Dynamic maintenance of states is handled by the explicit semantic links (activated in a controlled manner) and the standard update facilities built in in the DESIRE software environment.

- Limited reasoning of an agent is possible with explicit control by means of strategic knowledge.
- Complex, reflective agents can be specified in a conceptually transparent manner.
- Flexibility in modelling: it is easy to build different types of agents.
- Explicit and directed observation acts and their control, can be specified.
- Explicit and directed communication acts and their control, can be specified.
- The semantics of the dynamics can be based on (compositional) temporal models representing behavioural patterns.

It turned out to be possible to completely specify nontrivial multi-agent examples in DESIRE (see [3]). Moreover, [2] reports work on modelling interaction between a knowledge-based system, the world and a user based on the compositional multi-agent approach presented here. Integration of different types of components (e.g., neural networks, optimization algorithms) can be assured by specifying their interface and referring to an external component or specification. Further investigations are planned in a real world application area: diagnosis of an electricity network. Here the new version of DESIRE will be applied where it is possible to structure specifications according to hierarchical decomposition, enabling one to specify components that consist of other components.

Acknowledgements
Useful discussions about the subject have taken place with Frances Brazier. An earlier draft of this paper was read by Pascal van Eck; this led to a number of improvements in the text.

References


Let $MAS$ be a multi-agent system and let $A$ be an agent in $MAS$ or the material world $MW$.

a) A possible trace of a component $C$ is a sequence of information states $(M_t)_{t \in \mathbb{N}}$ in $IS(C)$. The set of all of them can be denoted by $IS(C)^\mathbb{N}$, or $\text{Traces}(C)$. A possible trace of $A$ is a sequence of information states $(M_t)_{t \in \mathbb{N}}$ in $IS(A)$. The set of all of them can be denoted by $IS(A)^\mathbb{N}$, or $\text{Traces}(A)$. A possible trace of $MAS$ is a sequence of information states $(M_t)_{t \in \mathbb{N}}$ in $IS(MAS)$. The set of all of them can be denoted by $IS(MAS)^\mathbb{N}$, or $\text{Traces}(MAS)$.

b) An element $(M_t)_{t \in \mathbb{N}} \in \text{Traces}(MAS)$ is called a temporal model of $MAS$ if for all time points $t$ the step from $M_t$ to $M_{t+1}$ is defined in accordance with an overall transition. The set of temporal models of $MAS$ forms a subset $\text{BehMod}(MAS)$ of $\text{Traces}(MAS)$. A temporal model describes a trace representing possible (intended) behaviour of the reasoning. One view is that the trace is generated by the transition functions (executing them), given some initial input information. From every initial information setting a trace can be generated by the transitions. All generated traces together form the set $\text{BehMod}(MAS)$. A slightly different view is that the transition functions define a set of (temporal) axioms or constraints $\text{BehTheory}(MAS)$ on temporal models in $\text{Traces}(MAS)$. The possible behavioural alternatives are given by the set of the temporal models satisfying these temporal constraints:

$$\text{TempMod}(MAS) = \{ M \in \text{Traces}(MAS) \mid M \models \text{BehTheory}(MAS) \}$$

where $M \models \text{BehTheory}(MAS)$ holds iff each formula from the set $\text{BehTheory}(MAS)$ has truth value $true$ at every time point in the temporal model $M$. This second view provides a formalization of the intended behavioural patterns in the form of the (intended) models of a logical (temporal) theory in a specific type of temporal logic, giving a declarative (Tarski) semantics. The formal semantics of the behaviour is defined by the set of models $\text{TempMod}(MAS)$. The first view corresponds to the notion of an executable temporal logic. Both views co-exist: executing a temporal theory is a useful technique to construct a model of this theory. The semantic approach based on temporal logic, as sketched above for the case of (compositional) multi-agent systems, already has been applied to classical proof systems, default logic, compositional reasoning systems and meta-level architectures (see [4], [6], [14]).

6 Conclusions and further research

In this paper it was investigated how multi-agent systems with complex agents can be designed and formally specified based on the notion of compositional architecture. Due to our experiences the following requirements for multi-agent systems are fulfilled by compositional architectures and the formal specification language DESIRE:

- Integration of various types of reasoning and acting is possible within one declarative logical framework with component wise classical (partial) semantics and overall temporal semantics.
with \( \pi_{\text{upd}} \) the standard update mapping and \( \pi_{\text{inf}} \) an (inference) mapping such that for each \( N \in IS_m(C) \) the mapping \( \pi_{\text{inf}}^N : IS_o(C) \to IS_o(C) \) defined by \( \pi_{\text{inf}}^N(M) = \pi_{\text{inf}}(M, N) \) is conservative and monotonic and \( \pi_{\text{inf}}^N(M) \leq d_{KBC}(M) \) for all \( M \).

