabet of the local process i.

Definition 7 (Distributable LTS). *Given I, an LTS over Σ, and $\Gamma = \langle \Sigma_1, \ldots, \Sigma_n \rangle$ an alphabet distribution of Σ, I is* distributable *if there exist component LTSs $I_1, \ldots, I_n$ with $\alpha I_i = \Sigma_i$ such that $\|_{i \in [n]} I_i \sim I$.*

The distributed synthesis problem consists on deciding whether an LTS is distributable and, if so, build the distributed component LTSs. Unfortunately, it is unknown if deciding whether an LTS is distributable is decidable in general [CMT99]. However, it has been solved for weaker equivalence notions such as isomorphism [Mor98, CMT99] and language equivalence [CMT99, Ste06], and for restricted forms of LTS such as deterministic LTS [CMT99].

The following is a formal yet abstract distribution algorithm for determinstic LTS defined in terms of the procedure in [CMT99, Ste06]. The procedure builds the component I_i by projecting I over Σ_i and then determinising (using a subset construction [HU79]) I_i.

Definition 8 (LTS distribution). *Let $I = \langle S, s^0, \Sigma, \Delta \rangle$ be an LTS and Γ an alphabet distribution then the distribution of I over Γ is $\mathcal{DIST}_{\Gamma}^{\mathcal{LTS}}[I] = \{I_1, \ldots, I_n\}$ where $\forall i \in [1, n] \cdot I_i = \langle S_i, s_i^0, \Sigma_i, \Delta_i \rangle$ and:*

- *$S_i \in 2^S$ where S_i is reachable from the initial state following Δ_i.*
- *$s_i^0 = \mathcal{C}_{\overline{\Sigma_i}}(s_0)$.*
- *$(s, t, q) \in \Delta_i \;\leftrightarrow\; q = \bigcup_{k \in s} \{k'' \in \mathcal{C}_{\overline{\Sigma_i}}(k') \mid k \xrightarrow{t}_{\mathrm{p}} k'\}$.*

When Γ is clear from the context we just write $\mathcal{DIST}^{\mathcal{LTS}}[I]$.

Theorem 1 (LTS Distribution Soundness and Completeness). *[CMT99] Let I be a deterministic LTS, Γ an alphabet distribution and $\mathcal{DIST}_{\Gamma}^{\mathcal{LTS}}[I] = \{I_1, \ldots, I_n\}$ then I is distributable (and in fact $\|_{i \in [n]} I_i \sim I$) iff $\mathcal{L}(I) = \mathcal{L}(\|_{i \in [n]} I_i)$.*

3 MTS Distribution

A distribution of an MTS according to an alphabet distribution Γ is simply a set of component MTSs $\{M_1, \ldots, M_n\}$ such that $\alpha M_i = \Sigma_i$. Of course, a first basic requirement for a distribution of a system MTS into component MTSs is soundness with respect to refinement: any implementation of the component MTSs, when composed in parallel, yields an implementation of the system MTS (i.e. if $M_i \preceq I_i$ for $i \in [n]$ then $M \preceq \|_{i \in [n]} I_i$).

A second desirable requirement is completeness, meaning no distributable implementation is lost: a decomposition of M over Γ into a set of components $\{M_1, \ldots, M_n\}$ such that every distributable implementation of M is captured by the components. In other words, $\forall I$ implementation of M that is distributable over Γ there are I_i with $i \in [n]$ such that $M_i \preceq I_i$ and $\|_{i \in [n]} I_i \sim I$.

As discussed in the background section, multiple definitions of distribution for LTS exist. We restrict to deterministic implementations but take the most general distribution criteria, namely bisimilarity which under determinism is the same as language equivalence. The restriction to deterministic implementations is because as an LTS is also an MTS and MTS refinement applied to LTS is bisimulation, solving sound distribution for non-deterministic MTS would solve distribution for non-deterministic LTS considering bisimulation equivalence. The latter is not known to be decidable [CMT99].

Definition 9 (Deterministic and Distributable Implementations). *Let M be an MTS and Γ a distribution. We define $\mathcal{DDI}_\Gamma[M] = \{I \in \mathcal{I}[M] \mid I$ is deterministic and distributable over $\Gamma\}$.*

Definition 10 (Complete and Sound MTS Distributions). *Given an MTS M and an alphabet distribution Γ, a complete and sound distribution of M over Γ are component MTSs $M_1, \ldots, M_n$ such that $\alpha M_i = \Sigma_i$ and:*

1. *(soundness) for any set of LTSs $\{I_1, \ldots, I_n\}$, if $M_i \preceq I_i$ then $M \preceq \|_{i\in[n]} I_i$.*
2. *(completeness) for every $I \in \mathcal{DDI}_\Gamma[M]$ there are I_i with $i \in [n]$ where $M_i \preceq I_i$ and $\|_{i\in[n]} I_i \sim I$.*

A general result for distribution of MTS is not possible. There are MTS for which all their distributable implementations cannot be captured by a set of component MTSs.

