Epistemic modelling and protocol dynamics

Wang, Y.

Citation for published version (APA):
Chapter 5

Composing Models

5.1 Introduction

In Part I, we proposed and studied several variants of PDL as logics for reasoning about knowledge, protocol and change. As we have argued, to verify epistemic protocols by model checking, it is important to have the right model. However, as argued in [vB09], to build a model is an art. For real life applications, the initial models and the action models can be huge (see, e.g., [DW07] for a case study in a protocol verification setting). Thus some (semi-) automatic method is inevitable in dynamic epistemic modelling. In this part, we make some modest contributions to dynamic epistemic modelling.

Our first inspiration is from the ETL approaches where the temporal epistemic models (interpreted systems) are generated in a distributed fashion with each agent acting as a component (cf. Definition 2.3.1). This distributed feature made ETL very handy in modelling various multi-agent scenarios [FHMV95], for example, adding one extra agent is done by adding one more component. On the other hand, DEL models are apparently not inherently distributed at the first glance: the static models and action models contain information about all the agents. For example, in DEMO [vE07], an implementation of DEL model checking, the initial models are generated by first considering an universal ignorance model, where the agents do not know any atomic proposition, and then restricting it by announcements. It is clear that this method cannot generate all the desired initial models nor is it easy to handle extra agents. To make the DEL approach more applicable for real-life applications, it is crucial to build a static model (or an event model) from local components according to each agent’s perspective. This clearly gives rise to the need for a formal way for composing models.

In fact, a clue is hinted at in DEL itself: the product update is indeed a way of composing models, though between two different types of models (static models vs. event models). A straightforward idea is to extend this product operation to compositions between two static models or two event models. However, there is a difficulty in defining such a composition operation: the models we want to compose
may have different vocabularies. Note that in practice, one agent may be able to observe only part of the atomic facts (propositions) in the whole vocabulary, e.g., in the Muddy Children scenario each child can see whether other children are muddy but can make no observations about herself. Thus it is reasonable to divide the whole vocabulary into parts (sets of observables) and build “partial models” for agents according to their local vocabularies.

In this chapter, we define and demonstrate the use of merging composition between static models and between action models with arbitrary vocabularies. For example, we show how a 2^n worlds Muddy Children model can be viewed as a composition of n two-node models, each talking only about the muddiness of a single child. Next, we extend standard event model update to an update operation which works on static models and event models with different vocabularies, by incorporating vocabulary expansion in the update process. We also look at the models generated in the distributed fashion of ETL and claim that our merging composition with arbitrary vocabularies can achieve the same goal in the DEL setting.

Related work Our merging composition of event models may be viewed as a notion of parallel composition of events. The first concurrent operation in the framework of DEL has been introduced in [vDvdHK03a, vDvdHK03b], where the authors follow the treatment of concurrency as in concurrent PDL [Pe87]. The concurrent operator \( \cap \) as in [vDvdHK03c] essentially splits the system into copies with each copy executing a concurrent component (see also [vDvdHK07, Chapter 5] for details). In some sense, composing actions in concurrent DEL may be viewed as merging agents who are acting differently, while in this chapter we focus on merging propositional information which is distributed among agents in both static and event models. Compared to the large body of research about parallel compositions in various process algebra frameworks (e.g., [Mil82, BK85, BHR84, GP94]), the distinct feature of our operator is the merging of different vocabularies and preconditions. The restriction to epistemic models (S5 models) also gives specific results meaningful in the epistemic setting.

Structure of the chapter In Section 5.2 we introduce the operator of merging composition on static models, under which the Kripke models form a commutative monoid. We then structurally characterize the induced pre-order by this monoid. Based on the merging composition, we study a natural operation which expands a model with a larger vocabulary. Various logical preservation results between the components and the composed model are proved. Section 5.3 addresses the problem of decomposition by looking at a specific class of models which are useful in a multi-agent setting. We demonstrate that we can decompose a model either by agents or by issues. We introduce the composition of event models and the extended product update in Section 5.4. We show that under certain conditions the action update distributes over merging composition. We point out some future directions in the last section.
5.2 Composing Static Models

5.2.1 Merging Composition

Recall that an S5 Kripke model $M$ is a tuple $(S, P, I, \sim, V)$ where $P$ is the (finite) vocabulary, $I$ is the (finite) set of agents, $\sim_i$ is an equivalence relation for each $i \in I$, and the valuation function $V : S \mapsto 2^P$ assigns a set of atomic propositions to each state. Given a Kripke model $M$, we use $S_M, P_M, I_M, \sim_M$ and $V_M$ to denote the corresponding elements in the definition of $M$. In this chapter we consider the compositions of models with different vocabularies but the same set of agents. We define the unit model $E$ as the model $(\{s\}, \emptyset, I, \sim, V)$ where $V(s) = \emptyset$ and $\sim_i = \{(s, s)\}$ for any $i$. In a picture:

$\emptyset \quad I$

Now we define the merging composition of two S5 models with arbitrary vocabularies.

5.2.1. Definition. (Merging Composition of Kripke Models) Given two models with the same set of agents $I : M = (S, P, I, \sim, V)$ and $N = (T, P', I, \sim', V')$, the merging composition $M \oplus N$ is given by $(S'', P \cup P', I, \sim'', V'')$, where:

- $S'' = \{(s, t) \mid s \in S, t \in T, V(s) \cap P' = V'(t) \cap P\}$,
- $(s, s') \sim''_i (t, t')$ iff $s \sim_i t$ and $s' \sim'_i t'$,
- $V''(s, t) = V(s) \cup V'(t)$.

Intuitively, the accessibility relations in the composed model are defined by “synchronizing” the corresponding relations in the components, in the usual way as in product updates, restricted to the pairs of worlds where the old valuations agree on the common vocabulary $P \cap P'$. It is clear that $V(s, s')$ agrees with $V(s)$ on $P$ and with $V'(s')$ on $P'$, thus merging the two component valuations. We say a state $s$ in $M$ is compatible with a state $t$ in $N$ if $(s, t)$ is in the composition $M \oplus N$. It is not hard to verify that any merging composition of S5 models is again an S5 model.

As a first example, here is a “compositional version” of the 2-Muddy Children scenario:

\[
\begin{array}{ccc}
  m_1 \quad 1 \quad \overline{m_1} & m_2 & m_1m_2 \quad 1 \quad \overline{m_1m_2} \\
  \oplus & 2 & \oplus & 2 & \oplus & 2 \\
  \overline{m_2} & \overline{m_1} & \overline{m_2} & \overline{m_1m_2} & \overline{m_1m_2} & \overline{m_1m_2}
\end{array}
\]
where \( m_i \) expresses “child \( i \) is muddy”, the set of agents is \( \{1, 2\} \), the vocabulary of each model \( m_i \) is \( \{m_i\} \), and as usual we leave out reflexive arrows which are present for all agents. Intuitively, each 2-world model represents the children’s observational power on whether child \( i \) is muddy, e.g., \( m_i \leftarrow \bar{m}_i \) captures the situation that child \( i \) does not know whether she herself is muddy while all the others do know whether child \( i \) is muddy.

In the same fashion, composing the above models with a third model does give the 3-Muddy Children model:

5.2.2. Example. Composing Muddy Children

Multidimensional hypercubes with more and more children present can be composed in the same way.