b) Suppose \( C_1 \) and \( C_2 \) are components, \( x, y \in \{ \text{obj, meta} \} \) and \( I \) is a semantic link from \( C_1 \) to \( C_2 \) of type \( < x, y > \). A transition function for the semantic link \( I \) is a mapping

\[
\pi : IS_o(C_1) \times IS_o(C_2) \times IS_m(I) \to IS_o(C_2) \times IS_m(I)
\]

For correct functioning of the system it is required that for all \( M_1, M_2, N, M', N' \) with \( \pi(M_1, M_2, N) = < M'_2, N' > \) and atom \( b \) it holds \( M'_2(b) = M_2(b) \) or

\[
I(<a, M_1(a)>, <b, M'_2(b)>)
\]

for some atom \( a \).

If \( I \) is an internal semantic link of agent \( A \) (i.e., a semantic link between components \( C_1 \) and \( C_2 \) of \( A \)) then a transition function function for \( I \) is called an internal interaction transition function of \( A \); otherwise it is called an external interaction transition function.

c) An internal transition function for (an agent or the material world) \( A \) is a mapping

\[
\pi : IS(A) \to IS(A)
\]

induced by an internal component transition function of \( A \), or by an internal interaction transition function of \( A \).

d) An external (interaction) transition function from (an agent or the material world) \( A \) to (an agent or the material world) \( B \) distinct from \( A \) is a mapping

\[
\pi : IS(A) \times IS(B) \to IS(B) \times IS(A)
\]

induced by an external interaction transition function from a component \( C_1 \) of \( A \) to a component \( C_2 \) of \( B \).

e) An overall transition function for the system \( \text{MAS} \) is a mapping

\[
\pi : IS(\text{MAS}) \to IS(\text{MAS})
\]

induced by an internal transition function for an agent or the material world, or induced by an external transition function between an agent (or the material world) and another, distinct agent (or the material world).

Note that an external transition from an agent \( A \) in principle also changes the information state of the agent itself (e.g., to store that some information has been communicated); however, this change only depends on the information state of the sender \( A \), not of the receiver.

In Definition 5.2 we treated the sequential case: just one agent is active at a time. The definition can be extended to cover the parallel case as well. Important then is to maintain that information coming from different sources is disjoint (e.g., labeled with the source) or to take into account the order in which it is received.

The following definition shows how traces generated by iteratively applying a transition function on the current information state can be interpreted as temporal models, giving a declarative description of the semantics of the behaviour of the system. They can be viewed as the so-called intended (behavioural) models of the system.

**Definition 5.3 (traces and temporal models)**
an agent is a combination of information states of components whereas a state of the whole system is a combination of the latter information states.

**Definition 5.1 (combined and overall information states)**

Let $\text{MAS}$ be a multi-agent system and let $A$ be an agent or the material world $\text{MW}$ in $\text{MAS}$. Its *set of components* is denoted by $\text{C}(A)$ and its *set of semantic links* by $\text{I}(A)$. Here all outgoing semantic links (to other agents and the material world in $\text{MAS}$) are included in $\text{I}(A)$.

a) The *set of possible (combined) information states* for a component $C$ of $A$ is defined by:

$$\text{IS}(C) = \text{IS}_o(C) \times \text{IS}_m(C)$$

b) The *set of possible overall information states* for $A$ is defined by:

$$\text{IS}(A) = \prod_{C \in \text{C}(A)} \text{IS}(C) \times \prod_{I \in \text{I}(A)} \text{IS}_m(I)$$

c) The *set of possible overall information states* for the multi-agent system $\text{MAS}$ is:

$$\text{IS}(\text{MAS}) = \text{IS}(\text{MW}) \times \prod_{A \in \text{A}(\text{MAS})} \text{IS}(A)$$

where $\text{A}(\text{MAS})$ denotes the set of all agents of $\text{MAS}$.

Note that any mapping on one or more factors of a cartesian product can be extended in a canonical manner to (i.e., it *induces*) a mapping on the whole cartesian product (by leaving the other factors out of consideration). Furthermore, a mapping $\pi$ of information states is called *conservative* if $M \leq \pi(M)$ for all $M$; it is called *monotonic* if $\pi(M) \leq \pi(N)$ for all $M, N$ with $M \leq N$. In general we do not require idempotency because we do not assume exhaustive reasoning. However, for correct functioning we always assume that the deductive closure is an upper bound. By $\text{dc}_{KB_C}(M)$ we denote the *deductive closure* of $M$ under the component $C$'s knowledge base $\text{KB}_C$; i.e., the partial model just containing all (literal) information derivable, under exhaustive inferences, from the information (the literals true) in $M$ and $\text{KB}_C$. For any component $C$ we assume that in the meta-information states also information on the history of the inferences (dependencies between literals) is maintained: a simple book-keeping system that can be used for truth maintenance purposes. Based on this we assume for any $C$ a simple standard *update function* $\pi_{\text{upd}} : \text{IS}(C) \rightarrow \text{IS}(C)$ is given. The idea is that this update function simply retracts all literals that are not supported anymore because they depend on updated literals. Moreover, it updates object and meta-information states with respect to each other. For more details, consult [12].

