Compositional Verification of Multi-Agent Systems in Temporal Multi-Epistemic Logic

Joeri Engelfriet, Catholijn M. Jonker, Jan Treur

Vrije Universiteit Amsterdam
Department of Mathematics and Computer Science, Artificial Intelligence Group
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
URL: http://www.cs.vu.nl/~{joeri,jonker,treur}, Email: {joeri,jonker,treur}@cs.vu.nl

Abstract

Compositional verification aims at managing the complexity of the verification process by exploiting compositionality of the system architecture. In this paper we explore the use of a temporal epistemic logic to formalize the process of verification of compositional multi-agent systems. The specification of a system, its properties and their proofs are of a compositional nature, and are formalized within a compositional temporal logic: Temporal Multi-Epistemic Logic. It is shown that compositional proofs are valid under certain conditions. Finally, the possibility of incorporating default persistence of information in a system, is explored.

1 Introduction

It is a recent trend in the literature on verification to study the use of compositionality and abstraction to structure the process of verification; for example, see [1], [8], [18]. In [19] a compositional verification method was introduced for (formal specifications of) multi-agent systems. In that paper, properties to be verified were formalized semantically in terms of temporal epistemic models, and proofs were made by hand, like mathematicians do. The current paper focuses on the requirements for the choice and use of a suitable logic within which both the properties to be verified and their proofs can be formalized. For the particular application of the logic the following requirements for the logic itself and for the use of the logic are of importance:

- compositional structure: properties and proofs can be structured in a compositional manner, in accordance with the compositional structure of the system design.
- dynamics and time: dynamic properties can be expressed, reasoning and induction over time is possible.
- incomplete information states can be expressed.
- transparency: the proof system and the semantics are transparent and not unnecessarily complicated.

In the following sections, *Temporal Multi-Epistemic Logic* (TMEL) is introduced and shown to be a suitable logic; this logic is a generalization of the Temporal Epistemic Logic TEL introduced in [10], [11]; see also [9], [12]. The generalization is made by adding multiple epistemic operators according to the hierarchical compositional structure of the system to be verified. This generalization was inspired by [17], were multiple modal operators were introduced (in their case without hierarchical compositional structure) to verify multi-agent systems specified in Concurrent METATEM. By choosing temporal epistemic logic as a point of departure, a choice was made for a discrete and linear time structure and for time to be global.

The structure of the paper is as follows. In Section 2 the compositional verification method for multi-agent systems is briefly described and an example is given. In Section 3 the temporal multi-epistemic logic is defined. Section 4 discusses compositional temporal theories, Section 5 compositional proof structures, and

Section 6 focuses on how to treat non-classical semantics related to default persistence of information.

2 Compositional Verification

The purpose of verification is to prove that, under a certain set of assumptions, a system satisfies a certain set of properties, for example the design requirements. In the approach introduced in [19], this is done by mathematical proof (i.e., a proof in the form mathematicians are accustomed to), which proves that the specification of the system together with the assumptions implies the properties that the system needs to fulfill. A compositional multi-agent system can be viewed and specified at different levels of abstraction. Viewed from the top level, denoted by L_0, the complete multi-agent system is one component S, where internal information and processes are left unspecified at this level of abstraction (information and process hiding). At the next level of abstraction, L_1, the internal structure of the system is given in terms of its components (as an example, see the agents A and B and the external world EW in Figure 1), but the details of the components are hidden. At the next lower level of abstraction, L_2, (for example) the agent A is specified as a composition of sub-components (see Figure 2). Some components may not be composed of sub-components; such components are called *primitive*. The example has been designed using the compositional development method DESIRE, see [6]. This is a method to develop multi-agent systems according to a compositional structure. The approach to compositional verification addressed in this paper can be used for multi-agent systems designed on the basis of DESIRE, but also for systems designed on the basis of any other method using compositionality as a design principle.

Compositional verification takes into account this compositional structure during the verification process. Properties of a component are only to be expressed using the language specified for the component's interfaces (and not the languages specified for sub-components or super-components); this restricts the space of the properties that can be formulated drastically. Verification of a composed component is done using properties of the sub-components it embeds and the component's specification (which specifies how it is composed of its sub-components). The assumptions on its sub-components under which the component functions properly, are properties to be proven for these sub-components. This implies that properties at different levels of abstraction are involved in the verification process. These properties have hierarchical logical relations in the sense that at each level, given the component's specification, a property is logically implied by (a conjunction of) the lower level properties that relate to it in the hierarchy (see Figure 3); of course, also logical relations between properties within one abstraction level may exist.

The example multi-agent model used in this paper is composed of two co-operative information gathering agents, A and B, and a component EW representing the external world (see Figure 1). Each of the agents is able to acquire partial information about the external world (by observation). Each agent's own observations are insufficient to draw conclusions of a desired type, but the combined information of both agents is sufficient. Therefore communication is required to be able to draw conclusions. The agents can communicate their own observation results and requests for observation information of the other agent. This quite common situation is simplified to the following materialized form. The world situation consists of an object that has to be classified. One agent can only observe the bottom view of the object (e.g., a circle), the other agent the side view (e.g., a square). By exchanging and combining observation information they are able to classify the object (e.g., a cylinder, expressed by the atom object_type(cylinder)).

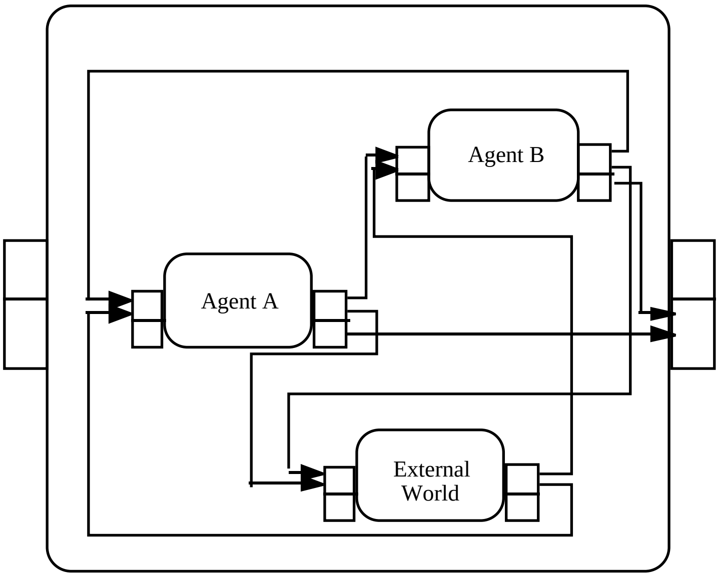

Fig. 1. The example Multi-Agent System for Co-operative Information Gathering

Communication from the agent A to B takes place in the following manner:

- the agent A generates at its output interface a statement of the form:

  to_be_communicated_to(<*type*>, <*atom*>, <*sign*>, B)

- the information is transferred to B; thereby it translated into

  communicated_by(<*type*>, <*atom*>, <*sign*>, A)

In the example <*type*> can be filled with a label request or world_info, <*atom*> is an atom expressing information on the world, and <*sign*>, is one of pos or neg, to indicate truth or falsity.