Here is an example of composing models with intersected vocabularies:

Note that according to our definition, self-composition \( M \oplus M \) is not always bisimilar to \( M \). Consider the model \( M \):

\[ M : \begin{array}{c}
p \downarrow 1 \bar{p} \\
2 \\
p \downarrow 3 \bar{p}
\end{array} \]

\(^1\)Note that the set of agents needs to be extended from \( \{1, 2\} \) to \( \{1, 2, 3\} \).
Let us call the upper and lower $p$ worlds $s$ and $t$ respectively. It is clear that $(s, t)$ is in the composed model $M \uplus M$ and $V_{M \uplus M}(s, t) = \{p\}$. However, according to the definition of relations in the composed model, $(s, t)$ cannot reach a $\neg p$ world by just one step in the composed model. Therefore $(s, t)$ is not bisimilar to any world in $M$. Nevertheless, Kripke models with different vocabularies do form a commutative monoid:

5.2.3. **Theorem.** Kripke models with the same set of agents form a commutative monoid under the $\uplus$ operation, with total bisimilarity (see Definition 2.2.2) as the appropriate equality notion. In particular, we have:

$$
E \uplus M \sqsubseteq M \\
M \uplus E \sqsubseteq M \\
M \uplus (N \uplus K) \sqsubseteq (M \uplus N) \uplus K \\
M \uplus N \sqsubseteq N \uplus M
$$

**Proof** Commutativity and axioms about the unit are immediate. We only check associativity here. Let $A(l)_{M}^{x} y$ be the abbreviation of $V_{M}(l) \cap P_{y}$ for $l \in \{s, t, k\}$ and $x, y \in \{M, N, K\}$, e.g., $A(s)_{M}^{N}$ represents $V_{M}(s) \cap P_{N}$. Thus the condition

$$
PC := (A(s)_{M}^{N} = A(t)_{M}^{N} \text{ and } A(s)_{K}^{M} = A(k)_{K}^{M} \text{ and } A(t)_{K}^{N} = A(k)_{K}^{N})
$$

expresses that $s, t, k$ are pairwise compatible. A moment of reflection should assure that:

$$(s, (t, k)) \in S_{M \uplus (N \uplus K)} \iff PC \iff ((s, l), k) \in S_{(M \uplus N \uplus K)}$$

Then it is straightforward to see that $M \uplus (N \uplus K) \sqsubseteq (M \uplus N) \uplus K$.

Note that $\sqsubseteq$ is indeed a congruence of this monoid:

5.2.4. **Proposition.** If $M_{1} \sqsubseteq M_{2}$ and $N_{1} \sqsubseteq N_{2}$ then $M_{1} \uplus N_{1} \sqsubseteq M_{2} \uplus N_{2}$

**Proof** Let $Z_{1}$, $Z_{2}$ be the total bisimulations witnessing $M_{1} \sqsubseteq M_{2}$ and $N_{1} \sqsubseteq N_{2}$ respectively. Then the relation $Z \subseteq S_{M_{1} \uplus N_{1}} \times S_{M_{2} \uplus N_{2}}$ defined by:

$$(s_{1}, t_{1})Z(s_{2}, t_{2}) \iff s_{1}Z_{1}s_{2} \text{ and } t_{1}Z_{2}t_{2}$$

is clearly a total bisimulation between $M_{1} \uplus N_{1}$ and $M_{2} \uplus N_{2}$.

The commutative monoid yields the algebraic preordering $\leq$ on the class of Kripke models with different vocabularies:

$$M \leq N \text{ iff there is a } K \text{ with } M \uplus K \sqsubseteq N.$$
invariance condition restricted to proposition letters in the vocabulary of $M$, and without the Zig condition (see Definition 2.2.2). Formally, given two models $M$ and $N$ such that $P_M \subseteq P_N$, a left-simulation between $M$ and $N$ is a relation $R \subseteq S_M \times S_N$ such that $sRt$ implies that the following hold:

**Restricted Invariance** $V_M(s) = V_N(t) \cap P_M$

**Zag** If for some $i \in I$ there is a $t' \in S_N$ with $t \overset{i}{\rightarrow} t'$ then there is a $s' \in S_M$ with $s \overset{i}{\rightarrow} s'$ and $s'Rt'$.

We will use $M, s \leftarrow \rightarrow N, t$ to indicate that there is a left-simulation that connects $s$ and $t$, and $M \leftarrow \rightarrow N$ to indicate that there is a total left-simulation between $M$ and $N$: there is a left-simulation $R$ that links every world in $N$ to some world in $M$.

Ditmarsch and French [vDF09] prove that for finite static models $M$ and $N$:

$$M \text{ is a simulation of } N \iff \text{there exists an event model } A \text{ st. } N \models M \otimes A$$

Here we prove a similar result in our setting:

**5.2.5. Theorem.** For any models $M, N$ with arbitrary vocabularies:

$$M \leq N \implies M \leftarrow \rightarrow N$$

**Proof** $\Rightarrow$: Assume $M \leq N$. Then there is a model $K$ with $M \otimes K \models N$. Let $Z$ be a total bisimulation between $S_{M \otimes K}$ and $S_N$. Define $R$ as $sRt$ iff there is some world $x \in S_K$ with $(s, x)Zt$. $R$ is easily seen to be a total left-simulation between $M$ and $N$. The restricted invariance property follows from the definition of the valuation on $M \otimes K$. The zag property follows from the definition of the accessibility relations on $M \otimes K$. Thus, $M \models N$.

Note that the converse does not hold without restrictions on the models. For example, let $M$ and $N$ be the following two $S_5$ models:

$M$:

```
M:       p 1 p
         2
         \overline{p}
```

$N$:

```
N:       pq 1 \overline{pq}
         2
         \overline{pq}
```

It is clear that $M \models N$. Now suppose towards a contradiction that there exists an $M'$ such that $M \oplus M' \models N$. Since there is a $pq$ world in $N$, there must be a world $t$ in $M'$ such that $q \in V_M(t)$ and $t$ is compatible with any $p$ world in $M$. Let us denote the upper-right world in $M$ as $s$. Then $(s, t)$ must be in the composed model $M \oplus M'$ and $V((s, t)) = \{p, q\}$. However, according to the definition of $\oplus$, $(s, t)$ cannot reach a

\footnote{Note that the totality here is different from the totality of bisimulation which requires that any world in any one of the two models is linked to some world in the other model.}
5.2. Composing Static Models

∼p world in one step, thus (s, t) is not bisimilar to any pq world in N. It follows that
(s, t) is not bisimilar to any world in N.

Now we look at a subclass of models. A model M is called propositionally di-
ferentiated if for any s, t ∈ M: V_M(s) ≠ V_M(t). For example, the models for Muddy
Children and Russian Cards Problem as in Chapter 3 are indeed propositionally dif-
ferentiated. Restricted to this simple but useful class of models we have the exact
correspondence of ⇐ and ≤:

5.2.6. Theorem. Let M be a propositionally differentiated model. Then

M ≤ N ⇐⇒ M ⊈ N

Proof ⇒ follows from Theorem 5.2.5

For⇐: Assume M ⊈ N. Let R be a left-simulation between M and N. Since R is
total, for each t ∈ S_N there is at least one world s in S_M such that sRt. Note that since
M is propositionally differentiated and P_M ⊆ P_N, for each t ∈ S_N there is at most one
world s in S_M such that V_M(s) = V_N(t) ∩ P_M. Therefore for each t ∈ N there is one
and only one world s such that sRt.

We will show that M ⊆ N ⊈ N. Let the relation Z between M ⊆ N and N be
defined as:

(s, t)Zt′ iff t = t′

We claim Z is a total bisimulation. Totality is straightforward. Suppose (s, t)Zt. We
now check the three conditions for bisimulation.

By the construction of M ⊆ N, V_{M∩N}((s, t)) = V_M(s) ∪ V_N(t) = V_N(t). This proves
the invariance property.

Suppose (s, t) ∼ (s′, t′). By the construction of M ⊆ N this means s ∼ s′ and t ∼ t′.
By the definition of Z, (s′, t′)Zt′. This proves the Zig property.

Suppose t ∼ t′. Recall that since M ⊈ N and M is propositionally differentiated,
there must be a unique s in M such that sRt. Then since R is left-simulation,
there must be some s′ such that s ∼ s′ and s′Rt′. Since s′Rt′, V_M(s′) = V_N(t′) ∩ P_M so
(s′, t′) ∈ M ⊈ N. Since s ∼ s′ and t ∼ t′ then (s, t) ∼ (s′, t′). By the definition of Z,
(s′, t′)Zt′. This proves the Zag property.

H

5.2.2 Expansion

Based on the merging composition operation, we can define the expansions of models
with new vocabularies. Let M^i = (S, P, I, ~, V) with S = P(P), ~ = S × S, V = id. Given M we define the expansion of
M w.r.t. vocabulary P as follows: M + P = M ⊆ M^i. Note that P_{M+P} = P_M ∪ P
and the states in the expansion are of the form (s, X) where s ∈ S_M and X ⊆ P. In the
sequent, we will use variables X, Y to denote (possibly empty) subsets of the whole
vocabulary in the context of expansions.

Here is an example of expanding with a single new proposition letter m2. Note:
Here and henceforth, worlds are i-linked if there is an i-path in the picture.
Model expansions will be used in Section 5.4 to define the event model update on models with arbitrary vocabularies. We now show that expansions w.r.t. different vocabularies are bisimilar to each other, as long as the expanded vocabulary stays the same:

**5.2.7. PROPOSITION.** For any model $M$, and vocabularies $X, Y$ of proposition letters, if $X \cup P_M = Y \cup P_M$ then $M \triangleleft X \Leftrightarrow M \triangleleft Y$.