Property 2. In general, a complete and sound distribution does not always exist.

Proof. Let's consider the MTS M in Figure 2(a) and the distribution $\Gamma = \langle \Sigma_1 = \{a, w, y\}, \Sigma_2 = \{b, w, y\}\rangle$. The MTSs in Figures 2(b) and 2(c) refine M. Let J and K be the optimistic implementations of the MTSs in Figures 2(b) and 2(c) respectively. As the MTSs in the aforementioned figures refine M, its implementations are also implementations of M. It is easy to see that J and K are both distributable over Γ. Then, a compact complete distribution of M should capture J and K. We shall show that in order to capture J and K the distribution cannot be sound.

Let M_1, M_2 be a complete distribution of M over Γ with $\alpha M_i = \Sigma_i$. As it is complete and J is distributable, there must be implementations of M_1 and M_2 that composed are bisimilar to J. Analogously, there must be implementations of M_1 and M_2 that composed are bisimilar to K. Let us consider a characteristic that an implementation J_1 of M_1 must have in order to yield J when composed with an implementation J_2 of M_2. As $J \xrightarrow{}$, $a \in \alpha M_1$ and $a \notin \alpha M_2$, it must be the case that $J_1 \xrightarrow{a}$.

The same reasoning can be applied to an implementation K_2 of M_2: In order to yield K when composed with an implementation K_1 of M_1, as $K \xrightarrow{b}$, $b \in \alpha M_2$ and $b \notin \alpha M_1$, it must be the case that $K_2 \xrightarrow{b}$. Hence, we have an implementation J_1 of M_1 such that $J_1 \xrightarrow{a}$ and an implementation K_2 of M_2 such that $K_2 \xrightarrow{b}$. This entails that $J_1 \parallel K_2 \xrightarrow{ab}$. As $M \overset{ab}{\not\longrightarrow}$ then $J_1 \parallel K_2$ is not a refinement of M.

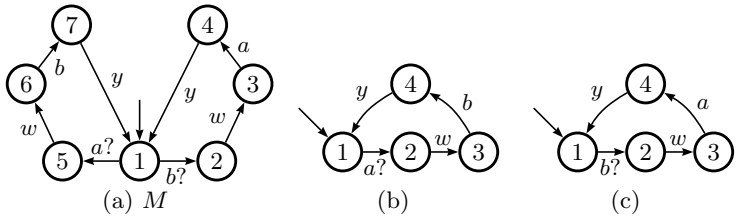


Fig. 2. MTSs used for proof of Property 2

Having assumed that M_1 and M_2 where a complete distribution of M over Γ we have concluded that it is not a sound distribution of M over Γ. □

This above property is reasonable: not all distributable implementations of an MTS can be achieved by refining independently partial specifications of components. Some decisions (or lack of them) regarding system behaviour captured in the system MTS may require coordinated refinement of component MTSs. In the counter-example described above, the system MTS states that either a or b will occur initially but not both. The decision on which will be provided in the final implementation requires coordinated refinement of the component models: Either J provides a and K does not provide b or the other way round.

3.1 Distribution of a Deterministic MTS

Despite negative result in Property 2 there is a relevant class of MTSs for which a sound and complete distribution is guaranteed to exist and for which an algorithm that produces such distribution can be formulated. The class is that of deterministic MTSs which assign modalities consistently and their optimistic implementation (M^+) is a distributable LTS.

We first give an overview of the distribution algorithm for MTS, then prove soundness of the distributions produced by the algorithm, then define modal consistency of transitions and prove the distributions produced by the algorithm are also complete under modal consistency.

The distribution algorithm requires a deterministic MTS M for which its optimistic implementation M^+ is a distributable LTS. The algorithm builds on the LTS distribution algorithm for deterministic LTS under bisimulation equivalence (see Background). The main difference is that it associates modalities to transitions of component models it produces based on the modalities of the system MTS.