Interaction between agent A and the world takes place as follows:

- the agent A generates at its output interface a statement of the form:

  to_be_observed(<*atom*>)

- the information is transferred to EW; thereby it is translated into

  to_be_observed_by(<*atom*>, A)

- the external world EW generates at its output interface a statement of the form:

  observation_result_for(<*atom*>, <*sign*>, A)

- the information is transferred to A; thereby it is translated into

  observation_result(<*atom*>, <*sign*>)

Part of the output of an agent are conclusions about the classification of the object of the form object_type(ot); these are transferred to the output of the system.

To be able to perform its tasks, each agent is composed of four components, see Figure 2: three for generic agent tasks (world interaction management, or WIM for short, which reasons about the interaction with the outside world, agent interaction management, or AIM, which reasons about the interaction with other agents, and own process control, or OPC, which reasons about the control of the agent itself; in this example it determines the agent characteristics, for example whether the agent is pro-active or reactive), and one for an agent specific task (object classification, or OC). Since the two agents have a similar architecture, the notation A.WIM is used, for example, to denote component WIM of agent A. As an example of how this agent model works, information describing communication by the agent B to the agent A is transferred to the (input interface of the) component AIM within A (in the form of an atom communicated_by(<*type*>, <*atom*>, <*sign*>, A)). In the component AIM the communicated information is identified (by a meta-reasoning process that interprets the communication) and at the output interface of AIM the atom new_world_info(<*atom*>, <*sign*>) is generated. From this output interface the information is transferred to the component OC, where it is stored as object level information in the form <*atom*> or not <*atom*>, depending on whether <*sign*> is pos or neg. A similar process takes place when observation information is received by the agent, this time through the component WIM.

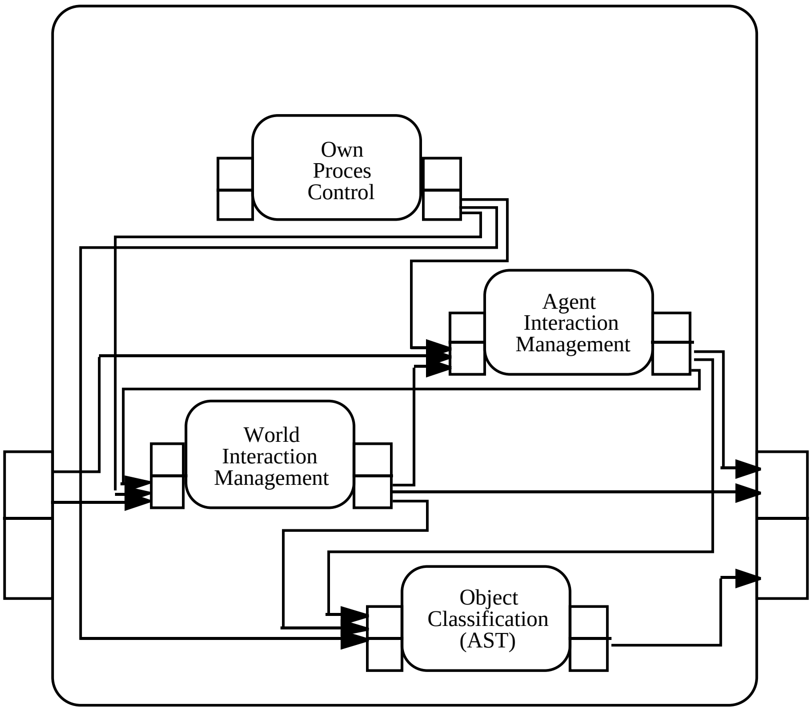

Fig. 2. Composition of an agent

This example multi-agent system has been verified for all 64 cases where each of the two agents may be pro-active or reactive with respect to observation, communication and/or reasoning in any combination (see [19]; in Figure 3 a small part of the properties and logical relations found is depicted). The example used to illustrate the formalization in the current paper is restricted to a pro-active agent A and a reactive agent B.

The *compositional verification method* can be formulated informally as follows (for a formalization, see Section 5 below):

A. Verifying one abstraction level against the other

For each abstraction level the following procedure is followed:

1. Determine which properties are of interest for the (higher level) component D; these properties can be expressed only in terms of the vocabulary defined for the interfaces of D.
2. Determine assumed properties for the lower level components (expressed in terms of their interface languages) that guarantee D's properties.
3. Prove D's properties on the basis of the properties of its sub-components, using the system specification that defines how D is composed of its sub-components.

B. Verifying a primitive component

For primitive knowledge-based components a number of verification techniques exist in the literature, see for example [20].

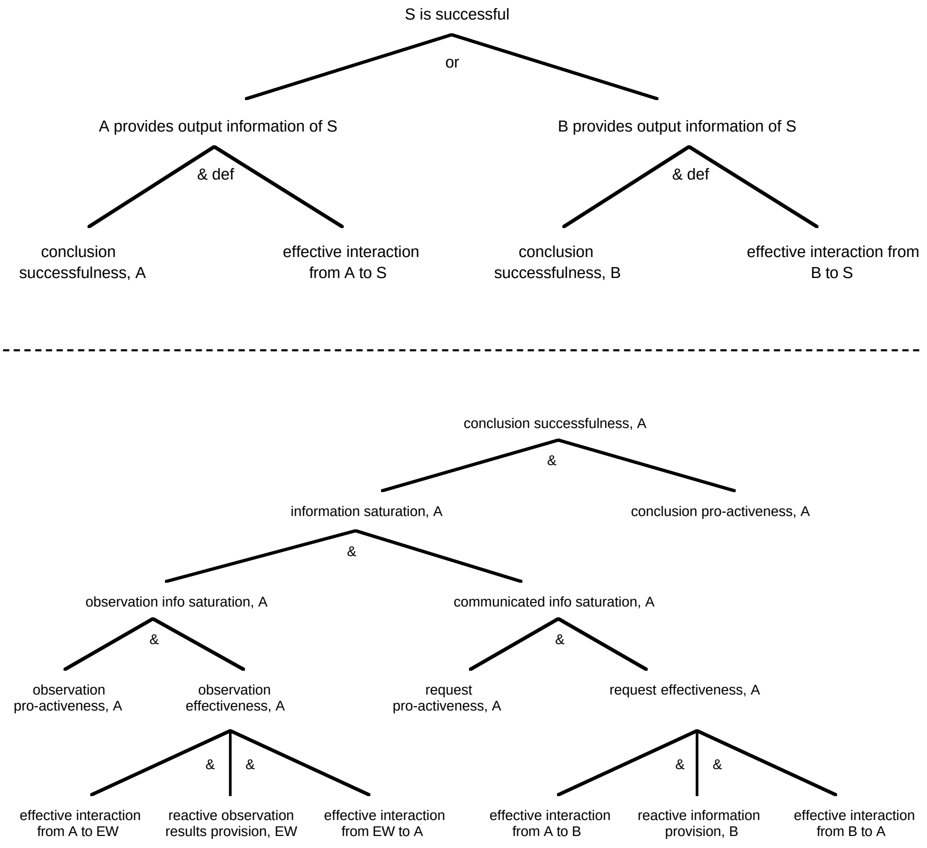
S is successful
or
A provides output information of S
B provides output information of S
& def
& def
conclusion successfulness, A
effective interaction from A to S
conclusion successfulness, B
effective interaction from B to S
conclusion successfulness, A
&
information saturation, A
conclusion pro-activeness, A
&
observation info saturation, A
communicated info saturation, A
&
&
observation pro-activeness, A
observation effectiveness, A
request pro-activeness, A
request effectiveness, A
& &
& &
effective interaction from A to EW
reactive observation results provision, EW
effective interaction from EW to A
effective interaction from A to B
reactive information provision, B
effective interaction from B to A