**PROOF** Let relation $Z \subseteq S_{M \triangleleft X} \times S_{M \triangleleft Y}$ be given by:

$$(s, X') Z (s', Y') \iff s = s' \text{ and } V_M(s) \cup X' = V_M(s') \cup Y'$$

We claim that $Z$ is a total bisimulation. Totality follows from the fact that $X \cup P_M = Y \cup P_M$. Now we check the three conditions of bisimulation. Suppose $(s, X') Z (s', Y')$ then by definition of $Z$, $V_M(s) \cup X' = V_M(s') \cup Y'$, namely the invariance condition holds. Then based on totality, it is easy to show the Zig and Zag conditions also hold. \hfill \qed

Also the expansion is monotonic in the sense that the expansion with a larger extra vocabulary is restricted bisimilar (see Definition 2.2.2) to the expansion with a smaller extra vocabulary:

**5.2.8. PROPOSITION.** For any model $M$, any vocabularies $X, Y$ such that $Y \subseteq X$, if $X \cap P_M = \emptyset$ then $M \triangleleft X \Leftrightarrow M \triangleleft Y$.

**PROOF** Let relation $Z \subseteq S_{M \triangleleft X} \times S_{M \triangleleft Y}$ be given by:

$$(s, X') Z (s', Y') \iff s = s' \text{ and } Y' = X' \cap Y$$

It is not hard to verify that $Z$ is a total bisimulation restricted to the vocabulary $P_M \cup Y$. \hfill \qed

If $M$ is left-similar to $N$ then the expansion of $M$ with $P_N$ is also left-similar to $N$.

**5.2.9. PROPOSITION.** If $M, s \subseteq N, t$ then $M \triangleleft P_N, (s, V_N(t)) \Leftrightarrow N, t$
5.2. Composing Static Models

Let $R$ be a left-simulation which witnesses $\mathcal{M}, s \preceq \mathcal{N}, t$. Let

$$R' = \{(s, V_N(t)) \mid (s, t) \in R\}$$

Note that when $(s, t) \in R$, $(s, V_N(t))$ is indeed in the model $\mathcal{M} \triangleleft \mathcal{P}_N$ due to the restricted invariance condition of $R$. Thus $R'$ is well-defined. Totality follows from the totality of $R$. We claim that $R'$ is a left-simulation between $\mathcal{M} \triangleleft \mathcal{P}_N$, $(s, V_N(t))$ and $\mathcal{N}, t$. The condition of restricted invariance is obvious. For the Zag condition, suppose $t \xrightarrow{i} t' \in \mathcal{N}$ then there is an $s'$ such that $s \xrightarrow{i} s'$ in $\mathcal{M}$ and $s'Rt'$. Since $\mathcal{M} \triangleleft \mathcal{P}_N = \mathcal{M} \triangleright \mathcal{P}_N$, we have $(s, V_N(t)) \xrightarrow{i} (s', V_N(t'))$ in $\mathcal{M} \triangleleft \mathcal{P}_N$ and $(s', V_N(t'))Rt'$.

5.2.3 Preservation

Now let us consider the PDL language over $\mathcal{P}, \mathcal{I}$ (notation: $\text{PDL}_{\mathcal{P}, \mathcal{I}}$):

$$\phi ::= \top | p | \neg \phi | \phi \lor \phi | \langle \pi \rangle \phi$$

$$\pi ::= i | ?\phi | \pi \cdot \pi | \pi + \pi | \pi^*$$

The semantics for $\text{PDL}_{\mathcal{P}, \mathcal{I}}$ is defined as usual (see Section 2.3.1). Note that the truth value of a $\text{PDL}_{\mathcal{P}, \mathcal{I}}$ formula may not be defined on a model with a different vocabulary other than $\mathcal{P}$. We will study a three valued semantics in Chapter 7, while in this chapter, we stick to the 2-valued semantics and make sure the formulas are evaluated on the models where the semantics is defined.

Since PDL is bisimulation invariant, as a straightforward consequence of Proposition 5.2.8, we have:

5.2.10. Proposition. For any model $\mathcal{M}$, if $X \cap P_M = \emptyset$ and $Y \subseteq X$ then for any $\phi \in \text{PDL}_{\mathcal{P}, \mathcal{I}}$:

$$\mathcal{M} \triangleleft X, (s, X') \models \phi \iff \mathcal{M} \triangleleft Y, (s, X' \cap Y) \models \phi.$$  

We will use this proposition to prove Theorem 5.4.5 in Section 5.4.

The diamond fragment of $\text{PDL}_{\mathcal{P}, \mathcal{I}}$ is given by the following $\phi$ form of formulas:

$$\psi ::= \top | p | \neg \psi | \psi \lor \psi$$

$$\pi ::= i | ?\phi | \pi \cdot \pi | \pi + \pi | \pi^*$$

$$\phi ::= \psi | \langle \pi \rangle \phi | \phi \lor \phi | \phi \land \phi.$$  

We can define the box fragment of $\text{L}_{\mathcal{P}, \mathcal{I}}$ i.e. the collection of formulas which are logically equivalent to $\neg \phi$ for some $\phi$ in the diamond fragment.

It is well-known that diamond formulas are preserved under simulation. The following theorem generalizes this to cases where the vocabularies of the two models may be different.

5.2.11. Theorem. If $\mathcal{M}, s \preceq \mathcal{N}, t$ then all formulas $\phi$ in the diamond fragment of $\text{PDL}_{\mathcal{P}, \mathcal{I}}$ are preserved from right to left under left simulation: if $\mathcal{N}, t \models \phi$ then $\mathcal{M}, s \models \phi$. Equivalently, the box fragment of $\text{PDL}_{\mathcal{P}, \mathcal{I}}$ is preserved from left to right under left simulation.
Chapter 5. Composing Models

Proof Suppose $M, s \leq N, t$. From Theorem 5.2.5 we know that $M, s \models N, t$. Let $R$ be a left simulation with $sRt$. We prove the property by induction on the construction of $\phi$. The purely boolean cases are trivial according to the invariance property of $R$. We only need to show the case of $(\pi)\phi$. Suppose $N, t \models (\pi)\phi$, then there is a path $w$ starting from $t$ in $M$ such that $w \in L(\pi)$ which leads to some $\phi$ world $t'$. By the Zag property of $R$, it is not hard to see that there is a matching $w$ path in $M$ starting from $s$ to some world $s'$ such that $s'Rt'$. By the induction hypothesis, $M, s' \not\models \phi$, and therefore $M, s \not\models (\pi)\phi$.

Based on Proposition 5.2.9, we can relax the restrictions of $P_M$ on the formulas:

5.2.12. Corollary. If $M, w \leq N, t$ then all formulas $\phi$ in the diamond fragment of $PDL_{P,1}$ are preserved from right to left under left simulation: if $N, t \models \phi$ then $M \diamond P_N, (w, V(I)) \models \phi$.

Theorem 5.2.11 suggests a way of checking the properties (in terms of formulas in the box fragment) of a big model by looking at its components. The following theorem shows that the components can carry more information about the composed models, if we restrict ourselves to certain decomposition of the models. Formally we say model $M$ is decomposable into $M_0, \ldots, M_n$ if $M \models M_0 \otimes \ldots \otimes M_n$. A pointed model $(M, s)$ is decomposable into $(M_0, s_0), \ldots, (M_n, s_n)$ if $M, s \models M_0 \otimes \ldots \otimes M_n, (s_0, \ldots, s_n)$.

5.2.13. Theorem (Preservation). If a pointed model $(M, s)$ is decomposable into models $(M_0, s_0), \ldots, (M_n, s_n)$ with disjoint vocabularies $P_0, P_1, \ldots, P_n$, then for any $i$: $M, s_i \models P_i$, $M, s$. Therefore for any $\phi$ in $PDL_{P,1}$: $M, s_i \models \phi \iff M, s \models \phi$.