As a running example consider the MTS N in Figure 3 with alphabet $\Sigma = \{a, b, c, d\}$ and the alphabet distribution $\Gamma = \langle \Sigma_1 = \{a, b\}, \Sigma_2 = \{b, c, d\} \rangle$. Conceptually, the algorithm projects N^+ onto the component alphabets and determinises each projection. The modality of a component MTS transition is set to required if and only if at least one of its corresponding transitions in the system MTS is required. The projections of N^+ on Σ_1 and Σ_2 are depicted in Figure 4, the deterministic versions of these projections are depicted in Figure 5, and the

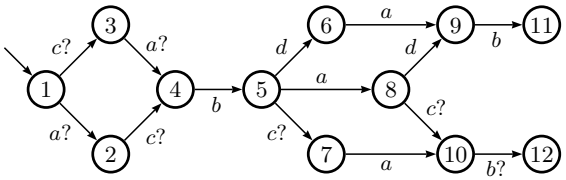
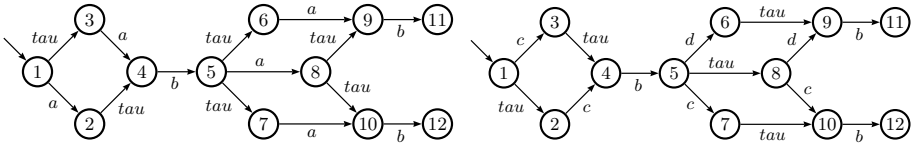


Fig. 3. Running example: N



(a) Projected onto Σ_1. (b) Projected onto Σ_2.

Fig. 4. N^+ projected onto the local alphabets

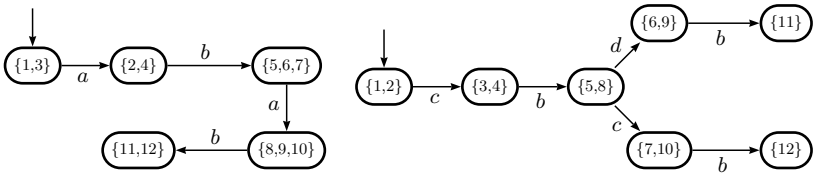
component MTS resulting from adding modalities to transitions is depicted in Figure 6. Note that the numbers in states of the deterministic MTS in Figures 5 and 6 correspond to the states of N as a result of determinisation.

We now present a formal yet abstract distribution algorithm defined in terms of the subset construction for determinising LTS models [HU79] and the LTS distribution algorithm in [Ste06].

Definition 11 (MTS distribution). *Let $M = \langle S, s^0, \Sigma, \Delta^p, \Delta^r \rangle$ be an MTS and Γ a distribution then the distribution of M over Γ is $\mathcal{DIST}_{\Gamma}^{\mathcal{MTS}}[M] = \{M_1, \ldots, M_n\}$ where $\forall i \in [1,n] M_i = \langle S_i, s_i^0, \Sigma_i, \Delta_i^p, \Delta_i^r \rangle$ and:*

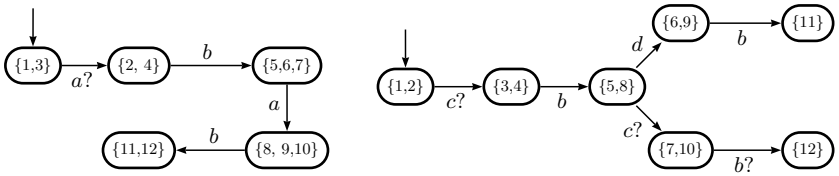
- $S_i \in 2^S$ *where S_i is reachable from the initial state following Δ_i^p.*
- $s_i^0 = \mathcal{C}_{\overline{\Sigma_i}}(s_0)$.
- $(s,t,q) \in \Delta_i^p \;\leftrightarrow\; q = \bigcup_{k \in s} \{k'' \in \mathcal{C}_{\overline{\Sigma_i}}(k') \mid k \xrightarrow{t}_{\mathrm{p}} k'\}$.
- $(s,t,q) \in \Delta_i^r \;\leftrightarrow\; (s,t,q) \in \Delta_i^p \wedge \exists k \in s \cdot k \xrightarrow{t}_{\mathrm{r}}$.

When Γ is clear from the context we just write $\mathcal{DIST}^{\mathcal{MTS}}[M]$.