Fig. 3. Logical relations between a number of properties at different levels of abstraction for the example multi-agent model

C. The overall verification process

To verify the complete system:

1. Determine the properties are that are desired for the whole system.
2. Apply the above procedure **A** iteratively.
   In the iteration the desired properties of abstraction level L_i are either:
   - those determined in step 1, if i = 0, or
   - the assumptions made for the higher level L_{i-1}, if i > 0
3. Verify the primitive components according to **B**.

The results of verification are:

- Properties and assumptions at the different abstraction levels.
- Logical relations between the properties of different process abstraction levels (cf. Figure 3).

Note that both static and dynamic properties and connections between them are covered. Furthermore, process and information hiding limits the complexity of the verification per abstraction level.

3 Temporal Multi-Epistemic Logic

In this section we introduce a logic that can be used to formalize the dynamic aspects of reasoning and the incomplete information states that play a role: temporal multi-epistemic logic. Our approach is in line with what in [15] is called *temporalizing* a given logic; in our case the given logic is a multi-modal epistemic logic based on the component hierarchy of a multi-agent system to be verified. As the base language in which the multi-agent system can express its knowledge and conclusions, we will take a propositional language. Let $\mathbf{COMP}$ be a given set of component names with a hierarchical relation $\mathbf{sub}$ between them, defining a finite tree structure. The following definition formalizes information states and a temporalization of these states, using linear discrete time with a starting point. For convenience we will take the set of natural numbers $\mathbb{N} = \{0, 1, 2, ...\}$ as the time frame.

Definition 3.1 (compositional temporal epistemic model)
a) A *signature* Σ is an ordered sequence of (propositional) atom names. A *compositional epistemic state*, or *compositional information state*, based on Σ, is a collection $(\mathbf{Min}_X, \mathbf{Mint}_X, \mathbf{Mout}_X)_{X \in \mathbf{COMP}}$ of triples of sets $\mathbf{Min}_X$, $\mathbf{Mint}_X$, $\mathbf{Mout}_X$ of propositional models of signature Σ for each of the components $\mathbf{X}$ in $\mathbf{COMP}$.
The set of compositional information states based on Σ is denoted by $\mathbf{CIS(\Sigma)}$, or shortly $\mathbf{CIS}$.
b) Let Σ be a signature. A (propositional) *compositional temporal epistemic model* $\mathbf{M}$ of signature Σ is a mapping $\mathbf{M}: \mathbb{N} \rightarrow \mathbf{CIS(\Sigma)}$. We will sometimes use the notation $(\mathbf{M}_t)_{t \in \mathbb{N}}$ for $\mathbf{M}$.

In the language we introduce modal operators $\mathbf{Cin}_X$, $\mathbf{Cint}_X$, $\mathbf{Cout}_X$ for each component $\mathbf{X}$ in $\mathbf{COMP}$, expressing the input, internal, and output knowledge of the component. We call these operators the *epistemic operators.* Modal formulae can be evaluated in compositional epistemic states at any point in time: a modal formula $\mathbf{Cout}_X\ \alpha$ (where α is propositional) is true in a compositional epistemic state $\mathbf{M}$, denoted $\mathbf{M} \models \mathbf{Cout}_X\ \alpha$, if $\mathbf{m} \models \alpha$ for all $\mathbf{m} \in \mathbf{Mout}_X$ (and similarly for $\mathbf{Cin}_X$ and $\mathbf{Cint}_X$). The operators $\mathbf{Cin}_X$, $\mathbf{Cint}_X$, $\mathbf{Cout}_X$ are very similar to the modal $\mathbf{K}$ operator, so for instance the formula $\neg\, \mathbf{Cout}_X\ \alpha \wedge \neg\, \mathbf{Cout}_X \neg\, \alpha$ denotes that α is unknown in the output state of component $\mathbf{X}$ (i.e., neither known to be true nor known to be false).

We will need a language to express changes over time. To this end in [10], [11] the temporal (uni-modal) epistemic language TEL and its semantics were introduced. To obtain a compositional temporal logic, this logic TEL is generalized in the following manner (the result is called Temporal Multi-Epistemic Logic, or TMEL). Formulae of the form $\mathbf{Cin}_X\ \alpha$, $\mathbf{Cint}_X\ \alpha$ and $\mathbf{Cout}_X\ \alpha$ play the role of atomic propositions. The temporal operators $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{F}$ and $\mathbf{G}$ are used. Intuitively, the temporal formula $\mathbf{F}\alpha$ is true at time $\mathbf{t}$ means that viewed from time point $\mathbf{t}$, the formula α will be true at *some* time in the future (in *some* future information state), $\mathbf{G}\alpha$ is true at time $\mathbf{t}$ means that viewed from time point $\mathbf{t}$, the formula α will be true at *all* time points in the future, and $\mathbf{X}\alpha$ is true at time $\mathbf{t}$ means that α will be true in the next information state. The operator $\mathbf{Y}$ means “true at the previous time point”. Some examples of temporal formulae will be given in the next section. For more details of TEL, see [10], [11]. For temporal epistemic logic different entailment relations can be used, both classical and non-classical; see e.g., [9], [12].

4 Compositional Temporal Theories

In order to embed the compositional verification proofs in temporal multi-epistemic logic, a multi-agent system specification is translated into a temporal theory. As a requirement on this translation we impose that the compositional structure is preserved. This means that instead of one global temporal theory, each component in the hierarchy is translated into a separate temporal theory for this component. Therefore, we introduce collections of sub-languages and collections of temporal theories that are labelled by the set of components **COMP**. A language for a component defines the terms in which its internal information, as well as the information in its input and output interface can be expressed.