Proof Suppose without loss of generality that $M, s = M_0 \otimes \ldots \otimes M_n, (s_0, \ldots, s_n)$ where $s_i \in M_i$. Given a tuple $\bar{t} = (t_0, t_1, \ldots, t_n) \in M$, we let $[\bar{t}]$ be the $i$th element in the tuple $t$. Let $Z_i$ be the relation on $S_M \times S_M$ given by $Z_i$ if $[\bar{t}] \models t$. We show that $Z_i$ is a $P_i$-restricted bisimulation. It is clear that $sZ_j s_i$. Assume $Z_i t$, for some $\bar{t} \in M$ and $t \in M_i$. Then $V_M(t) \cap P_i = V_M(t)$, by the definition of the merging composition. Thus, $P_i$-restricted invariance holds. Next suppose $\bar{t} \overset{i}{\rightarrow} \bar{t}'$. Then by the definition of the accessibility relations on $M$, $[\bar{t}] \models \bar{t}'$, whence, by definition of $Z_i$, there is a $t''$ in the domain of $M_i$ with $Z_i t''$. It follows that the Zig condition holds. Finally, assume $t \overset{i}{\rightarrow} t''$ in $M_i$ and $Z_i t$. Now consider the state $\bar{t}$ given by $[\bar{t}] = t$ and $[\bar{t}] = [\bar{t}]$ for $j \neq i$. Since $P_i$ is disjoint from any other vocabulary $P_j$ for $j \neq i$, $\bar{t}$ must be in $M$ and $Z_j t''$. Then by the reflexivity of the $S5$ component models and the fact that $t \overset{i}{\rightarrow} t'$, we have $\bar{t} \overset{i}{\rightarrow} \bar{t}'$. This proves the Zig condition.

As an example, consider the case of Muddy Children. From the above theorem, we know that any epistemic statement that talks about the muddiness of a single child in the big model can be checked in a two-world component, e.g., at the component model $m_1 \overset{1}{\rightarrow} m_2$, we can verify that agent 2 knows that agent 1 does not know whether she herself is muddy.
5.3 Decomposition

At this stage a natural question to ask is: what kind of model can be decomposed into what kind of form? In this section we look at a particular class of models which is useful in multi-agent systems. In the interpreted systems literature, a basic proposition \( p \in P \) is \( i \)-local for \( i \in I \) in a model \( M \), if for any \( s, t \) in \( S_M \): \( s \sim_i t \) implies that \( (p \in V_M(s) \iff p \in V_M(t)) \) (cf., e.g., [EvdMM98]). Intuitively, the \( i \)-local propositions are the atomic observables of agent \( i \) and thus agent \( i \) also knows whether they are true. Here we extend this idea by considering not only basic propositions but also their boolean combinations. We say \( M \) is locally generated if, for every agent \( i \), there is a non-empty set of boolean formulas \( \Phi_i \) (the set of local observables) based on \( P_M \) such that:

\[
\text{for all } s, s' \in S_M, s \sim_i s' \text{ iff for all } \phi \in \Phi_i, M, s \models \phi \iff M, s' \models \phi
\]

Intuitively, a model is locally generated if those local observables determine the epistemic relations in the model. The Muddy Children model is a typical example of a locally generated model (the set of local observables for \( i \) is \( \{m_j \mid j \neq i, j \in I\} \)). As the following two propositions will show, locally generated models are essentially propositionally differentiated models, which we considered in Theorem 5.2.6.

5.3.1. Proposition. A locally generated model is bisimilar to a propositionally differentiated model. More precisely, its bisimulation contraction (see Definition 2.2.3) is propositionally differentiated.

Proof. Given a locally generated model \( M \), suppose \( \Phi_i \) is the set of local observables for \( i \). Let \( Z = \{(s, t) \mid V_M(s) = V_M(t)\} \). We show \( Z \) is a bisimulation. Assume \( sZt \). The invariance condition is trivial. For Zig, suppose \( s \sim_i s' \). Since \( M \) is locally generated, for any \( \phi \in \Phi_i : M, s \models \phi \iff M, s' \models \phi \). Since \( \Phi_i \) contains only boolean formulas and \( V_M(s) = V_M(t) \), we have for any \( \phi \in \Phi_i : M, t \models \phi \iff M, s' \models \phi \). Again due to the definition of the relations in a locally generated model, we have \( t \sim_i s' \). Obviously \( s'Zs' \), thus it proves the Zig condition. The same argument works for the Zag condition. Therefore it is easy to see that the bisimulation contraction of \( M \) is propositionally differentiated.

We also have:

5.3.2. Proposition. Propositionally differentiated models are locally generated.

Proof. Suppose \( M \) is propositionally differentiated. Let \( |S_M| \) be the partitioning of \( S_M \) according to the equivalence relation \( \sim \). Since \( M \) is propositionally differentiated, we can characterize each world by a conjunction of literals. Then we can characterize each equivalence class in \( |S_M| \) by a disjunction of these characterizations. Let \( \Phi_i \) be the set of these disjunctions. Then clearly \( M \) is locally generated from these \( \Phi_i \).

We can decompose a locally generated model into certain components in an intuitive way.
5.3.3. Theorem (Decomposition by agents). Given a set of agent \( I = \{1,2,\ldots,n\} \). If \( M = (S,P,I,\sim,V) \) is locally generated w.r.t. \( \Phi_1,\ldots,\Phi_n \), then there are models \( M_1,\ldots,M_n \) and \( M_0 \) such that:

- \( M \equiv (M_0 \oplus M_1 \oplus \cdots \oplus M_n) \);
- \( |S_{M_1}| \leq |S| \) and \( M_i \) is bisimulation contracted model;
- \( P_{M_i} = \{ p \in P_M \mid p \text{ appears in } \Phi_i \} \) for \( j > 0 \);

Proof. Let \( P_i = \{ p \in P_M \mid p \text{ appears in } \Phi_i \} \) and \( P_0 = P_M \) then we define \( N_i = (S_M,P_i,I,\sim_{N_i},V_i) \) where \( V_i \) is the restriction of \( V_M \) to \( P_i \) and

\[
s \sim_{N_i} s' \iff \begin{cases} s \sim_{M} s' & \text{if } j = i \\ \text{always if } i \neq j \end{cases}
\]

Intuitively, for each \( i \in I : N_i \) is a “local” model for agent i obtained by ignoring the non-local information: atomic propositions not mentioned in the i-observables and epistemic accessibility relations for agents other than \( i \). Note that by ignoring the epistemic relations for \( j \) we mean setting \( \sim_j \) to be universal. For example:

\[
\begin{array}{c|c|c}
m_1m_2 & 1 & m_1m_2 \\
\hline 2 & m_1 & m_2 \\
m_1m_2 & 1 & m_2
\end{array}
\quad
\begin{array}{c|c|c}
m_2 & 1,2 & m_2 \\
\hline 2 & m_2 & m_2 \\
m_2 & 1,2 & m_2
\end{array}
\]

By our definition, the relations in \( N_0 \) are universal. Intuitively, \( N_i \) captures all the possible states of affairs in \( M \). Let a relation \( Z \subseteq S_M \times S_{N_0 \oplus \cdots \oplus N_n} \) be given as follows:

\[
sZ(s_0,s_1,\ldots,s_n) \iff s = s_0
\]

Now let us verify that \( Z \) is indeed a total bisimulation. Totality and invariance are trivial by definition of \( Z \).

For Zig: Suppose \( s \sim_{M} s' \) and \( sZ(s_0,s_1,\ldots,s_n) \) then \( s = s_0 \). Since \( s,s_1,\ldots,s_n \) exists, then \( V_M(s) \cap P_i = V_{N_i}(s_i) \). Therefore \( s \) and \( s_i \) satisfy the same set of boolean formulas based on \( P_i \). Since \( M \) is locally generated then we know that \( s' \) and \( s_i \) agree on the formulas in \( \Phi \). Therefore \( s_i \) and \( s' \) must also agree on the truth values of the formulas in \( \Phi \) thus \( s \sim_{M} s' \). Since \( \sim_{N_i} \) is universal for \( j \neq i \), it is clear that \( (s,s_1,\ldots,s_n) \sim_{N_i} (s',s',\ldots,s') \) in \( N_0 \oplus \cdots \oplus N_n \) and \( sZ(s',s',\ldots,s') \).

For Zag: Suppose \( sZ(s_0,s_1,\ldots,s_n) \) then \( sZ(s_0',s_1',\ldots,s_n') \) and \( sZ(s_0,s_1,\ldots,s_n) \), we then have \( s = s_0 \) and \( s_i \sim_{N_i} s_i' \). By the definition of \( N_i \), we have \( s_i \sim_{M} s_i' \). Thus \( s_i \) and \( s_i' \) agree on formulas in \( \Phi_i \). Since \( s_0 \) in \( M \) is compatible with \( s_i \) in \( N_i \) and \( s_i' \) in \( N_i \) is
5.3. Decomposition

compatible with $s'_0$ in $M$ thus $s'_0$ also agrees with $s_0$ on formulas in $\Phi_i$. Therefore $s \sim_M s'_0$ and $s'_0 Z(s_0', s_1', \ldots, s'_n)$. This proves the Zag condition.