**Definition 5.2 (transition functions)**

Let $\text{MAS}$ be a multi-agent system and let $A, B$ be agents in $\text{MAS}$ or the material world $\text{MW}$.

a) A *transition function for an internal component* $C$ (or *internal component transition function*) of $A$ is a function transforming information states for $C$ to information states for $C$; i.e., a mapping

$$\pi : \text{IS}(C) \rightarrow \text{IS}(C)$$

For correct functioning of $\text{MAS}$ it is required that $\pi = \pi_{\text{inf}} \pi_{\text{upd}}$ for mappings $\pi_{\text{upd}}$, $\pi_{\text{inf}} : \text{IS}(C) \rightarrow \text{IS}(C)$.
4.2.2 Execution of an action

The execution of an action entails an interaction from an agent A to the world (and not back: if A wants to check the resulting information it should perform an observation afterwards). We assume that the agent A decides whether an action is currently required and which one in particular (this takes place in generating actions). If an action is required, the necessary information (i.e., on which action: \textit{selected\_action(\alpha)}) is transferred from generating actions to world execution control (type O-O). After that the world is active in processing this action. The component world execution control translates the information \textit{selected\_action(\alpha)} (on which action is required) into the properties of the world (assumed to be represented by atomic facts a, b, c) that result (the effect of the action: e.g., \textit{affected(a, true), affected(b, false), affected(c, true)}). Next, it transforms them by an internal interaction into new facts of the world state (a transformation of type O-A to the component current world state). Since world execution control may need information on the current world state, we also assume that there is an internal interaction from current world state to world execution control (of type E-O).

4.3 Communication between agents

We view communication as the supply of information by one agent to another, i.e., executing a semantic link. We discuss a common pattern of communication between agents: agent A requests information from agent B and gets a response from B. Let us assume that the agent A (in the component generating questions in the domain “A on communication”) made three decisions: \textit{whether} communication is required, \textit{which} information is needed and \textit{from which} agent it may be obtained.

If communication with agent B is required, the relevant information (which observation - \textit{requested(v)}) is transferred from A to B: an interaction between generating questions of A and receiving questions of B (type O-O) is initiated.

Agent B translates (within receiving questions) the information \textit{requested(v)} into the corresponding atoms (\textit{required(a), required(b)}) and transforms them by an internal interaction into goals a, b of the component world state analysis of B (type O-T). Finally, an interaction from world state analysis of B (back) to world state analysis of A is performed by transferring the information of the truth values of the goals a, b to A (a transformation of type O-O) from the world state.

Similar to the case of observation, an alternative way back is possible here. First transform within agent B the information to receiving questions in B (an E-O transformation) and then by a second transformation from there to generating questions in A (an O-O transformation). Finally, A can decide to internally transform the information to world state analysis in A by an internal interaction (transformation of type O-A).

5 On formal semantics of compositional multi-agent systems

In this section we sketch the semantics of a compositional multi-agent system. This semantics gives a complete account of the behaviour of such a system. It is based on information states, transitions between them and traces. An overall information state of
4.1.2 Compositional structure of an example agent

In order to design our agent, its (sub)tasks, represented by components, have to be identified. The knowledge they use is classified according to the five generic domains introduced in Section 2.

The component world state analysis contains knowledge on the domain "Material world". Based on knowledge from "Mental world of the agent itself" we have three components: epistemic state, assumption generation and supervisor. No tasks make use of knowledge from "Mental world of other agents" (for an example covering this, see [3]: the wise men's puzzle). The domain "Interaction with the material world" is used in two components generating observations and generating actions. Finally, "Communication with other agents" is used in two components generating questions and receiving questions. Some of the internal semantic links are depicted in Fig. 1. Here the transformation from world state analysis to the epistemic state is of type E-O (an upward reflection). The semantic link between epistemic state and generating assumptions is of type O-O, while the connection from the latter component to world state analysis is of type O-A (a downward reflection). Other semantic links and their types will be described in the Sections 4.2 and 4.3.

4.2 Interaction with the material world

We describe the acts of observation and action execution in more detail. For simplicity we consider only observations and actions performed instaneously, not taking into account executions of plans of actions and/or observations (although it is quite possible to cover this case as well).