Definition 4.1 (language composition)
Let **COMP** be a set of component names with a binary sub-component relation **sub**. *Primitive* components are elements $D \in COMP$ for which no $C \in COMP$ exists with **C sub D**. The other components are called *composed*.
A *language composition* is a collection of sub-languages $(L_C)_{C \in COMP}$, where in each language L_C only the epistemic operators Cin_C, $Cint_C$ and $Cout_C$ are used (and no epistemic operators for other components).
The collection of *interface languages* for the language composition $(L_C)_{C \in COMP}$ is the collection $(L^{if}_C)_{C \in COMP}$ where for any component **D**, the language L^{if}_D is the restriction of L_D to formulae in which the epistemic operator $Cint_D$ does not occur.
The collection of *bridge languages* for the language composition $(L_C)_{C \in COMP}$ is the collection $(L^+_C)_{C \in COMP}$ defined for any component **D** by

$$L^+_D = L_D \cup \bigcup_{C \text{ sub } D} L^{if}_C$$

The *cumulative language composition* for the language composition $(L_C)_{C \in COMP}$ is the collection
$(L^*_C)_{C \in COMP}$ defined for any component **D** by

$L^*_D = L_D \cup \bigcup_{C \text{ sub } D} L^*_C$ if **D** is a composed component
$L^*_D = L_D$ if **D** is a primitive component

Example 4.2 (language composition)
We give part of the languages of some of the components of the example multi-agent system (for **ot** varying over the object types, **r** over shapes, **X** is the agent **A** or **B**, **sign** is **pos** or **neg**):

L_S	$Cout_S$	object_type(ot)
L_A	$Cout_A$	to_be_observed(view(A, r)),
	$Cout_A$	to_be_communicated_to(request, view(B, r), pos, B)
	Cin_A	observation_result(view(A, r), pos)
	Cin_A	communicated_by(world_info, view(B, r), pos, B)
L_{EW}	Cin_{EW}	to_be_observed_by(view(X, r), X),
	$Cout_{EW}$	observation_result_for(view(X, r), sign, X)

Definition 4.3 (theory composition)
Let $(L_C)_{C \in COMP}$ be a language composition. A *compositional temporal theory* for $(L_C)_{C \in COMP}$ is a collection $(T_C)_{C \in COMP}$ where each temporal theory T_C is a theory in the language L^+_C .
Let $(T_C)_{C \in COMP}$ be a compositional temporal theory. The *collection of cumulative theories* $(T^*_C)_{C \in COMP}$ is defined for any component **D** as:

$T^*_D = T_D \cup \bigcup_{C \text{ sub } D} T^*_C$ if **D** is a composed component
$T^*_D = T_D$ if **D** is a primitive component

Example 4.4 (partial compositional theory; a composed component) For each of the components of the multi-agent system its specification can be translated into a temporal theory. The part of the theory for the top level component that is relevant to prove successfulness of the system is the following (again, **ot** ranges over the object types, **r** over shapes, **X** is the agent **A** or **B**, **sign** is **pos** or **neg**):

$$
\begin{array}{ll}
T_S: & \mathbf{Y}\ \mathbf{Cout}_X\ \textbf{to\_be\_observed(view(X,r))} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{EW}\ \textbf{to\_be\_observed\_by(view(X,r), X)} \\
 & \mathbf{Y}\ \mathbf{Cout}_A\ \textbf{to\_be\_communicated\_to(request, view(B,r), pos, B)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_B\ \textbf{communicated\_by(request, view(B,r), pos, A)} \\
 & \mathbf{Y}\ \mathbf{Cout}_B\ \textbf{to\_be\_communicated\_to(world\_info, view(B,r), sign, A)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_A\ \textbf{communicated\_by(world\_info, view(B,r), sign, B)} \\
 & \mathbf{Y}\ \mathbf{Cout}_X\ \textbf{object\_type(ot)} \quad \rightarrow\ \mathbf{Cout}_S\ \textbf{object\_type(ot)} \\
 & \mathbf{Y}\ \mathbf{Cout}_X\ \neg\ \textbf{object\_type(ot)} \quad \rightarrow\ \mathbf{Cout}_S\ \neg\ \textbf{object\_type(ot)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{EW}\ \textbf{observation\_result\_for(view(X,r), sign, X)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_X\ \textbf{observation\_result(view(X,r), sign, X)}
\end{array}
$$

For example, the last formula is part of the description of the information links from EW to A and from EW to B. This formula expresses that the information previously in the output of EW is currently contained in the input interface of the agent A (under a simple translation). The part of the theory for agent A that is relevant to prove successfulness of the system is the following:

$$
\begin{array}{ll}
T_A: & \mathbf{Y}\ \mathbf{Cin}_A\ \textbf{observation\_result(view(A,r), sign)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{A.WIM}\ \textbf{observation\_result(view(A,r), sign)} \\
 & \mathbf{Y}\ \mathbf{Cin}_A\ \textbf{communicated\_by(world\_info, view(B,r), sign, B)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{A.AIM}\ \textbf{communicated\_by(world\_info, view(B,r), sign, B)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.WIM}\ \textbf{to\_be\_observed(view(A,r), sign)} \\
 & \qquad \rightarrow\ \mathbf{Cout}_A\ \textbf{to\_be\_observed(view(A,r), sign)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.AIM}\ \textbf{to\_be\_communicated\_to(request, view(B,r), pos, B)} \\
 & \qquad \rightarrow\ \mathbf{Cout}_A\ \textbf{to\_be\_communicated\_to(request, view(B,r), pos, B)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.OC}\ \textbf{object\_type(ot)} \quad \rightarrow\ \mathbf{Cout}_A\ \textbf{object\_type(ot)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.OC}\ \neg\ \textbf{object\_type(ot)} \rightarrow\ \mathbf{Cout}_A\ \neg\ \textbf{object\_type(ot)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.AIM}\ \textbf{communicated\_by(request, view(A,r), sign, B)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{A.WIM}\ \textbf{requested(view(A,r))} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.AIM}\ \textbf{new\_world\_info(view(B,r), pos)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{A.OC}\ \textbf{view(B,r)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.AIM}\ \textbf{new\_world\_info(view(B,r), neg)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{A.OC}\ \neg\textbf{view(B,r)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.WIM}\ \textbf{new\_world\_info(view(A,r), pos)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{A.OC}\ \textbf{view(B,r)} \\
 & \mathbf{Y}\ \mathbf{Cout}_{A.WIM}\ \textbf{new\_world\_info(view(A,r), neg)} \\
 & \qquad \rightarrow\ \mathbf{Cin}_{A.OC}\ \neg\textbf{view(B,r)}
\end{array}
$$

Example 4.5 (partial compositional theory; a primitive component) Primitive components can, for example, be specified by logical rules of the form 'conjunction of literals' implies 'literal'), as is the case in DESIRE. Consider the following rule of the knowledge base of the primitive component object classification:

if view(A, circle) and view(B, square) then object_type(cylinder)

This rule can be formalized in **TMEL** by:

$$\phi \wedge \mathbf{Y}\ \mathbf{Cin}_{X.OC}\ \textbf{view(A, circle)} \wedge \mathbf{Y}\ \mathbf{Cin}_{X.OC}\ \textbf{view(B, square)} \rightarrow \mathbf{Cout}_{X.OC}\ \textbf{object\_type(cylinder)}$$

where ϕ is a formula expressing control information that allows the rule to be used (for example, the component should be active).