(a) Projected onto Σ_1 (b) Projected onto Σ_2

Fig. 5. N^+ projected onto the local alphabets and determinised



(a) Component N_1. (b) Component N_2

Fig. 6. Distribution of MTS in Figure 3

Note that in component N_1 of Figure 6 the required b transition from state $\{8, 9, 10\}$ to $\{11, 12\}$ is a consequence of the required b transition from 9 to 11 and the maybe b transition from 10 to 12 in N. Had the transition from $\{8, 9, 10\}$ to $\{11, 12\}$ in N_1 been a maybe rather than required then the distribution would not be sound. Let N_1' be such component. N_1' allows an implementation as in Figure 5(a) but without the last b transition from $\{8, 9, 10\}$ to $\{11, 12\}$. We refer to this implementation as I^1: $I^1 \xrightarrow{aba}_{\mathrm{p}} \not\xrightarrow{b}$. Let I^2 be the LTS in Figure 5(b). I^2 is actually an implementation of N_2. But $I^1 \parallel I^2$ is not an implementation of N as $I^1 \parallel I^2 \xrightarrow{acbad}_{\mathrm{p}} \not\xrightarrow{b}$ and $N \xrightarrow{acbad}_{\mathrm{p}} \xrightarrow{b}_{\mathrm{r}}$. Hence the need to make the b transition $\{8, 9, 10\}$ to $\{11, 12\}$ required in order to ensure soundness.

We now discuss soundness of MTS distributions as constructed in Definition 11. First, note that Definition 11 when applied to LTS is equivalent to Definition 8, that is the distribution constructed when the MTS is a deterministic LTS is, in effect, a distribution of the LTS. What follows is a sketch of the more general soundness proof.

Theorem 2 (Soundness). *Let M be a deterministic MTS and Γ a distribution such that M^+ is a distributable LTS over Γ, then the MTS distribution (Definition 11) is sound (as defined in Definition 10).*

Proof. We need to prove that for any $I_1, \ldots, I_n$ such that $M_i \preceq I_i$ then $M \preceq \|_{i\in[n]} I$. As refinement is a precongruence with regards to $\|$ meaning that if $M_i \preceq I_i$ for $i \in [n]$ then $\|_{i\in[n]} M_i \preceq \|_{i\in[n]} I_i$ we just need to prove $M \preceq \|_{i\in[n]} M_i$. Thus $M \preceq \|_{i\in[n]} I_i$.

We now prove $M \preceq \|_{i\in[n]} M_i$. M^+ is distributable and the component MTSs produced by $\mathcal{DIST}^{\mathcal{MTS}}[M]$ are isomorphic, without considering the transitions' modality, to the component LTSs produced by $\mathcal{DIST}^{\mathcal{LTS}}[M^+]$. So the parallel composition of the component MTSs is isomorphic, again without considering the transitions' modality, to the parallel composition of the component LTSs. When the component MTSs are created if, after the closure, there is a required transition then the component will have a required transition and so the composition may have a required transition where the monolithic MTS had a maybe transition. But any possible behaviour in the composed MTS is also possible

in the monolithic MTS. Therefore the composed MTS is a refinement of the monolithic MTS. □

We now define modal consistency of transitions, which is one of the conditions for Definition 11 to produce complete distributions.

We say that the modalities of an MTS M are inconsistent with respect to an alphabet distribution Γ when there is an action ℓ such that there are two traces w and y leading to two transitions with different modalities on ℓ (i.e. a required and a maybe ℓ-transition) and that for each component alphabet $\Sigma_i \in \Gamma$ where $\ell \in \Sigma_i$, the projection of w and y on Σ_i are the same.

The intuition is that if M is going to be distributed to deterministic partial component models, then some component contributing to the ocurrence of the ℓ after w and y must have reached both points through different paths (i.e. $w|_{\Sigma_i} \neq y|_{\Sigma_i}$). If this is not the case, then the distribution will have to make ℓ after w and y always maybe or always required.

Definition 12 (Alphabet Distribution Modal Consistency). *Let Γ be an alphabet distribution and $M = \langle S, s_0, \Sigma, \Delta^r, \Delta^p \rangle$ an MTS then M is* modal consistent *with respect to Γ iff $\forall w, y \in \Sigma^*, \ell \in \Sigma \cdot M \xrightarrow{w}_{\mathrm{p}} \xrightarrow{\ell}_{\mathrm{r}} \wedge M \xrightarrow{y}_{\mathrm{p}} \xrightarrow{\ell}_{\mathrm{m}}$ implies $\exists i \in [n] \cdot \ell \in \Sigma_i \wedge w|_{\Sigma_i} \neq y|_{\Sigma_i}$.*

Consider model N from Figure 3. This MTS is modal consistent for $\Gamma = \langle \Sigma_1 = \{a, b\}, \Sigma_2 = \{b, c, d\} \rangle$ as the only w, y and ℓ such that $N \xrightarrow{w}_{\mathrm{p}} \xrightarrow{\ell}_{\mathrm{m}}$ and $N \xrightarrow{y}_{\mathrm{p}} \xrightarrow{\ell}_{\mathrm{m}}$ are $\ell = b$, and w and y sequences leading to states 9 and 10 (for instance $w = cabda$ and $y = acbac$). However, all sequences leading to 9 when projected onto Σ_2 yield *cbd* while those leading to 10 yield *cbc*. Hence, consistency is satisfied.