4.2.1 The act of observation

An observation activity of an agent A entails a bi-directional interaction with the world. First, we assume that the agent A decides whether an observation is currently required and which one in particular (implying what property of the world should be observed). This takes place in generating observations based on knowledge from the domain "A on interaction with the world". If an observation is required, the necessary information (i.e., selected_observation(t)) is transferred from generating observations to the world execution control component of the material world (semantic link of type O-O).

Next, the world is active in processing this observation. In world execution control the information selected_observation(t) is translated into the corresponding properties (assumed represented by atomic facts, say required(a), required(b)) of the world state; by an internal interaction this information is transformed into goals a, b of the component current world state (a transformation of type O-T). Finally, an interaction from the current world state (back) to the agent A (world state analysis) is performed transferring from the world state the information concerning the truth values of a, b to A (a transformation of type O-O).

An alternative, maybe conceptually more elegant, specification for interaction back to A can be obtained by first transforming within the world the information from current world state to world execution control (an E-O transformation) and then by a second transformation to generating observations (an O-O transformation). Finally, A can decide to internally transform the information to world state analysis: an internal interaction of type O-A.
4.1.1 Compositional structure of the material world

We distinguish two types of interactions between agents and the material world: performing observations and executing actions. In both cases the agent transfers information to the world which is expected to respond. Within the world we distinguish two components:

- **current world state**
  providing complete object level information about the current world state (the truth values of the atoms). This information will change and the agents may react on this.

- **world execution control**
  providing the input information about the required observations or actions in order to perform them. Semantically spoken, this component constitutes a meta-level with respect to the current world state.

In practical applications of multi-agent systems, a detailed description of the world is not a part of the system itself - only the interface with the world is specified.

Fig 1 The components of the example agent A
3.3 Combining agents into a compositional multi-agent system

While combining agents into a compositional multi-agent system we distinguish
semantic connectivity and control connectivity, taking into account also connections
with the (material) world.

3.3.1 Semantic links between agents
Interactions can take place not only between components of one agent (internal
interactions), but also between components of different agents (external interactions). In
the latter case communication between agents is induced. In our approach we restrict
interactions to supply of information and base them on semantic links. For example, a
request for information a from A to B amounts to the supply of meta-information
from A that it wants to know a to B.

The same applies to interactions with the material world. We specify the world in a
compositional manner, similar to an agent, structured by components where both
object and meta-information is included. For example, a directed observation from agent
A is the supply of meta-information to the world ("A wants to know ...”). The world
responds by initiating transfer of the referred information.

If an agent A concludes to execute an action, the action is actually executed by
transferring this as meta-information from A to the world component. The world
is specified in such a manner that its internal behaviour just simulates its reactions on
imposed observations and actions. In this manner both the act of communication
between agents and interactions with the world can be specified by the execution of a
semantic link.

The use of explicitly defined and executed semantic links has two important
advantages: one can precisely specify the required part of information and the moment
of execution of a connection.

3.3.2 Control connectivity between agents
For the global control of agents there are two possibilities. In the case of centralized
global control it is specified explicitly under which conditions an agent has to be active
and in these cases it is activated. In the decentralized case, agents are active all the time
and react on the information they receive from other agents. The first possibility can
already be specified in the current version of DESIRE, by means of supervisor rules as
discussed in Section 3.2.2. The second type of control is currently being developed.

4 Example architecture and patterns

In this section we present an example architecture of an agent and of the material world.
Interaction patterns are described (observation and action execution between an agent and
the world) and communication patterns (between two agents).

4.1 An example of a compositional agent and world

Each agent is specified by components realizing various tasks making use of our
taxonomy of knowledge. First, we give a short description of the structure of the
material world. Next, we design an example agent with capabilities of reasoning,
oberving, action execution and communication.
Note that, in comparison to Definition 3.3, here we have a further semantic distinction within the category of meta-facts. In DESIRE one can specify an arbitrary number (tower) of meta-levels.

Besides internal semantic links an agent has also semantic links with (components of) other agents and the material world. These external semantic links allow an external exchange of information.

### 3.2.2 Control structure within an agent

To specify coherent agent reasoning and acting patterns, we distinguish a global control component within the mental domain of the agent. We adopt the supervisor structure of DESIRE. We present it briefly (see [10]), assuming that the internal agent's processing is sequential, while different agents may proceed in parallel.

For each component it is described when and how it is activated, the latter possibly including a number of interactions necessary to provide input facts for it. When a component is activated, its exhaustiveness and target set has to be specified. This exhaustiveness type (any, any-new, every or all-possible) together with the target set specifies the goal the module has to attain in order to terminate successfully.