Now we have shown $M \equiv N_0 \oplus \cdots \oplus N_n$. Let $M_i$ be the bisimulation contraction of $N_i$. From Proposition 5.2.4 we know $M \equiv (M_0 \oplus M_1 \oplus \cdots \oplus M_n)$. This proves the Zag condition.

In the above proof, $M_0$ is used to rule out unnecessary worlds created by the merging composition. It is not hard to see that if $\bigcup_{i \in I} P_i = P_M$ and for any $P \subseteq P_M$ there is an $s \in S_M$ such that $V_M(s) = P$, then we can drop the $M_0$ in the decomposition.

The above theorem gives another way to decompose the Muddy Children models different from the one in Example 5.2.2. Recall that an $n$-Muddy Children model is locally generated by sets of observables $\Phi_1, \ldots, \Phi_n$ where $\Phi_i = \{m_j \mid j \neq i, j \in I\}$. For example, if $n = 3$ then the set of observables for agent 1 is $\{m_2, m_3\}$. We can then decompose the 3-Muddy Children model $M$ by $M_1, \ldots, M_3$, where e.g., $M_1$ is as follows:

\[
\begin{array}{c|c|c}
m_2 & m_3 & 2,3 \\
m_2 & m_3 & 2,3 \\
\end{array}
\]

Compared to the two-world model decomposition in Example 5.2.2 the above decomposition requires bigger size components ($2^{n-1}$ worlds for the $n$ children case). This is because we decompose the model in an *agent-based* fashion: each component represents one agent’s observational power regardless of the others. Thus if the vocabulary of the set of observables $\Phi_i$ is big then so is the component model. In the Muddy Children example, if there are more children then the vocabulary of the observables for each child also increases (e.g., new $m_i$), therefore the component model for this agent also grows bigger. However, in other applications the vocabulary of observables may not increase even when the initial model grows bigger. For example, in the Russian Cards scenario, the agents can only observe their own cards, no matter how many other agents there are (cf. the locally generated initial model of $RCP_{k,n}$ in Section 3.4). Therefore, the size of each component can be constant and relatively small.

To decompose a Muddy Children model as in Example 5.2.2, we decompose the model in an *issue-based* fashion (every proposition is an issue), as the following theorem demonstrates:

**Theorem (Decomposition by Issues).** Given a set of agent $I = \{1, 2, \ldots, n\}$ and a set of proposition letters $P = \{p_1, \ldots, p_k\}$, if $M = (S, P, I, \sim, V)$ is locally generated by $\Phi_1, \ldots, \Phi_n$ such that $\Phi_i$ only contains atomic propositions (i.e., $\Phi_i \subseteq P$), then there are models $M_1, \ldots, M_n$ and $M_0$ such that:

- $M \equiv (M_0 \oplus M_1 \oplus \cdots \oplus M_n)$;
- $P_{M_j} = \{p_j\}$ for $j > 0$ and $P_0 = P$.
Chapter 5. Composing Models

- \(|S_M| = 2\) for \(j > 0\)

**Proof.** Let \(M_0\) be the same as in the proof of Theorem 5.3.3. For \(j > 0\), let \(M_j = (S_j, P_j, L \sim M_j, V_j)\) where:

- \(S_j = \{p_j, \overline{p_j}\}\);
- \(P_j = \{p_j\}\);
- \(V_j(p_j) = \{p_j\}\) and \(V_j(\overline{p_j}) = \emptyset\) with the obvious interpretation;
- for \(j > 0\): \(sZ i \sim M_j\) \(t \iff s = t\) or \((s \neq t\) and \(p_j \notin \Phi_i\)).

Let a relation \(Z \subseteq S_M \times S_{M_0 \oplus \ldots \oplus M_k}\) be given as follows:

\[
sZ(s_0, s_1, \ldots, s_k) \iff s = s_0
\]

If \(sZ(s_0, s_1, \ldots, s_k)\) then for \(0 < j \leq k : s_j\) intuitively represents the truth value of \(p_j\) in \(s\). Now let us verify that \(Z\) is indeed a total bisimulation. Totality and invariance are trivial.

For Zig: Suppose \(s \sim_M s'\) and \(sZ(s_0, s_1, \ldots, s_k)\). Since \((s_0, s_1, \ldots, s_k)\) exists, then \(s_0 = s\) and \((s_j = p_j \iff p_j \in V_M(s))\). It is easy to see there is some world \((s', s'_1, \ldots, s'_k)\) in \(M_0 \oplus \ldots \oplus M_k\) such that \(s'Z(s'_0, s'_1, \ldots, s'_k)\). We need to show that for each \(j > 0\) :

\[
s_j \sim_M s'_j.
\]

If \(s_j = s'_j\) then \(s_j \sim_M s'_j\) by definition of \(\sim_M\). Now suppose \(s_j \neq s'_j\). Since \(M\) is locally generated by \(\Phi_1, \ldots, \Phi_n\), \(s\) and \(s'\) agree on the truth values of the atomic propositions in \(\Phi_i\). Therefore \(p_j \notin \Phi_i\), thus \(s_j \sim_M s'_j\).

For Zag: Suppose \((s_0, s_1, \ldots, s_k) \sim_M (s'_0, s'_1, \ldots, s'_k)\) and \(sZ(s_0, s_1, \ldots, s_k)\), we have \(s_0 = s\) and for any \(j \leq k : s_j \sim_M s'_j\). By the definition of \(M_n\), we have \(s_j = s'_j\) or \(s_j \neq s'_j\) and \(p_j \notin \Phi_i\). Namely, for \(p_j \in \Phi_i : s_j = s'_j\). Therefore \(s\) and \(s'\) agree on the truth values of the propositions in \(\Phi_i\). Since \(M\) is locally generated by \(\Phi_1, \ldots, \Phi_n\) then \(s \sim_M s'\). This proves the Zag condition.

According to the above theorem, a locally generated model by sets of atomic propositions can be decomposed by components based on each atomic proposition. This gives us the desired decomposition of the \(n\)-Muddy Children models as in Example 5.2.2. Theorems 5.3.3 and 5.3.4 show that we can decompose a locally generated model. On the other hand, there are models which are not locally generated but decomposable in a non-trivial way. For example, consider the following model (to ease the presentation, we use solid lines for agent 1 and dotted lines for agent 2):
This model is not bisimilar to any propositionally differentiated model. From Proposition 5.3.1 follows that it is not bisimilar to any locally-generated model. Nevertheless, $M$ can be decomposed into two models as follows:

If we take the boldface states as the real worlds in these two models respectively, then the two models capture the situations where agent 2 is not sure whether 1 knows $m_1$ and agent 1 is not sure whether 2 knows $m_2$. If we interpret $m_1$ and $m_2$ as in Muddy Children, then the composed model, when taking the top-left corner state as the real world, captures the situation where the children can see each other’s faces but are not sure whether the other has a mirror (actually they do have mirrors). Since the vocabularies of the above two models are disjoint, from Proposition 5.2.13 we know that any true claim about only $m_2$ or $m_1$ will be preserved at the components. For example, agent 1 knows agent 2 does not know whether agent 1 knows $m_1$ can be verified in the left hand component model.

5.4 Composing Updates

Recall Definition 2.3.2 that an (SS) event model $\mathcal{M} = (E, I, \leftrightarrow, Pre)$ is like a static model, but with valuations replaced by precondition formulas taken from an appropriate language. Let $P_{\mathcal{M}}$ be the set of proposition letters appearing in the preconditions of $e \in E$ according to $Pre$. 

![Diagram of models and states]
Note that the standard product update as in Definition 2.3.3 is defined on the pairs of a static model and an event model where $P_A \subseteq P_M$. We will now generalize the standard product update to an operation that works on Kripke models and event models with arbitrary vocabularies.

Model expansion is used in the following definition of product update to ensure that no matter what the vocabulary of the static model is, we can always check the preconditions of the events model on the static model. The vocabulary of the resulting updated model is the union of the vocabulary of the static model and the vocabulary of the event model.