Now consider model P in Figure 7 (a modified version of N but with the following modalities changed: $5 \xrightarrow{a}_{\mathrm{m}} 8$ and $6 \xrightarrow{a}_{\mathrm{m}} 9$). P is not modal consistent with respect to $\Gamma = \langle \Sigma_1 = \{a, b\}, \Sigma_2 = \{b, c, d\} \rangle$: Now there are $w = acb$ and $y = acbc$ such that $P \xrightarrow{w}_{\mathrm{p}} \xrightarrow{a}_{\mathrm{m}}$ and $P \xrightarrow{y}_{\mathrm{p}} \xrightarrow{a}_{\mathrm{m}}$ yet the only Σ_i that includes a is Σ_1 and $w|_{\Sigma_1} = y|_{\Sigma_1} = ab$.

A sound and complete distribution of P would require a deterministic component MTS for $\Sigma_1 = \{a, b\}$ that would either require a after ab or have a maybe a after ab. The former would disallow the implementation I_1 of Figure 8(b) which in turn would make impossible having a component implementation I_2 such that $I_1 \parallel I_2$ yields I of Figure 8(a) which is a deterministic distributable implementation of P. Hence requiring a after ab would lead to an incomplete distribution. Choosing the latter would allow implementation I_1 which would make the distribution unsound: In order to have implementations that when composed yield P^+, an implementation with alphabet $\Sigma_2 = \{b, c, d\}$ bisimilar to Figure 8(c) is needed. However, such an implementation, when composed with I_1 is not a refinement of P.

Theorem 3 (Completeness). *Let M be a deterministic MTS and Γ a distribution such that M^+ is a distributable LTS over Γ, and M is modal consistent*

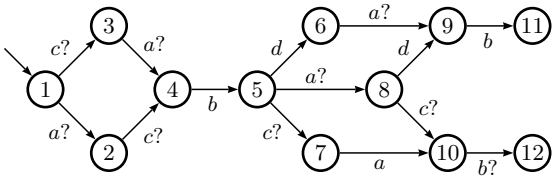
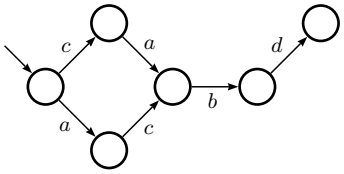
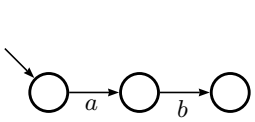


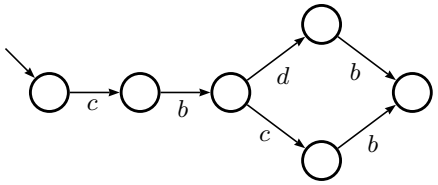
Fig. 7. P: Modal Inconsistent MTS



(a) Implementation of Figure 7



(b) Component P_1^+



(c) Component P_2^+

Fig. 8.

then the MTS distribution (Definition 11) is complete (as defined in Definition 10).

The proof of this theorem uses the following lemmas:

Lemma 1. *Let M, N be deterministic MTSs with $\alpha N = \alpha M$ if $\forall w \in \Sigma^*, t \in \Sigma$*

- $N \xrightarrow{w}_{\mathrm{p}} \implies M \xrightarrow{w}_{\mathrm{p}}$.
- $N \xrightarrow{w}_{\mathrm{p}} \wedge M \xrightarrow{w}_{\mathrm{p}} M' \xrightarrow{t}_{\mathrm{r}} \implies N' \xrightarrow{t}_{\mathrm{r}}$.

Then $M \preceq N$.