5 Compositional Proof Structures

Verification proofs are composed of proofs at different levels of abstraction (see Figure 3). These proofs involve properties of the components at these abstraction levels.

Definition 5.1 (composition of properties)
A *composition of properties* for a language composition $(L_C)_{C \in COMP}$ is a collection $(P_C)_{C \in COMP}$ where for each C the set P_C is a set of temporal statements in the language L^{if}_C.

Note that in our approach it is not allowed to phrase properties of a component in terms other than those of its interface language.

Example 5.2 (a composition of properties)
In the proof of the successfulness property of S (a small part of which is depicted in Figure 3) the following composition of properties is used (see also Example 4.2):

System S *as a whole*

P_S : $\bigwedge_{ot}$ (F $Cout_S$ object_type(ot) ∨ F $Cout_S$ ¬ object_type(ot))

Agent A *(the pro-active agent)*

P_A : [$\bigwedge_r$ (Cin_A observation_result(view(A,r), pos) ∨
Cin_A observation_result(view(A,r), neg))]
∧ [$\bigwedge_r$ (Cin_A communicated_by(world_info, view(B,r), pos, B) ∨
Cin_A communicated_by(world_info, view(B,r), neg, B))]
→ $\bigwedge_{ot}$ (F $Cout_A$ object_type(ot) ∨ F $Cout_A$ ¬ object_type(ot))
(conclusion pro-activeness, A)

$\bigwedge_r$ F $Cout_A$ to_be_observed(view(A,r))
(observation pro-activeness, A)

$\bigwedge_r$ F $Cout_A$ to_be_communicated_to(request, view(B,r), pos, B),
(request pro-activeness, A)

Agent B *(the reactive agent)*

P_B : $\bigwedge_r$ [Cin_B communicated_by(request, view(B,r), pos, A)
→ (F $Cout_B$ to_be_communicated_to(world_info, view(B,r), pos) ∨
F $Cout_B$ to_be_communicated_to(world_info, view(B,r), neg))]
(reactive information provision, B)

External World EW

P_{EW} : $\bigwedge_r$ [Cin_{EW} to_be_observed_by(view(X,r), X)
→ (F $Cout_{EW}$ observation_result_for(view(X,r), pos) ∨
F $Cout_{EW}$ observation_result_for(view(X,r), neg))]
(reactive observation results provision, EW)

Components within A

$P_{A.OPC}$: F $Cout_{A.OPC}$ pro-active
(pro-activeness, OPC)

$P_{A.AIM}$: [$Cin_{A.AIM}$ pro-active
→ $\bigwedge_r$ F $Cout_{A.AIM}$ to_be_communicated_to(request, view(B,r), pos, B)],
(conditional request pro-activeness, AIM)

[$\bigwedge_r$ $Cin_{A.AIM}$ communicated_by(world_info, view(B,r), sign, B)
→ F $Cout_{A.AIM}$ new_world_info(view(B,r), sign)]

$$\mathbf{P_{A.WIM}}: \quad [\ \mathbf{Cin_{A.WIM}\ pro\text{-}active} \rightarrow \bigwedge_{r} \mathbf{F\ Cout_{A.WIM}\ to\_be\_observed(view(A,r))}\ ],$$

(conditional observation pro-activeness, WIM)

$$[\ \bigwedge_{r} \mathbf{Cin_{A.WIM}\ observation\_result(view(A,r),\ sign)} \\ \rightarrow \mathbf{F\ Cout_{A.WIM}\ new\_world\_info(view(A,r),\ sign)}\ ]$$

$$\mathbf{P_{A.OC}}: \quad \bigwedge_{r,X} (\ \mathbf{Cin_{A.OC}\ view(X,r)} \vee \mathbf{Cin_{A.OC}} \neg\, \mathbf{view(X,r)}\ ) \\ \rightarrow \bigwedge_{ot} (\ \mathbf{F\ Cout_{A.OC}\ object\_type(ot)} \vee \mathbf{F\ Cout_{A.OC}} \neg\, \mathbf{object\_type(ot)}\ )$$

In the proof of the properties shown in Example 5.2, the theories shown in Example 4.4 and 4.5 are used.

Definition 5.3 (compositional and global provability)
For the language composition $\mathbf{(L_C)_{C \in COMP}}$, let a composition of properties $\mathbf{(P_C)_{C \in COMP}}$ and a compositional temporal theory $\mathbf{(T_C)_{C \in COMP}}$ be given. Let $\mathrel{|\!\sim}$ be an entailment relation for temporal multi-epistemic logic.
a) The composition of properties $\mathbf{(P_C)_{C \in COMP}}$ is *compositionally provable* with respect to $\mathrel{|\!\sim}$ from the compositional temporal theory $\mathbf{(T_C)_{C \in COMP}}$ if for each component $\mathbf{D}$ the following holds:

$$\mathbf{T_D} \cup \bigcup_{\mathbf{C\ sub\ D}} \mathbf{P_C} \mathrel{|\!\sim} \mathbf{P_D} \qquad \text{if } \mathbf{D} \text{ is composed}$$

$$\mathbf{T_D} \mathrel{|\!\sim} \mathbf{P_D} \qquad \text{if } \mathbf{D} \text{ is primitive}$$

b) The composition of properties is *globally provable* with respect to $\mathrel{|\!\sim}$ from the compositional temporal theory $\mathbf{(T_C)_{C \in COMP}}$ if for each component $\mathbf{D}$ the following holds:

$$\mathbf{T^*_D} \mathrel{|\!\sim} \mathbf{P_D}$$

For example, the collection of success properties of Example 5.2 turns out to be globally provable from the compositional temporal theory $\mathbf{(T_C)_{C \in COMP}}$, with respect to the provability relation of classical entailment in TMEL, augmented with a default persistence assumption (see the next section).