5.4.1. Definition. (Extended Product Update) Given a static model $M = (S, P, I, \sim, V)$ and an event model $A = (E, I, \leftrightarrow, Pre)$ for the same set of agents $I$. Let $X$ be the differential vocabulary, i.e., $X = P_A - P$. Then the extended product update $M \bowtie A$ is the static model $(S', P \cup P_A, I, \sim', V')$ given by $(M \bowtie X) \otimes A$, where $\otimes$ denotes the usual update product.

This definition boils down to the following:

1. $S' = \{(s, X', e) \mid s \in S, e \in E, X' \subseteq X, M \bowtie X(s, X') \models Pre(e)\}$,
2. $(s, X', e) \sim'(t, X'', f)$ iff $s \sim t$ and $e \leftrightarrow f$,
3. $V'(s, X', e) = V(s) \cup X'$.

Note that the definition of accessibility relations does not require $X = X'$ since all the different values for the novel atoms are the same for $i$. From Proposition 5.2.8, we can equivalently (modulo bisimulation) define the update as $(M \bowtie P_A) \otimes A$. However, for the ease in proofs we will stick to the above definition where $M$ is expanded with $P_A - P_M$. Here is an example of an update with a public announcement "at least one of you is muddy" (i.e., an event model with only one world whose precondition is $m_1 \lor m_2$). As usual, we denote this event model as $! (m_1 \lor m_2)$. Note that the update involves model expansion:

Here is an update of the other component of the 2-Muddy Children model:

And here is the outcome of composing the two update results:
5.4. Composing Updates

This is the same as the result of public announcement of \( m_1 \lor m_2 \) on the composition of \( m_1 \xrightarrow{1} \overline{m}_1 \) and \( m_2 \xrightarrow{1} \overline{m}_2 \). In the following, we will show that this outcome is not accidental: updating a composed model yields the same result (modulo bisimulation) as composing the updates of its components, provided the event model has certain properties. We call an event model \( \mathcal{A} \) propositionally differentiated if the preconditions are purely boolean formulas and any two states in \( \mathcal{A} \) have mutually exclusive preconditions. For a boolean precondition \( \text{Pre}(e) \) of \( e \) in \( \mathcal{A} \), a vocabulary \( \mathcal{P} \) such that \( \mathcal{P}_\mathcal{A} \subseteq \mathcal{P} \), and a set \( X \subseteq \mathcal{P} \), we write \( X \models_{\mathcal{P}} \text{Pre}(e) \) if \( X \) (viewed as a valuation for \( \mathcal{P} \)) makes \( \text{Pre}(e) \) true. It is clear that \( X \cap \mathcal{P}_\mathcal{A} \models_{\mathcal{P}} \text{Pre}(e) \) iff \( X \models_{\mathcal{P}} \text{Pre}(e) \). In case \( \mathcal{P} = \bigcup_{i \in \mathbb{N}} \mathcal{P}_M \) we write \( X \models_{\mathcal{P}_{\{M,i\in\mathbb{N}\}}} \text{Pre}(e) \).

5.4.2. Theorem. If \( \mathcal{A} \) is a propositionally differentiated event model then:

\[(\mathcal{M} \oplus \mathcal{N}) \otimes \mathcal{A} \rightleftharpoons (\mathcal{M} \otimes \mathcal{A}) \oplus (\mathcal{N} \otimes \mathcal{A})\]

**Proof.** Let \( \mathcal{M}_1 = (\mathcal{M} \oplus \mathcal{N}) \otimes \mathcal{A} \) and \( \mathcal{M}_2 = (\mathcal{M} \otimes \mathcal{A}) \oplus (\mathcal{N} \otimes \mathcal{A}) \). Let relation \( Z \subseteq S_{\mathcal{M}_1} \times S_{\mathcal{M}_1} \) be given by:

\[((s, t, X), e) Z ((s', X, e'), (t', X, e'')) \text{ iff } s = s', t = t', e = e' = e'' \text{ and } X = X_1 \cap X_2\]

We need to show that \( Z \) is a total bisimulation between \( \mathcal{M}_1 \) and \( \mathcal{M}_2 \). The totality of \( Z \) is proved in Lemma 5.4.3. Here we focus on the three conditions of bisimulation. Suppose \( ((s, t, X), e) \) and \( ((s, X, e'), (t, X, e'')) \) exist in \( \mathcal{M}_1 \) and \( \mathcal{M}_2 \) respectively and \( ((s, t, X), e) Z ((s, X, e'), (t, X, e'')) \).

For invariance we need to show

\[V_{\mathcal{M}_1}(s) \cup V_{\mathcal{N}}(t) \cup (X_1 \cap X_2) = V_{\mathcal{M}_1}(s) \cup X_1 \cup V_{\mathcal{N}}(t) \cup X_2\]

Since the only difference between the left hand side and right hand side is about \( X_1, X_2 \subseteq \mathcal{P}_{\mathcal{A}}, \) then showing the following suffices:

\[(V_{\mathcal{M}_1}(s) \cup V_{\mathcal{N}}(t) \cup (X_2 \cap X_2)) \cap \mathcal{P}_{\mathcal{A}} = (V_{\mathcal{M}_1}(s) \cup X_1 \cup V_{\mathcal{N}}(t) \cup X_2) \cap \mathcal{P}_{\mathcal{A}} \quad (\star)\]

Since \( (s, X_1, e) \) in \( \mathcal{M} \otimes \mathcal{A} \) and \( (t, X_2, e) \) in \( \mathcal{N} \otimes \mathcal{A} \) are compatible, we have:

\[(V_{\mathcal{M}_1}(s) \cup X_1) \cap \mathcal{P}_{\mathcal{A}} = (V_{\mathcal{N}}(t) \cup X_2) \cap \mathcal{P}_{\mathcal{A}} \quad (\star\star)\]
Now let us massage the left hand side of (⋆):

\[
(V_M(s) \cup V_N(t) \cup (X_1 \cap X_2)) \cap P_A
\]

\[
= ((V_M(s) \cup V_N(t) \cup X_1) \cap (V_M(s) \cup V_N(t) \cup X_2)) \cap P_A
\]

\[
= ((V_M(s) \cup V_N(t) \cup X_1) \cap P_A \cap (V_M(s) \cup V_N(t) \cup X_2)) \cap P_A
\]

\[
= (((V_M(s) \cup X_1) \cap P_A) \cup (V_N(t) \cap P_A)) \cap (((V_N(t) \cup X_2) \cap P_A) \cup (V_M(s) \cap P_A))
\]

\[
= (((V_M(s) \cup X_1) \cap P_A) \cup (V_N(t) \cap P_A)) \cap ((V_M(s) \cup X_1) \cap P_A) \cup (V_M(s) \cap P_A) \quad \text{(by (⋆))}
\]

\[
= ((V_M(s) \cup X_1) \cap P_A) \cup (V_N(t) \cap V_M(s) \cap P_A)
\]

\[
= ((V_M(s) \cup X_1) \cap P_A) \quad \text{(since } V_N(t) \cap V_M(s) \subseteq V_M(s))
\]

\[
= (V_M(s) \cup X_1) \cap P_A \cup (V_N(t) \cup X_2) \cap P_A \quad \text{(from (⋆))}
\]

\[
= (V_M(s) \cup X_1 \cup V_N(t) \cup X_2) \cap P_A
\]

This proves the invariance requirement.

Now assume \((s, t, X, e)Z((s, X_1, e_1), (t, X_2, e_2))\) and \((s, t, X, e) \overset{\imath}{\rightarrow} ((s', t', X'), e')\) in \(M_1\), then \(e = e_1 = e_2, s \overset{\imath}{\rightarrow} s'\) in \(M_1\), \(t \overset{\imath}{\rightarrow} t'\) in \(N\) and \(e \overset{\imath}{\rightarrow} e'\) in \(A\). From totality (Lemma 5.4.3) in \(M_2\) there exists \(((s', X_1', e'), (t', X_2', e'))\) for some \(X_1'\) and \(X_2'\) such that

\[
((s', t', X'), e')Z((s', X_1', e'), (t', X_2', e')).
\]

According to the definition of relations in \(M_2\), it is not hard to see that

\[
((s, X_1, e), (t, X_2, e)) \overset{\imath}{\rightarrow} ((s', X_1', e'), (t', X_2', e'))
\]

This proves the Zig requirement.