Lemma 2. *Let M be an MTS and $I \in \mathcal{DDI}[M]$. For every $\Sigma_i \in \Gamma$ let M_i and I_i be the components corresponding to Σ_i in $\mathcal{DIST}^{\mathcal{MTS}}[M]$ and $\mathcal{DIST}^{\mathcal{LTS}}[I]$ respectively then $\forall w \in \Sigma_i \cdot I_i \xrightarrow{w}_{\mathrm{p}} \implies M_i \xrightarrow{w}_{\mathrm{p}}$.*

Proof (Theorem 3). Let $\mathcal{DIST}_{\Gamma}^{\mathcal{MTS}}[M] = \{M_1, \ldots, M_n\}$. We need to prove that for every $I \in \mathcal{DDI}_{\Gamma}[M]$ there are I_i with $i \in [n]$ where $M_i \preceq I_i$ and $\|_{i\in[n]} I_i \sim I$. As I is distributable over Γ then $\mathcal{DIST}_{\Gamma}^{\mathcal{LTS}}[I] = \{Q_1 \ldots Q_n\}$ and $\|_{i\in[n]} Q_i \sim I$.

Recall that the distribution algorithms produce deterministic components. Therefore we can use Lemma 1 to show that each MTS component is refined

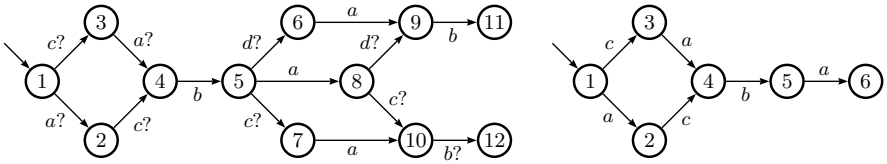
by its corresponding LTS component. Let M_i and Q_i be the MTS and LTS components for $\Sigma_i \in \Gamma$. Every possible trace in Q_i is possible in M_i (Lemma 2). Then the only way Q_i is not a refinement of M_i is because there is some required behaviour in M_i that is not present in Q_i. So lets suppose $M_i \not\preceq Q_i$, then $\exists z \in \Sigma_i^*, t \in \Sigma_i$ such that $M_i \xrightarrow{z}_{\mathrm{p}} T \xrightarrow{t}_{\mathrm{r}} \wedge Q_i \xrightarrow{z}_{\mathrm{p}} Q \overset{t}{\not\longrightarrow}$.

We now present an algorithm that creates, for every $i \in [n]$, a new component I_i from Q_i by adding the missing required transitions from M_i in order to get $M_i \preceq I_i$. The algorithm iteratively takes a pair (M_i, I_i^j), where I_i^j is the component I_i constructed up to iteration j, such that $M_i \not\preceq I_i^j$ and adds a required transition for a pair mirroring M_i structure. The structure of M_i has to be kept in the resulting I_i in order to avoid trying to add infinite required transitions due to a loop of required transitions in M_i. If the added transitions are part, and complete, a loop in M_i then that same loop will be created in I_i when the algorithm adds the required transitions. Furthermore, the added transitions do not modify the composition (Lemma 3).

Algorithm 1. Extension to each Q_i to get a refinement of M_i

Input: $\{(M_1, Q_1), \ldots, (M_n, Q_n)\}$
Output: $\{I_1, \ldots, I_n\}$
  $I_1 = Q_1; \ldots; I_n = Q_n;$
  while $\exists i \in [n] \cdot M_i \not\preceq I_i$ **do**
    take $(M_i, I_i) \cdot M_i \not\preceq I_i$;
    take $z \in \Sigma_i^* \cdot M_i \xrightarrow{z}_{\mathrm{p}} P \xrightarrow{t}_{\mathrm{r}} P' \wedge I_i \xrightarrow{z} Q \overset{t}{\not\longrightarrow}$;
    if $\exists u \in \Sigma_i^* \cdot M_i \xrightarrow{u}_{\mathrm{p}} P' \wedge I_i \xrightarrow{u} Q'$ **then**
      $Q \xrightarrow{t} Q'$;
    else
      Add a new state Q' to I_i and then the transition $Q \xrightarrow{t} Q'$;
    end if
  end while