The activation condition for a component is described in terms of the termination status of another one: if its goal is attained, the termination status is succeeded. If it has not reached its goal, but may do so when additional input is provided (without revisions), its status is failed, otherwise, if reaching the goal needs some revision of the inputs, it is c-failed.

The global control of the system can be specified by means of supervisor knowledge. The most simple form of it is by supervisor rules of the following type.

\[
\text{if} \quad \text{termination}(\text{epistemic\_state, epistemic\_facts, succeeded}) \\
\text{then next-module}(\text{generating\_observations, facts\_to\_observe, without-requests, any}) \\
\text{and next-pre-trans}(\text{transfer\_epistemic\_info})
\]

This rule specifies that, if the component epistemic\_state has succeeded with respect to the target set epistemic\_facts, the transformation transfer\_epistemic\_info is applied, the component generating\_observations is activated, and it should try to derive any element of the target set facts\_to\_observe without additional requests.

<table>
<thead>
<tr>
<th>object</th>
<th>target</th>
<th>assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>O-O</td>
<td>O-T</td>
<td>O-A</td>
</tr>
<tr>
<td>R-O</td>
<td>R-T</td>
<td>R-A</td>
</tr>
<tr>
<td>E-O</td>
<td>E-T</td>
<td>E-A</td>
</tr>
</tbody>
</table>

Table 1 Interaction types
Definition 3.3 (semantic link)
Let C, C1 and C2 be components and x, y ∈ {obj, meta}.

a) The set of object, resp. meta-level semantic units for C is defined by
\[ SU_\text{obj}(C) = \text{At}(\Sigma_x^C) \times \{0, 1, u\} \]
A semantic link I between C1 and C2 of type <x, y> is a relation
\[ I : SU_\text{obj}(C_1) \times SU_\text{obj}(C_2) \]

b) A meta-signature assignment assigns to each semantic link I a signature called its meta-signature, denoted by \( \Sigma_\text{meta}^I \). A meta-information state for I is a partial model of signature \( \Sigma_\text{meta}^I \). By IS\text{meta}(I) we denote any given set of partial models of signature \( \Sigma_\text{meta}^I \) assigned to I, called the set of meta-information states for I.

A kind of standard example of a semantic link is when the involved sets of semantic units have a common subset SU, and the relation I for <a, tv1> ∈ SU and tv2 = tv1 ≠ u.
\[ I(<a, tv1>, <b, tv2>) \text{ if } a = b, <a, tv1> ∈ SU \text{ and } tv2 = tv1 ≠ u. \]

Note that by restricting this subset SU one can specify restricted information exchange. It is also possible that the considered sets of semantic units have an empty intersection because the atoms that are meant to refer to the same are named differently. In that case a semantic link involves a renaming (of atoms) as well. As an example, within an agent no particular name of the agent itself is needed (one can use the generic term "I" to refer to oneself in a natural manner), whereas the same information needs a proper reference if another agent considers it; e.g., the information "I am busy" in agent A can be named "A is busy" in agent B. A semantic link enables one to identify these (syntactically different, but semantically identical) statements.

Usually changes provoked by an interaction may require additional update procedures within the component of destination. It is assumed that all components have generic standard facilities for this that are executed before starting inferences (built in in the DESIRE software environment: [12]).

The data flow can be performed by transforming output of one component into input of another one according to a semantic link. As defined in Definition 3.3, this exchange of information can refer to both object and meta-level semantic units. The meta-input facts can be used to guide or influence the reasoning process at the object-level. Common examples are targets and assumptions, specifying an object output fact a component should try to derive and an input fact that has to be assumed as a belief, respectively. Other meta-facts provide information about the component’s current process state; we provide two kinds of them: epistemic facts and requests. Epistemic facts describe the truth, falsity or undefinedness of the object-level facts. A request specifies an object-level input atom that must be known in order to continue the object-level reasoning. In this way, different types of reflective architectures can be described. A semantic link connecting the meta-output of one component to the object-input of another is called an upward reflection; in this case the second component reasons about the first’s meta-level description. Analogously, a semantic link in the opposite direction is called a downward reflection. A classification of all semantic links (in terms of interaction types) built-in DESIRE is given in Table 1 (see [10], [13]).
This represents the meta-information that currently it is known in this component whether the agent is busy performing a task and that it is not known whether resources are available. Moreover, a current goal for the component is to find out whether resources are available. Often meta-information states will be complete. However, we leave open the possibility that some meta-information is undefined.

In practice the meta-signature is related to the object signature by a naming relation for the object statements as terms in the meta-language. Notice that we do not allow a meta-knowledge base and meta-reasoning within a component. Meta-reasoning with respect to a component $C$ can be specified by another component $D$ that has object-meta semantic links to $C$.