Suppose \((s, t, X, e)Z((s, X_1, e_1), (t, X_2, e_2))\) and

\[
((s, X_1, e), (t, X_2, e)) \overset{\imath}{\rightarrow} ((s', X_1', e'), (t', X_2', e'))
\]

Therefore \(s \overset{\imath}{\rightarrow}_{M_1} s', t \overset{\imath}{\rightarrow}_{M_2} t', e \overset{\imath}{\rightarrow}_{A} e'.\) From Lemma 5.4.3 in \(M_1\) there exists \((s', X', e')\) for some \(X'\) such that:

\[
((s', t', X'), e')Z((s', X_1', e'), (t', X_2', e'))
\]

It follows that \((s, t, X, e) \overset{\imath}{\rightarrow} ((s', t', X'), e').\) This proves the Zag condition.

\[ \Box \]

5.4.3. Lemma. The relation \(Z\) defined above is total.

Proof We need to show for any state \(u\) that exists in \(M_1\) there is a state \(v\) that exists in \(M_2\) such that \(uZv\), and for any \(v\) that exists in \(M_2\) there is an \(u\) in \(M_1\) such that \(uZv.\)

Suppose \((s, t, X, e)\) exists in \(M_1\) then the following hold:

Fact 1 \(X \subseteq P_A - (P_M \cup P_N);\)

Fact 2 \(V_M(s) \cap P_N = V_N(t) \cap P_M'.\)
Fact 3 \( V_M(s) \cup V_N(t) \cup X \vDash_{M,N,\mathcal{A}} Pre(e). \)

Now we let:
\[
X_1 = X \cup ((V_N(t) - V_M(s)) \cap P_{\mathcal{A}}) \text{ and } X_2 = X \cup ((V_M(s) - V_N(t)) \cap P_{\mathcal{A}}).
\]
Clearly \( X = X_1 \cap X_2. \) To show \( ((s, X_1, e), (t, X_2, e)) \) exists in \( M_2, \) we need to show:

1. \( X_1 \) and \( X_2 \) are well-defined: \( X_1 \subseteq P_{\mathcal{A}} - P_M \) and \( X_2 \subseteq P_{\mathcal{A}} - P_N. \)

2. \( e \) can be executed on both \( (s, X_1) \) and \( (t, X_2): \) \( V_M(s) \cup X_1 \vDash_{M,N,\mathcal{A}} Pre(e) \) and \( V_N(t) \cup X_2 \vDash_{M,N,\mathcal{A}} Pre(e). \)

3. \( (s, X_1, e) \) and \( (t, X_2, e) \) can be composed: \( (V_M(s) \cup X_1) \cap (P_N \cup P_{\mathcal{A}}) = (V_N(t) \cup X_2) \cap (P_M \cup P_{\mathcal{A}}). \)

For (1): Recall that \( V_M(s) \cap P_N = V_N(t) \cap P_M \), thus we have \( V_N(t) \cap P_M \subseteq V_M(s) \) and \( V_M(s) \cap P_N \subseteq V_N(t) \). Therefore, \( (V_N(t) - V_M(s)) \cap P_{\mathcal{A}} = \emptyset \) and \( (V_M(s) - V_N(t)) \cap P_{\mathcal{A}} = \emptyset. \) It means \( ((V_N(t) - V_M(s)) \cap P_{\mathcal{A}}) \subseteq P_{\mathcal{A}} - P_M \) and \( (V_M(s) - V_N(t)) \cap P_{\mathcal{A}}) \subseteq P_{\mathcal{A}} - P_N. \) Also note that \( X \subseteq P_{\mathcal{A}} \cap (P_M \cap P_N) \), and similarly \( X \subseteq P_{\mathcal{A}} - P_N. \) Therefore by the definitions of \( X_1 \) and \( X_2 \) we have \( X_1 \subseteq P_{\mathcal{A}} - P_M \) and \( X_2 \subseteq P_{\mathcal{A}} - P_N. \)

For (2): By the definition of \( X_1: \)
\[
V_M(s) \cup X_1 = V_M(s) \cup X \cup ((V_N(t) - V_M(s)) \cap P_{\mathcal{A}})
\]
Then we have:
\[
(V_M(s) \cup X_1) \cap P_{\mathcal{A}} \quad \# \\
= (V_M(s) \cup X \cup ((V_N(t) - V_M(s)) \cap P_{\mathcal{A}})) \cap P_{\mathcal{A}} \\
= ((V_M(s) \cup X) \cap P_{\mathcal{A}}) \cup ((V_N(t) - V_M(s)) \cap P_{\mathcal{A}}) \\
= (V_M(s) \cup V_N(t) \cup X) \cap P_{\mathcal{A}}
\]
Since \( V_M(s) \cup V_N(t) \cup X \vDash_{M,N,\mathcal{A}} Pre(e) \) we have
\[
(V_M(s) \cup V_N(t) \cup X) \cap P_{\mathcal{A}} \vDash_{M,N,\mathcal{A}} Pre(e)
\]
Therefore from the derivation \( \# \), \( (V_M(s) \cup X_1) \cap P_{\mathcal{A}} \vDash_{M,N,\mathcal{A}} Pre(e) \) and then \( V_M(s) \cup X_1 \vDash_{M,N,\mathcal{A}} Pre(e) \). Similarly we can prove \( V_N(t) \cup X_2 \vDash_{M,N,\mathcal{A}} Pre(e) \).

For (3): By the definition of \( X_1: \)
\[
(V_M(s) \cup X_1) \cap (P_N \cup P_{\mathcal{A}}) = ((V_M(s) \cup X_1) \cap P_{\mathcal{A}}) \cup ((V_M(s) \cup X_1) \cap P_N)
\]
From \( \# \), we know that: \( (V_M(s) \cup X_1) \cap P_{\mathcal{A}} = (V_M(s) \cup V_N(t) \cup X) \cap P_{\mathcal{A}} \), thus
\[
(V_M(s) \cup X_1) \cap (P_N \cup P_{\mathcal{A}}) = ((V_M(s) \cup V_N(t) \cup X) \cap P_{\mathcal{A}}) \cup (V_M(s) \cap P_N) \cup (X_1 \cap P_N) \quad \dagger
\]
Note that
\[
X_1 \cap P_N \\
= (X \cup ((V_N(t) - V_M(s)) \cap P_{\mathcal{A}})) \cap P_N \\
= ((V_N(t) - V_M(s)) \cap P_{\mathcal{A}}) \cap P_N \quad \text{ (since } X \cap P_N = \emptyset) \\
= (V_N(t) - V_M(s)) \cap P_{\mathcal{A}} \quad \text{ (since } V_N(t) \subseteq P_N)
\]
Therefore go back to (‡) we have:

\[
(V_M(s) \cup X_1) \cap (P_N \cup P_A) \\
= ((V_M(s) \cup V_N(t) \cup X) \cap P_A) \cup (V_M(s) \cap P_N) \cup ((V_N(t) - V_M(s)) \cap P_A) \\
= ((V_M(s) \cup V_N(t) \cup X) \cap P_A) \cup (V_M(s) \cap P_N) \\
\]

(‡)

Similarly we can show

\[
(V_N(t) \cup X_2) \cap (P_M \cup P_A) = ((V_M(s) \cup V_N(t) \cup X) \cap P_A) \cup (V_N(t) \cap P_M) \\
\]

(§)

From the Fact 1 \((V_N(t) \cap P_M = V_M(s) \cap P_N)\), (‡) and (§) we have:

\[
(V_M(s) \cup X_1) \cap (P_N \cup P_A) = (V_N(t) \cup X_2) \cap (P_M \cup P_A) \\
\]

This proves (3).