Compositional provability does not necessarily imply global provability. However, the implication holds if the entailment relation satisfies, apart from reflexivity (if $\mathbf{V} \subseteq \mathbf{W}$, then $\mathbf{W} \mathrel{|\!\sim} \mathbf{V}$), the property of transitivity:

$$\mathbf{T} \mathrel{|\!\sim} \mathbf{U} \ \ \& \ \ \mathbf{U} \mathrel{|\!\sim} \mathbf{W} \Rightarrow \mathbf{T} \mathrel{|\!\sim} \mathbf{W} \qquad (\textit{Transitivity})$$

for all sets of formulae $\mathbf{T}$, $\mathbf{U}$, $\mathbf{W}$. It is well-known that transitivity and reflexivity imply monotonicity.

Proposition 5.4
If the entailment relation $\mathrel{|\!\sim}$ satisfies, in addition to reflexivity, transitivity, then compositional provability with respect to $\mathrel{|\!\sim}$ implies global provability with respect to $\mathrel{|\!\sim}$. In particular, if $\vdash$ is a classical provability relation for temporal multi-epistemic logic, then compositional provability with respect to $\vdash$ implies global provability with respect to $\vdash$.

This proposition shows that for classical entailment the implication holds. But, for example, for an entailment relation taking into account minimal change the implication does not hold. In the light of these results, for compositional verification a classical proof system is the best choice.

6 Default Persistence and Revision

The conditions under which a classical inference relation can be used depend on the specific form of semantics. For example, in DESIRE a default persistence assumption has been made: it is only specified what has to be changed; all other information is meant to persist in time. An exception is made for information that has to be retracted because it was derived from information that does not hold anymore. In this section we discuss a manner in which default persistence and revision can be treated within temporal multi-epistemic logic.

In principle, a compositional specification can be formalized by executable temporal formulae. Roughly spoken, executable temporal formulae are temporal formulae of the form

$$\textbf{declarative past} \quad \Rightarrow \quad \textbf{imperative future}$$

For more details on this paradigm, and the different variants within, see [2], [3]. For our purposes the following definition is chosen. *Simplified executable temporal formulae* are formulae of the form

$$\textbf{past and present} \quad \Rightarrow \quad \textbf{present}$$

The right hand side of these formulae **F** are called *heads*, denoted by **head(F)**; they are taken from the set

$$\begin{aligned}\textbf{HEADS} \;=\;\; & \{\, \mathbf{CL} \mid \mathbf{L}\ \textbf{propositional literal, C epistemic operator} \,\} \cup \\ & \{\, \neg\, \mathbf{CA} \wedge \neg\ \mathbf{C} \neg\, \mathbf{A} \mid \mathbf{A}\ \textbf{propositional atom, C epistemic operator} \,\}\end{aligned}$$

The left hand side of **F** is called *body*, denoted by **body(F)**. Within the body, the ‘past’ part is a formula that refers strictly to the past. The ‘present’ part is a conjunction of temporal literals that are either of the form **CL** or **¬CL**.

The intended semantics of these formulae is that it is only specified what has to be changed. All other information is meant to persist (default persistence) in time, with an exception for information that has to be revised because it was derived from information that does not hold anymore. In principle this entails non-classical semantics. However, a translation is possible into temporal theories with classical semantics if a form of temporal completion (similar to Clark’s completion in logic programming) is applied:

Let **T** be a temporal theory consisting of simplified executable temporal formulae. For each **H ∈ HEADS** define

$$\mathbf{T_H} \quad = \quad \{\, \mathbf{F} \in \mathbf{T} \mid \mathbf{head(F)} = \mathbf{H} \,\}$$

Let **L** be a literal and **C** an epistemic operator; define

$$\begin{aligned}\mathbf{tc(T_{CL})} \quad = \quad & [\bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{CL}}\} \vee \\ & \quad (\neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{C\sim L}}\} \wedge \\ & \quad \neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{\neg\, CL\, \wedge\, \neg\, C\, \sim L}}\} \wedge \\ & \quad \mathbf{YCL}\ )\ ] \\ & \leftrightarrow \mathbf{CL}\end{aligned}$$

$$\begin{aligned}\mathbf{tc(T_{\neg\, CL\, \wedge\, \neg\, C\, \sim L})} \;=\; & [\bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{\neg\, CL\, \wedge\, \neg\, C\, \sim L}}\} \vee \\ & \quad (\neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{C\sim L}}\} \wedge \\ & \quad \neg \bigvee \{\mathbf{body(F)} \mid \mathbf{F} \in \mathbf{T_{CL}}\} \wedge \\ & \quad \neg\, \mathbf{YCL} \wedge \neg\, \mathbf{YC \sim L}\ )\ ] \\ & \leftrightarrow \neg\, \mathbf{CL} \wedge \neg\, \mathbf{C \sim L}\end{aligned}$$

Here **~L** denotes the complementary literal of **L**. The intuition behind these formulae is the following: a literal is (known to be) true in a component exactly when either there was an applicable rule making it true, or it was true before, and all rules making the literal false or unknown, are not applicable.

The *temporal completion* of **T** is defined by

$$\begin{array}{l} \mathrm{tc(T)} = \{\ \mathrm{tc}(\mathrm{T}_{\mathrm{CL}})\ |\ \mathrm{L\ literal,\ C\ epistemic\ operator}\ \} \cup \\ \qquad\qquad \{\ \mathrm{tc}(\mathrm{T}_{\neg\,\mathrm{CL}\,\wedge\,\neg\,\mathrm{C}\,\sim\mathrm{L}})\ \ |\ \mathrm{L\ literal,\ C\ epistemic\ operator}\ \ \} \end{array}$$

Under a consistency assumption the right part $\{\ \mathrm{tc}(\mathrm{T}_{\neg\,\mathrm{CL}\,\wedge\,\neg\,\mathrm{C}\,\sim\mathrm{L}})\ \ |\ \mathrm{L\ literal,\ C\ epistemic\ operator}\ \}$ of the above union is already implied by the left part $\{\ \mathrm{tc}(\mathrm{T}_{\mathrm{CL}})\ |\ \mathrm{L\ literal,\ C\ epistemic\ operator}\ \}$.

Example 6.1 (temporal completion of a link formalization)

Let **T** be the temporal theory (a subset of $\mathbf{T}_S$) that formalizes the information link from EW to the agent X; see Example 4.4. The temporal completion of **T** contains the set of formulae:

$$\begin{array}{l} [\ \mathrm{Y\ Cout}_{EW}\ \text{observation\_result\_for(view(X,r), sign, X))}\ \ \vee \\ \ \ (\ \neg\,\mathrm{Y\ Cout}_{EW}\ \neg\ \text{observation\_result\_for(view(X,r), sign, X))}\ \wedge \\ \qquad\qquad \mathrm{Y\ Cin}_{X}\ \text{observation\_result(view(X,r), sign, X))]} \end{array}$$

$$\leftrightarrow \qquad \mathrm{Cin}_{X}\ \text{observation\_result(view(X,r), sign, X)}$$

$$\begin{array}{l} [\ \mathrm{Y\ Cout}_{EW}\ \neg\ \text{observation\_result\_for(view(X,r), sign, X))}\ \ \vee \\ \ \ (\ \neg\,\mathrm{Y\ Cout}_{EW}\ \text{observation\_result\_for(view(X,r), sign, X))}\ \wedge \\ \qquad\qquad \mathrm{Y\ Cin}_{X}\ \neg\ \text{observation\_result(view(X,r), sign, X))]} \end{array}$$

$$\leftrightarrow \qquad \mathrm{Cin}_{X}\ \neg\ \text{observation\_result(view(X,r), sign, X)}$$

Note that the result of temporal completion is a temporal theory that is not anymore in executable format.