During a session of the architecture basic steps can be modelled by transitions between information states. One information state can change into another one by means of inference processes or interactions between components. By making inferences both the object and meta-information state will change. In the former case, the change is a refinement into a more complete one: we assume that inferences in one component are conservative and monotonic. In practice this specifies a component's memory: all information obtained hitherto is stored. As regards the meta-information state, changes are based on non-conservative transitions. For example, when an unknown object-level atom $a$ becomes known as a result of a derivation, the meta-statement known($a$) changes from false to true. For more details, see Section 5 (or [6], [14]). Another manner to change information states, i.e., interactions between components, will be discussed in the next section.

### 3.2 Compositional agents

Components representing subtasks and making use of knowledge from the different domains of an agent can be viewed as building blocks for the agent. The glue consists of two ingredients: semantic connectivity and control connectivity. We will discuss both in more detail.

#### 3.2.1 Semantic links within an agent

The various domains resulting from our decomposition are not completely independent: they have mutual semantic relations. This means that an atom in one domain may have an immediate connection to an atom in another one. Indeed, these semantic links create the logical basis for exchanging information between components (see also [13]). Roughly spoken an interaction from component $C_1$ to component $C_2$ causes a change in the information state of $C_2$ on the basis of information available in $C_1$. This change can mean a refinement or (in case of updates) a non-conservative modification of the information state of $C_2$. In effect it implies an extension, update or revision of the information state. Also here we assume a (relative) principle of conservation: all information that is not explicitly changed by an interaction will remain available (a specific frame assumption).

A semantic link is defined on the level of semantic units: atoms and their truth values. In principle it relates a semantic unit of a component $C_1$, defined by a pair $<a, tv_1>$ of a ground atom $a$ of one signature and a truth value $tv_1$ to a semantic unit of another component $C_2$, defined by a pair $<b, tv_2>$ with $b$ of another signature. Distinguishing an object and meta-information state for any component implies a classification of semantic links and the interactions based on them accordingly into four types.
An object information state for \( C \) is a partial model of signature \( \Sigma^\text{obj}_C \). By \( \text{IS}_o(C) \) we denote any given set of partial models of signature \( \Sigma^\text{obj}_C \): the set of (object) information states for \( C \).

c) A knowledge base for \( C \) is a set of ground formulae \( \text{KB}_C \) of signature \( \Sigma^\text{obj}_C \).

In other words, the object information state of a component \( C \) provides a repository for all domain information already generated (i.e., input and derived facts) and received as a result of interaction with other agents. For simplicity we allow only ground literals to be communicated in interactions and as the conclusions of derivations. Thus information states can be represented by sets of ground literals, or, equivalently, by partial models for the signature of the component. Knowledge bases are expressed in rule format.

An example of an object information state is

\[
< \text{I_am_performing_a_task}:0, \text{resources_available}:u >
\]

stating that currently the agent is not busy performing a task and no information is present on whether resources are available.

Recall that some of the agent domains are reflective: they refer to internal mental states of the agent itself. To achieve an adequate specification of reflective aspects we introduce the notion of combined information state for a component (for more details, see [13], [14]). A combined information state consists of two types of information:

- an object-information state: the information about the current domain state;
- a meta-information state: information about the state of the component's reasoning and knowledge.

Important points of a meta-information state are explicit information on what is currently (not) explicitly known and on current goals. Technically, it summarizes a number of descriptors characterizing these process states, for example:

- the truth value of object statement \( a \) has not been determined;
- the object statement \( h \) is considered as a goal for the reasoning process;
- the degree of exhaustiveness of the reasoning.

Distinguishing between meta- and object level information, which is crucial in our approach, allows us to make a component \( C \)'s reasoning process subject of the reasoning of another component \( D \). If a component \( D \) reasons at a meta-level with respect to \( C \), the current meta-information state of \( C \) provides a domain state for \( D \).

Formally, for each component \( C \) we introduce a meta-signature \( \Sigma^\text{meta}_C \) in order to define a meta-information state as a truth assignment to its ground atoms. The object and the meta-information state together specify a more complete description of the state the component is in.

**Definition 3.2 (meta-signature and information state)**

A meta-signature assignment assigns to each component \( C \) a signature called its meta-signature, denoted by \( \Sigma^\text{meta}_C \). A meta-information state for \( C \) is a partial model of signature \( \Sigma^\text{meta}_C \).

By \( \text{IS}_m(C) \) we denote any given set of partial models of signature \( \Sigma^\text{meta}_C \) assigned to \( C \), called the set of meta-information states for \( C \).

An example of a meta-information state is

\[
< \text{known(I_am_performing_a_task)}:1, \\
\text{known(resources_available)}:0, \\
\text{goal(resources_available)}:1 >
\]
(e) Communication with other agents
The agent may need knowledge to determine what communications are possible and useful to obtain additional information together with the information on the availability of agents for communication.