Till now we have proved that for any state \(u\) that exists in \(M_1\) there is a state \(v\) exists in \(M_2\) such that \(uZv\). Now suppose \((s, X_1, e), (t, X_2, e')\) exists in \(M_2\) we need to show that there is an \(u\) in \(M_1\) such that \(uZv\). Since \(A\) is propositionally differentiated, no two actions can be executed under the same valuation over \(P_A\), thus events \(e = e'\). We now only need to show that \((s, t, X_1 \cap X_2, e)\) exists in \(M_1\). Formally we need to verify the following claims:

1. \(s\) and \(t\) are compatible.
2. \(X\) is well-defined: \(X_1 \cap X_2 \subseteq (P_M \cup P_N)\).
3. \(e\) can be executed on \((s, t, X_1 \cap X_2)\): 

\[
V_M(s) \cup V_N(t) \cup (X_1 \cap X_2) \models_{M,N,A} \text{Pre}(e). \\
\]

From the existence of \((s, X_1, e), (t, X_2, e')\), clearly \(s\) and \(t\) can be composed. Since \(X_1 \subseteq P_A - P_M\) and \(X_2 \subseteq P_A - P_N\), (2) is also straightforward. Now we prove (3). Since \((s, X_1, e)\) and \((t, X_2, e')\) can be composed

\[
V_M(s) \cup X_1 \cup V_N(t) = V_M(s) \cup X_2 \cup V_N(t) \\
\]

Now we have:

\[
V_M(s) \cup X_1 \cup V_N(t) \\
= (V_M(s) \cup X_1 \cup V_N(t)) \cap (V_M(s) \cup X_2 \cup V_N(t)) = (V_M(s) \cup V_N(t) \cup (X_1 \cap X_2)) \\
\]

Since \((s, X_1, e), (t, X_2, e')\) exists, it is not hard to see that \((V(s) \cup V(t) \cup X_1) \models_{M,N,A} \text{Pre}(e)\). Thus \(V_M(s) \cup V_N(t) \cup (X_1 \cap X_2) \models_{M,N,A} \text{Pre}(e)\). \(\blacksquare\)

Event models are very similar to static models, and it turns out that composition on event models can be defined in a natural way.

### 5.4.4. Definition. (Merging Composition of Event Models)

The composition \(\mathcal{A} \oplus \mathcal{B}\) of two event models \(\mathcal{A}\) and \(\mathcal{B}\) with the same set of agents \(I\) is given by \((E, I, \leftrightarrow, \text{Pre})\), where:
5.4. Composing Updates

- \( E = \{(e, f) \mid e \in E_A, f \in E_B\} \)
- \( (e, f) \leftrightarrow (e', f') \) iff \( e \leftrightarrow e' \) in \( A \) and \( f \leftrightarrow f' \) in \( B \)
- \( \text{Pre}((e, e')) = \text{Pre}_A(e) \land \text{Pre}_B(e') \).

Note that in the composed event model, some \( e \) may have an unsatisfiable precondition. We do not delete such non-executable actions in the composed model. The simplest example is composing two announcements \( !\phi \) and \( !\psi \), which results in an announcement of \( \phi \land \psi \). The composition operator presented here can be viewed as a kind of parallel compositions of events. Consider the following example (where \( I = \{1, 2\} \) and the propositions are preconditions):

\[
\begin{align*}
p & \quad \bar{q} & \quad \bar{p} & \quad p\bar{q} & \quad \bar{p}q \\
2 & \oplus & 2 & \text{pruned} & 2 \\
q & \quad q\bar{q} & \quad \bar{p}q & \quad \bar{p}q
\end{align*}
\]

The first model captures the event that agent 1 is being told that either \( p \) or \( q \) is true, while agent 2 can only see it without hearing the exact message. Similarly, the second model reflects the event that 2 is being told either \( p \) or \( q \) is false without 1 hearing the message. The composition of the two captures that both events are happening at the same time. As we can see, the effect of updating this composed event is the same as updating an announcement \( p \land \neg q \) or \( \neg p \land q \). Intuitively, if agent 1 is told \( p \) and he knows that 2 is (truthfully) told either \( \neg q \) or \( \neg p \) then he actually knows that \( \neg p \).

Updating with a composite event model should yield the same outcome as first updating with its components and then composing the results. The following theorem says that it does, notably, without any restriction to certain class of models.

5.4.5. Theorem. \( M \circledast (A \oplus B) \succeq (M \circledast A) \oplus (M \circledast B) \).

Proof: Let \( M_1 = M \circledast (A \oplus B) \) and \( M_2 = (M \circledast A) \oplus (M \circledast B) \). Let the relation \( Z \subseteq S_{M_1} \times S_{M_2} \) be given by:

\( (s, X, (e, f))Z((s', X_1, e'), (s'', X_2, f')) \) iff \( s = s' = s'', e = e', f = f' \) and \( X = X_1 \cup X_2 \)

We first show \( Z \) is total.

\( \Rightarrow \): Suppose that \( (s, X, (e, f)) \) is in \( S_{M_1} \), we need to show that there are \( X_1 \) and \( X_2 \) with \( X = X_1 \cup X_2 \) such that \( ((s, X_1, e), (s, X_2, f)) \) exists in \( S_{M_2} \). Notice that:

\[
\begin{align*}
(s, X, (e, f)) & \in S_{M_1} \\
\iff M \models \langle (P_A \cup P_B) - P_M \rangle, (s, X) \models P_{\text{Pre}}(e) \land P_{\text{Pre}}(f) \\
\iff M \models \langle (P_A - P_M) \cup (P_B - P_M) \rangle, (s, X \cap (P_A - P_M)) \models P_{\text{Pre}}(e) \\
& \quad \text{and } M \models \langle (P_B - P_M) \cup (P_A - P_M) \rangle, (s, X \cap (P_B - P_M)) \models P_{\text{Pre}}(f) \quad \text{(From Proposition 5.2.10)}
\end{align*}
\]
80

Chapter 5. Composing Models

Now let $X_1 = X \cap (P_A - P_M)$ and $X_2 = X \cap (P_B - P_M)$. Since $X \subseteq (P_A \cup P_B) - P_M$, $X_1 \cup X_2 = X \cap ((P_A - P_M) \cup (P_B - P_M)) = X$. From the above derivation, $(s, X_1, e) \in S_{M, A}$ and $(s, X_2, f) \in S_{M, B}$ exist and they are compatible. Therefore $((s, X_1, e), (s, X_2, f)) \in S_M$.

$\Leftarrow$: Suppose $((s, X_1, e), (s, X_2, f))$ exists in $S_M$. Let $X = X_1 \cup X_2$, clearly $X \subseteq (P_A \cup P_B) - P_M$. It is then easy to show that $(s, X, (e, f)) \in S_M$.

The invariance is straightforward since $X = X_1 \cup X_2$. Based on totality, Zig and Zag properties of $Z$ are immediate.

5.5 Discussion and Future Work

In this chapter, we presented a preliminary report on composing static models and event models with different vocabularies. We studied the algebraic properties of the newly introduced composition operator with the presence of the product update. We gave some results on the decomposition of locally generated models.

Definitely more questions about decomposability of models should be asked. For example, in [CLDQ09], a symmetry reduction technique is proposed in the setting of multi-agent systems, while it is not very clear whether every symmetric model can be decomposed in a non-trivial way by merging compositions. Note that non-symmetric models can also be decomposed: there are locally-generated models which do not have non-trivial symmetric structure. More ambitious agenda is to logically characterize some decomposable classes of models. To systematically answer such decomposition questions, we may need to find help in both graph theory and modal logic.

Ditmarsch and French [vDF09] studied between models with the same vocabulary in the context of product updates. They prove that when restricted to finite models:

$M, e \models N, t$ iff there exists an event model $(A, e)$ such that $N, t \models M \otimes A, (s, e)$.

Compared to this result, our Theorem 5.2.6 requires a much stronger condition. The reason is that essentially we only have very weak propositional “preconditions” when composing the model: matching the values of the basic propositions in the common vocabulary. This motivates the future extension with more complicated matching conditions when composing the models.

In Theorem 5.3.3, we decompose a locally-generated model w.r.t. the merging composition only. However, we can also decompose the model by composing $M_1, \ldots, M_n$ and then, instead of composing $M_0$, restrict the resulting model by a public announcement of the possible states of affairs. Here a more general question arises naturally: what are the natural basic operations to construct a model besides

---

3 We gave a characterization of models that can be decomposed into two-world building blocks in [EFSUS10]. However, the criteria are not very intuitive, thus we have chosen to omit it here.

4 When the sets of observables are not symmetric, the locally-generated model may be asymmetric.
composition? Is there a normal form of any model $M$ by composition, relativization (public announcement) and perhaps general product updates? Again some clues were hinted at in [vDF09]: the author showed that a simulation can be seen as a bisimulation transformation followed by a model restriction. We leave further studies on this topic to future work.

The combination of communicative actions and vocabulary expansion deserves further study. There is an obvious connection to the dynamics of awareness, as studied in [vBVQ09, dJ09], while our expansion operation can be seen as an action for (publicly) extending the awareness set. In this chapter we fix the set of agent in our discussion, while in other applications (e.g., about awareness) expansions with extra agents may be also relevant.

Finally our investigation is encouraging for epistemic model checking with dynamic epistemic logic, for it suggests ways to check relevant epistemic properties on small components of large models. We will pursue this line of research further in Chapter 7 of this thesis.