As an example of how the algorithm works consider the MTS E in Figure 9(a), that is like N from Figure 3 only that the d transitions are maybe in E instead of required, and $\Gamma = \langle \Sigma_1 = \{a, b\}, \Sigma_2 = \{b, c, d\} \rangle$. Let $\mathcal{DIST}^{\mathcal{MTS}}[E] = \{E_1, E_2\}$. E_1 is the same as component N_1 in Figure 6(a). E_2 is like component N_2 in Figure 6(b) only that the d transition from $\{5, 8\}$ to $\{6, 9\}$ is a maybe d transition. I^E in Figure 9(b) is an implementation of E and $\mathcal{DIST}^{\mathcal{LTS}}[I^E] = \{Q_1, Q_2\}$ (Figure 10(a) and 10(b)). The algorithm takes $\{(E_1, Q_1), (E_2, Q_2)\}$ and returns components I_1 (I_1 is the same as the LTS in Figure 5(a)) and I_2 (I_2 is in fact Q_2). As $E_2 \preceq Q_2$ then the algorithm will not change Q_2 so $I_2 = Q_2$. $E_1 \not\preceq Q_1$ because $E_1 \xrightarrow{aba}_{\mathrm{p}} \xrightarrow{b}_{\mathrm{r}}$ and $Q_1 \xrightarrow{aba}_{\mathrm{p}} \overset{b}{\not\longrightarrow}$. The algorithm then adds the missing transition to Q_1 and the result is I_1 (I_1 is the same as the LTS in Figure 5(a)). Now $E_1 \preceq I_1$ and the algorithm finishes. See how $I_1 \parallel I_2 \sim Q_1 \parallel Q_2$ as the added b transition to Q_1 in I_1 does not appear in the composition because Q_2 does not provide the needed synchronisation.



(a) E (b) I^E.

Fig. 9.

(a) Q_1 (b) Q_2.

Fig. 10.

Finally we prove that the algorithm finishes. As there are finite components it is sufficient to show that $M_i \preceq I_i^m$ with m finite where I_i^j is I_i after doing j additions of required transitions to I_i.

Each iteration adds a missing required t transition to a I_i^j that is present in M_i. If the required transition in M_i goes to P' and there is a $u \in \Sigma_i^*$ from M_i to P' such that u is possible in I_i^j leading to Q' then the new transition goes to Q'. Q' is already present in I_i^{j-1} and the algorithm never modifies possible transitions so any possible behaviour in I_i^{j-1} is possible in M_i and the same stands for I_i^j. On the other hand, if P' is not reachable by a word that is possible in I_i^j then the added required transition goes to a new state. This procedure modifies I_i until all reachable required transitions in M_i not present in I_i are added. As loops of required transitions in M_i that have to be added to I_i are added preserving the loop structure then the iterations for component M_i can not be more than the amount of required transitions present in M_i. And this is done for every pair of components but as they are n the algorithm finishes. □

The following lemma is used in the proof of Theorem 3. For all $i \in [n]$ I_i refines M_i and the added transitions do not modify the composition. Formally:

Lemma 3. *Let M be a deterministic MTS such that M^+ is distributable over Γ and modal consistent. Let $I \in \mathcal{DDI}[M]$, $\mathcal{DIST}^{\mathcal{LTS}}[I] = \{Q_1, \ldots, Q_n\}$, $\mathcal{DIST}^{\mathcal{MTS}}[M_i] = \{M_1, \ldots, M_n\}$ and $\{I_1, \ldots, I_n\}$ the output of Algorithm 1 for $\{(M_1, Q_1), \ldots, (M_n, Q_n)\}$ then:*

- $\forall i \in [n]$ $M_i \preceq I_i$.
- $\|_{i\in[n]} I_i \sim \|_{i\in[n]} Q_i$ *(and therefore $\|_{i\in[n]} I_i \sim I$).*

4 Related Work

Distributed implementability and synthesis has been studied for LTS for different equivalences notion like isomorphism, language equivalence and

bisimulation [Mor98, CMT99, Ste06, HS05]. The general distributed implementability problem has not been studied for MTS.

A component view of the system has been taken in the context of studies on parallel composition of MTS [BKLS09b], however such view is bottom-up: Given partial behaviour models of components, what is the (partial) behaviour of the system resulting of their parallel composition. The only notable example that takes a top-down approach is [KBEM09] A synthesis procedure is proposed that given system level OCL properties and UML scenarios, component partial behaviour models are automatically constructed such that their composition requires the behaviour required by system level properties and scenarios, and proscribes the behaviour not permitted by the same properties and scenarios.

In [QG08], MTS distribution is studied as a instance of more general contract-based formalism. The notion that corresponds to our definition of complete and sound MTS Distribution (see Definition 10) is called decomposability, Definition 3.8 [QG08]. Decomposability is a strictly stronger notion which requires all implementations of M to be captured by some distribution $\|_{i\in[n]}M_i$. Our definition only requires distributable implementations of M to be refinements of $\|_{i\in[n]}M_i$. In particular Figure 3, with transition from 6 to 9 changed to being only possible, is not distributable according to [QG08] but is according to our definition. Moreover, the distribution algorithm of [QG08] cannot handle examples such as Figure 3.3 in [Ste06] which can be handled by standard LTS distribution algorithms (and ours) by determinising projections.