The temporal completion allows to formalize proofs in a classical proof system. This means that, given a compositional theory $(\mathrm{T_C})_{\mathrm{C} \in \mathrm{COMP}}$, we should consider the completion of the union of these theories, i.e. $\mathrm{tc}(\mathrm{T^*_S})$ where **S** is the component of the entire system, for global provability. On the other hand, for compositional provability, we have to consider $(\mathrm{tc}(\mathrm{T_C}))_{\mathrm{C} \in \mathrm{COMP}}$. In general, however, $\mathrm{tc}(\mathrm{T^*_S})$ need not be identical to the union of $(\mathrm{tc}(\mathrm{T_C}))_{\mathrm{C} \in \mathrm{COMP}}$. This may occur when a literal occurs in the head of two rules belonging to different components. Then there will be one formula $\mathrm{tc}(\mathrm{T_{CL}})$ in $\mathrm{tc}(\mathrm{T^*_S})$, combining the two rules (and this is intended), but there will be two in the union of $(\mathrm{tc}(\mathrm{T_C}))_{\mathrm{C} \in \mathrm{COMP}}$, one for each component (and this is not intended). In the case of simplified executable temporal formulae we can give a simple criterion which ensures that $\mathrm{tc}(\mathrm{T^*_S})$ is equal to the union of $(\mathrm{tc}(\mathrm{T_C}))_{\mathrm{C} \in \mathrm{COMP}}$. The only thing that is required is that for each formula **CL**, the temporal formulae defining it, are all in one component, i.e., $\mathrm{T_{CL}} \subseteq \mathrm{T_D}$ for some component **D**. It is easy to see that this requirement is sufficient, and it is a requirement satisfied at least by all theories describing components in DESIRE.

Given that this requirement is satisfied, we can of course apply Proposition 5.4 to the compositional theory $(\mathrm{tc}(\mathrm{T_C}))_{\mathrm{C} \in \mathrm{COMP}}$:

Corollary 6.2

For the language composition $(\mathbf{L_C})_{\mathbf{C} \in \mathbf{COMP}}$, let a composition of properties $(\mathbf{P_C})_{\mathbf{C} \in \mathbf{COMP}}$ and a compositional temporal theory $(\mathbf{T_C})_{\mathbf{C} \in \mathbf{COMP}}$ be given. Let $\vdash$ be a classical provability relation for temporal multi-epistemic logic.
If $(\mathbf{P_C})_{\mathbf{C} \in \mathbf{COMP}}$ is compositionally provable with respect to $\vdash$ from the compositional temporal theory $(\mathbf{tc(T_C)})_{\mathbf{C} \in \mathbf{COMP}}$ then $(\mathbf{P_C})_{\mathbf{C} \in \mathbf{COMP}}$ is globally provable with respect to $\vdash$ from the compositional theory $(\mathbf{tc(T_C)})_{\mathbf{C} \in \mathbf{COMP}}$.

The notion of temporal completion defined above expresses default persistence for all information in the system. This implies that in all cases where no default persistence is intended, explicit temporal rules are required that prohibit the persistence. For example, to describe retraction of information that deductively depends on other information that was revised (such as occurs, for example, in the truth maintenance process of primitive components in DESIRE), it is needed in addition to explicitly express a temporal rule, e.g., (for the Example 4.5) of the form:

$$\phi \wedge \neg\, (\,\mathbf{Y}\ \mathbf{Cin}_{\mathrm{X.OC}}\ \mathbf{view(A,\ circle)} \quad \wedge\ \mathbf{Y}\ \mathbf{Cin}_{\mathrm{X.OC}}\ \mathbf{view(B,\ circle)}\,) \rightarrow$$
$$\neg\, \mathbf{Cout}_{\mathrm{X.OC}}\ \mathbf{object\_type(sphere)} \wedge \neg\, \mathbf{Cout}_{\mathrm{X.OC}}\ \neg\, \mathbf{object\_type(sphere)}$$

where ϕ is again a formula expressing control information that allows the rule to be used (for example, the component should be active). Another approach is to define a more sensitive form of temporal completion already taking this into account, in which case these separate rules for retraction are not needed.

7 Conclusions

The compositional verification method formalized in this paper can be applied to a broad class of multi-agent systems. Compositional verification for one process abstraction level deep is based on the following very general assumptions:

- a multi-agent system consists of a number of agents and external world components.
- agents and components have explicitly defined input and output interface languages; all other information is hidden; information exchange between components can only take place via the interfaces (*information hiding*).
- a formal description exists of the manner in which agents and world components are composed to form the whole multi-agent system (*composition relation*).
- the semantics of the system can be described by the evolution of states of the agents and components at the different levels of abstraction (*state-based semantics*).

This non-iterative form of compositional verification can be applied to many existing approaches, for example, to systems designed using Concurrent METATEM [16], [17]. Compositional verification involving more abstraction levels assumes, in addition:

- some of the agents and components are composed of sub-components.
- a formal description exists of the manner in which agents or components are composed of sub-components (*composition relation*).
- information exchange between components is only possible between two components at the same or adjacent levels (*information hiding*).

Currently not many approaches to multi-agent system design exist that exploit iterative compositionality. One approach that does is the compositional development method for multi-agent systems DESIRE. The compositional verification method formalized in this paper fits well to DESIRE, but not exclusively.

Two main advantages of a compositional approach to modelling are the transparent structure of the design and support for reuse of components and generic models. The compositional verification method extends these main advantages to (1) a well-structured verification process, and (2) the reusability of proofs for properties of components that are reused.

The first advantage entails that both conceptually and computationally the complexity of the verification process can be handled by compositionality at different levels of abstraction. Apart from the work reported in [19], a generic model for diagnosis has been verified [7] and a multi-agent system with agents negotiating about load-balancing of electricity use [5]. The second advantage entails: if a modified component satisfies the same properties as the previous one, the proof of the properties at the higher levels of abstraction can be reused to show that the new system has the same properties as the original. This has high value for a library of reusable generic models and components. The verification of generic models forces one to find the assumptions under which the generic model is applicable for the considered domain, as is also discussed in [13]. A library of reusable components and generic models may consist of both specifications of the components and models, and their design rationale. As part of the design rationale, at least the properties of the components and their logical relations can be documented.