Within each of these five domains some subdomains can be distinguished. The depth of these distinctions is a matter of choice (related to the type of agent being created).

3 Compositional Multi-Agent Systems

Knowledge and reasoning capabilities referring to the domains of an agent's interest should be specified in some way. In this section we introduce a specific type of architecture for multi-agent systems, called compositional architecture. The general idea is that each complex agent is composed from (primitive) components that describe subtasks to be performed. Components are connected to each other according to predefined types of semantic links. Each component has a simple logical description and makes use of specific knowledge. Complex behaviour covering both reasoning and acting can be obtained by nontrivial (and dynamic) patterns of interaction between components. In a similar manner the system as a whole can be composed from its agents. In the Sections 3.1, 3.2 and 3.3 we introduce the notions of component, compositional agent and compositional multi-agent system, respectively.

3.1 Agent components

We describe the domain of a component \( C \) by a signature \( \Sigma_C \): a lexicon of basic language elements in terms of propositional or many-sorted predicate logic. Using this signature we can define:
- possible states of the domain as a set of truth assignments to its ground atoms;
- the knowledge described in the component.

We distinguish the component's knowledge base (the static part) and the facts base or information state (the dynamic part).

**Definition 3.1 (object signature and information state)**

a) A partial model of signature \( \Sigma \) is a mapping

\[ M : \text{At}(\Sigma) \rightarrow \{0, 1, \text{u}\} \]

where \( \text{At}(\Sigma) \) is the set of ground atoms for \( \Sigma \). An atom \( a \) is true in \( M \) if \( 1 \) is assigned to it, and false if \( 0 \) is assigned; otherwise it is called undefined or unknown. A partial model \( M \) is called complete if no \( M(a) \) equals \( \text{u} \). The partial model \( N \) is called a refinement of \( M \), denoted by \( M \leq N \), if for all atoms \( a \) it holds: \( M(a) \leq N(a) \), where the (partial) ordering on truth values is defined by \( \text{u} \leq 0, u \leq 1, u \leq u, 0 \leq 0, 1 \leq 1 \).

b) An object signature assignment assigns to each component \( C \) a signature called the object signature of \( C \), denoted by \( \Sigma_C \) or \( \Sigma_C^{\text{obj}} \).

A domain state for \( C \) is a complete model of signature \( \Sigma_C^{\text{obj}} \). By \( \text{DS}_\alpha(C) \) we denote any given set of complete models of signature \( \Sigma_C^{\text{obj}} \) (assigned to \( C \)), called the set of domain states for \( C \).
a time-dependent definition of an overall structure of an agent's knowledge. This knowledge can be treated as a collection of knowledge states for each type of knowledge involved, called *(dynamic) information state* or facts base. In contrast to the time-dependent part of the information, for each type of knowledge we distinguish an invariant part (holding in all states), called *(static) knowledge base*. Thus one can reason about the current state as well as the past and (possible) future states and their connections. We briefly explain the generic domains in more detail.

**(a) Material world**

This knowledge describes the current world state: material (e.g., physical) aspects of the world, including the agents. This (object) knowledge constitutes the basis for drawing conclusions about the current world state. Also (temporal) knowledge about past and possible future states and their connecting processes (events, actions) can be covered, for example for the sake of planning.

**(b) Mental world of the agent itself**

This kind of knowledge is essential from the (meta-level) perspective of guiding complex patterns of reasoning and acting. It can take many forms:

- **Strategic control knowledge** about the agent's current goals and reasoning and acting processes, guiding the agent in its behaviour; i.e., helping it in achieving its goals, to follow heuristics. In particular, if the agent lacks some information it has to decide whether this information can be acquired by making observations from the material world, communications with other agents, introducing assumptions - beliefs. Sophisticated strategic knowledge enables an agent to perform flexible behaviour.

- Speaking about agent’s knowledge we also take into account its possibility of *introspection*. Introspection, i.e., the process of examining one’s own beliefs, has two interesting sides. The *positive introspection* principle assumes that an agent knows what it knows. Analogously, the *negative introspection* principle states that an agent knows what it does not know. In our approach we consider both types of introspection useful. We call this part of an agent’s knowledge *epistemic knowledge*. This kind of knowledge should contain (among others) explicit information about qualifications of the agent's knowledge (e.g., observed, assumed, derived, based on ...).

- If it turns out hard or impossible to acquire necessary information, there still remains the possibility to rely on beliefs only. Beliefs can be generated by generating *additional assumptions*. The agent needs (defeasible) knowledge to do this.

**(c) Mental world of other agents**

Sometimes one agent has knowledge about some aspects of the mental state of another one and can reason about it; for example knowing another agent’s goals or past reasoning processes.