5 Conclusions

In this paper we provide results that support moving from iterative refinement of a monolithic system models to component-wise iterative refinement. We present a distribution algorithm for partial behaviour system models specified as MTS to component-wise partial behaviour models given as sets of MTSs. We precisely characterise when the decomposition provided is sound and complete, we also discuss why the restrictions to the distribution problem (namely determinism, modal consistency and distributability of M^+) are reasonable, are unlikely to be avoidable for any sound and complete distribution method, and can be seen as a natural extension of the limitations of existing LTS distribution results.

Future work will involve experimenting with case studies to assess the practical limitations imposed by the restrictions introduced to enforce completeness of distributions. We expect insights gained to allow for definition of more generally applicable sound but not complete distribution algorithms and elaboration techniques to support refinement of system models into models for which distribution algorithms exist.

References

[BKLS09a] Beneš, N., Křetínský, J., Larsen, K.G., Srba, J.: Checking Thorough Refinement on Modal Transition Systems Is EXPTIME-Complete. In: Leucker, M., Morgan, C. (eds.) ICTAC 2009. LNCS, vol. 5684, pp. 112–126. Springer, Heidelberg (2009)

[BKLS09b] Beneš, N., Ketínský, J., Larsen, K.G., Srba, J.: On determinism in modal transition systems. Theor. Comput. Sci. 410(41), 4026–4043 (2009)

[CMT99] Castellani, I., Mukund, M., Thiagarajan, P.S.: Synthesizing Distributed Transition Systems from Global Specifications. In: Pandu Rangan, C., Raman, V., Sarukkai, S. (eds.) FST TCS 1999. LNCS, vol. 1738, pp. 219–231. Springer, Heidelberg (1999)

[DFFU07] D'Ippolito, N., Fishbein, D., Foster, H., Uchitel, S.: MTSA: Eclipse support for modal transition systems construction, analysis and elaboration. In: Eclipse 2007: Proceedings of the 2007 OOPSLA Workshop on Eclipse Technology Exchange, pp. 6–10. ACM (2007)

[FBD+11] Fischbein, D., Brunet, G., D'Ippolito, N., Chechik, M., Uchitel, S.: Weak alphabet merging of partial behaviour models. In: TOSEM, pp. 1–49 (2011)

[GP11] Godefroid, P., Piterman, N.: Ltl generalized model checking revisited. STTT 13(6), 571–584 (2011)

[HS05] Heljanko, K., Stefanescu, A.: Complexity results for checking distributed implementability. In: Proc. of the Fifth Int. Conf. on Application of Concurrency to System Design, pp. 78–87. IEEE Computer Society Press (2005)

[HU79] Hopcroft, J.E., Ullman, J.D.: In: Introduction to automata theory, languages, and computation. Addison-Wesley (1979)

[KBEM09] Krka, I., Brun, Y., Edwards, G., Medvidovic, N.: Synthesizing partial component-level behavior models from system specifications. In: ESEC/FSE 2009, pp. 305–314. ACM (2009)

[LT88] Larsen, K.G., Thomsen, B.: A modal process logic. In: LICS 1988, pp. 203–210. IEEE Computer Society (1988)

[Mil89] Milner, R.: Communication and Concurrency. Prentice-Hall, New York (1989)

[MK99] Magee, J., Kramer, J.: Concurrency - State Models and Java Programs. John Wiley (1999)

[Mor98] Morin, R.: Decompositions of Asynchronous Systems. In: Sangiorgi, D., de Simone, R. (eds.) CONCUR 1998. LNCS, vol. 1466, pp. 549–564. Springer, Heidelberg (1998)

[QG08] Quinton, S., Graf, S.: Contract-based verification of hierarchical systems of components. In: SEFM 2008, pp. 377–381 (2008)

[Ste06] Stefanescu, A.: Automatic Synthesis of Distributed Systems. PhD thesis (2006)

[Sto05] Stoll, M.: MoTraS: A Tool for Modal Transition Systems. Master's thesis, Technische Universitat Munchen, Fakultat fur Informatik (August 2005)

[SUB08] Sibay, G., Uchitel, S., Braberman, V.: Existential live sequence charts revisited. In: ICSE 2008, pp. 41–50 (2008)

[UKM04] Uchitel, S., Kramer, J., Magee, J.: Incremental elaboration of scenario-based specifications and behaviour models using implied scenarios. ACM TOSEM 13(1) (2004)