The usefulness of a temporal multi-epistemic logic, TMEL, a generalization of temporal epistemic logic was investigated to formalize verification proofs. As a test, the properties and proofs that were found for verification of an example multi-agent system for co-operative information gathering [19] were successfully formalized within the logic TMEL. Our study shows that Temporal Multi-Epistemic Logic provides enough expressivity for dynamics and reasoning about time, and formalizes incomplete information states in an adequate manner. To obtain the right structure in accordance with the compositional system design, the logic is equipped with a number of compositional structures: compositions of sub-languages, compositional theories, and compositional provability. It was established that under the assumption that the provability relation is reflexive and transitive, compositional provability implies global provability. Therefore this logic is adequate if the executable temporal theories formalizing a specification are temporally completed, a temporal variant of Clark’s completion for logic programs. In this case classical provability can be used, which is much more transparent than the more complicated non-classical provability relations that are possible.

In [17] a temporal belief logic, TBL, was introduced to define semantics and verify properties for systems specified in Concurrent METATEM [16]. A similarity with our approach as introduced above is that in both cases modal operators are used to distinguish knowledge of different agents, and a discrete linear time temporal logic is built on top of the multi-modal logic. A main difference in comparison to [17] is that our approach exploits compositionality. In Concurrent METATEM no iterated compositional structures can be defined, as is the case in DESIRE. Therefore verification in TBL always takes place at the global level, instead of the iterated compositional approach to verification in TMEL. Another difference is that in our approach the states in the base logic are in principle three-valued, whereas the states in Concurrent METATEM are two-valued: an atom in a state that is not true is assumed false in this state.

A future continuation of this work will consider the development of tools for compositional verification. To support the handwork of verification it would be useful to have tools to assist in the creation of the proof.

References

1. Abadi, M. and L. Lamport (1993). Composing Specifications, *ACM Transactions on Programming Languages and Systems*, Vol. 15, No. 1, pp. 73-132.
2. Barringer, H., M. Fisher, D. Gabbay, and A. Hunter (1991). Meta-Reasoning in Executable Temporal Logic, in: J. Allen, R. Fikes, E. Sandewall, *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning,* KR'91.
3. Barringer, H., M. Fisher, D. Gabbay, R. Owens, and M. Reynolds (1996). *The Imperative Future: Principles of Executable Temporal Logic*, Research Studies Press Ltd. and John Wiley & Sons.

4. Benthem, J.F.A.K. van (1983). *The Logic of Time : a Model-theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*, Reidel, Dordrecht.

5. Brazier, F.M.T., F. Cornelissen, R. Gustavsson, C.M. Jonker, O. Lindeberg, B. Polak, and J. Treur, (1998). Compositional Design and Verification of a Multi-Agent System for One-to-Many Negotiation. In: Proceedings of the Third International Conference on Multi-Agent Systems, ICMAS'98, IEEE Computer Society Press.

6. Brazier, F.M.T., B.M. Dunin-Keplicz, N.R. Jennings, and J. Treur, (1995) Formal Specification of Multi-Agent Systems: a Real World Case, In: Lesser, V. (ed.), *Proc. of the First International Conference on Multi-Agent Systems, ICMAS'95*, MIT Press, pp. 25-32. Extended version in: Huhns, M. and Singh, M. (eds.), *International Journal of Co-operative Information Systems, IJCIS* vol. 6 (1), special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems, pp. 67-94.

7. Cornelissen, F., C.M. Jonker, and J. Treur (1997). Compositional Verification of Knowledge-based Systems: a Case Study for Diagnostic Reasoning. In: E. Plaza, R. Benjamins (eds.), *Knowledge Acquisition, Modelling and Management, Proc. of the 10th EKAW*, Lecture Notes in AI, vol. 1319, Springer Verlag, pp. 65-80.

8. Dams, D., R. Gerth, and P. Kelb (1996). *Practical Symbolic Model Checking of the full μ-calculus using Compositional Abstractions*. Report, Eindhoven University of Technology, Department of Mathematics and Computer Science.

9. Engelfriet, J. (1996). Minimal Temporal Epistemic Logic, *Notre Dame Journal of Formal Logic*, vol. 37, pp. 233-259 (special issue on Combining Logics).

10. Engelfriet, J., and J. Treur (1996). Specification of Nonmonotonic Reasoning. *Proc. International Conference on Formal and Applied Practical Reasoning, FAPR'96* , Springer-Verlag, Lecture Notes in Artificial Intelligence, vol. 1085, pp. 111-125.

11. Engelfriet, J., and J. Treur (1996). Executable Temporal Logic for Nonmonotonic Reasoning; *Journal of Symbolic Computation*, vol. 22, no. 5&6, pp. 615-625.

12. Engelfriet, J., and J. Treur (1997). An Interpretation of Default Logic in Temporal Epistemic Logic. *Journal of Logic, Language and Information*, to appear.

13. Fensel, D., and R. Benjamins (1996). Assumptions in model-based diagnosis. In: B.R. Gaines, M.A. Musen (eds.), *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'96* , Calgary: SRDG Publications, Department of Computer Science, University of Calgary, pp. 5/1-5/18.

14. Fensel, D., A. Schonegge, R. Groenboom, and B. Wielinga (1996). Specification and verification of knowledge-based systems. In: B.R. Gaines, M.A. Musen (eds.), *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'96* ,Calgary: SRDG Publications, Department of Computer Science, University of Calgary, pp. 4/1-4/20.

15. Finger, M. and D. Gabbay (1992). Adding a Temporal Dimension to a Logic System, *Journal of Logic, Language and Information* **1**, pp. 203-233.

16. Fisher, M. (1994). A survey of Concurrent METATEM - the language and its applications. In: D.M. Gabbay, H.J. Ohlbach (eds.), Temporal Logic - Proceedings of the First International Conference, Lecture Notes in AI, vol. 827, pp. 480-505.

17. Fisher, M., and M. Wooldridge, (1997). Specification and Verification of Multi-Agent Systems. In: Huhns, M. and Singh, M. (eds.), *International Journal of Co-operative Information Systems, IJCIS* vol. 6 (1), special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems,

18. Hooman, J. (1994). Compositional Verification of a Distributed Real-Time Arbitration Protocol. *Real-Time Systems*, vol. 6, pp. 173-206.

19. Jonker, C.M. and J. Treur (1997). Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness. In: W.P. De Roever, H. Langmaack, A. Pnueli, (eds.). *Proceedings of the International Symposium on Compositionality, COMPOS'97*, Springer Verlag, to appear.

20. Treur, J., and M. Willems (1994). A logical foundation for verification. In: *Proceedings of the Eleventh European Conference on Artificial Intelligence, ECAI'94*, A.G. Cohn (ed.), John Wiley & Sons, Ltd., pp. 745-749.