**(d) Interaction with the material world**

Observations are usually an important source of additional information. To determine which ones are possible in the current world state specialized knowledge may be needed. In taking the decision another factor should be considered: the costs or effort of an observation. Observations form one kind of interaction with the material world. Another one is created by actions considered for execution in the material world (to actually change the world state).
patterns of behaviour. The research reported here has also resulted in identifying a number of features to be added.

The paper is organized as follows. In Section 2 we discuss the generic domains of an agent that can be distinguished according to the types of knowledge to be dealt with. In Section 3 we show how a complex agent can be composed from components that represent subtasks, using specific composition principles defining semantic links between them in a formal, standardized manner. In Section 4 we describe an example architecture created in this manner and some patterns of reasoning and acting. In Section 5 we sketch formal semantics of our approach. Finally, in Section 6 we evaluate our investigations and end up with a number of issues for further research.

2 Distinguishing generic domains of agent knowledge

Since we consider an agent as a rather complex reasoning and acting entity, our first concern is a transparent structure for it. Solving a problem by an agent can be viewed as an alternation of reasoning and other activities:

- actively gather observational information from the material world;
- draw conclusions from this information;
- initiate and carry out communication with other agents;
- generate beliefs by making additional (defeasible) assumptions;
- change the world by executing actions.

Individual agents make local decisions: each of them pursues activities interesting from its local viewpoint. These viewpoints may vary from one agent to another, and the compatibility of the individual viewpoints determines whether agents are interested in cooperation, competition, or merely coexistence. In order to cooperate, they must recognize that cooperation is in their self-interest. Thus, they must have local knowledge guiding them into cooperation. Speaking about agent knowledge we implicitly assume basic reasoning capabilities related to it.

In this section different types of agent knowledge are distinguished. The first, basic type of knowledge to be distinguished is the knowledge about the material world. As this knowledge is very broad, it needs to be considered in more detail. The core of this knowledge is a description of the external material world, but as a special part of the world's entities material aspects of agents (including itself) should be distinguished. Secondly, in view of possible cooperation, each agent is expected to have (reflective) knowledge about mental aspects of itself and other agents. Thirdly, knowledge allowing communication with other agents and interaction with the external world (directing and carrying out observations, communications and actions) is needed. A combination of various types of knowledge constitutes a proper basis for making decisions. To promote a cleaner separation of concerns, we classify agent’s knowledge in the following way:

(a) Material world
(b) Mental world of the agent itself
(c) Mental world of other agents
(d) Interaction with the material world
(e) Communication with other agents

Another essential point is the dynamics and incompleteness of knowledge: at each moment in time only a limited amount of information has been acquired. This suggests
Compositional Formal Specification of Multi-Agent Systems

Barbara Dunin-Keplicz and Jan Treur

1University of Warsaw, Institute of Informatics
ul. Banacha 2, 02-097 Warszawa Poland
Email: keplicz@mimuw.edu.pl

2Free University Amsterdam, Dept. of Mathematics and Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam The Netherlands
Email: treur@cs.vu.nl

Abstract
In this paper it is investigated how multi-agent systems with complex agents can be designed and formally specified based on the notion of a compositional architecture. After identifying the types of knowledge required for an agent we formally define a general multi-agent system. Moreover, a specific type of agent with various capabilities of reasoning and acting is given. Some essential patterns of integrated reasoning, communication and interaction with the material world are described. Finally, we present an overview of formal semantics for our approach.

1 Introduction
Research on multi-agent systems focuses on the one hand on theoretical foundations, sometimes not too strongly related with implementable applications, and on the other hand on specific architectures and programming languages, and environments for applications (e.g., see [1], [7], [8], [11]). According to our view a clear methodology including a specification method for the development of complex multi-agent systems is needed. In this paper we study the possibilities for the development of a high level formal specification language for multi-agent systems. The structure underlying our language is based on the notion of a compositional architecture (see [6], [9]) allowing to specify complex agents in a transparent manner as well as to integrate reasoning and acting in one (declarative) logical framework. We aim at a type of agent that is able to behave in a flexible manner using a sophisticated local structure and control allowing it to make its own decisions about subproblems to solve and subproblems or solutions to communicate.

As a point of departure we take a formal specification language for compositional architectures: DESIRE (framework for DEsign and Specification of Interacting REasoning components; see [9], [10]). This framework turns out to cover major parts of the specification of a multi-agent system. First, it is easy to design different variants of agents. Second, one can completely specify the dynamics of reasoning patterns and acting behaviour. The executability of DESIRE specifications (there exists a software environment realizing this), gives the ability to carry out experiments with (alternative)

This work was part of the ESPRIT III Basic Research project 6156 DRUMS II.