Chapter 5

A Semantic Structure for Agent Dynamics

This chapter presents the semantic structure for agent dynamics. Section 5.1 formally defines the constructs provided by the semantic structure, focusing on static aspects of the constructs. The dynamic aspects are presented in Section 5.2. Section 5.3 summarises the semantic structure defined so far. Proofs of propositions presented in this chapter are given in Section 5.4.

5.1 The Constructs of the Semantic Structure

The main constructs provided by the semantic structure are components and information links. As stated in Section 2.1, three aspects of the first construct (components) are distinguished: the information state (i.e., the information contents of a component), the interfaces and the compositional structure in terms of subcomponents and links. The first two aspects are formally defined in Section 5.1.1. The second construct (information links) is defined in Section 5.1.2. The third aspect of a component, composition structure, is formally defined in Section 5.1.3.

5.1.1 Components, Interfaces and State

Assume that a set of components *Comp* (or, more precisely, component identifiers or names) is given. Elements of this set are typically denoted by capitals *C*, *D*, etc. As stated in Chapter 2, a dynamically changing information state is distinguished for each component. Each information state consists of input, internal and output substates. The input and output substates of an information state define the interface of a component, in the sense that these substates are accessible to other components (and the component itself) via information links. The current section assumes that for each component *C*, three non-empty sets of states $\mathcal{S}_{C,in}$, $\mathcal{S}_{C,int}$ and $\mathcal{S}_{C,out}$ are given (for the input, internal and output substates, respectively), without further commitment to the contents of these sets. (Chapter 9 provides a formal language to specify these sets.) The (overall) state of a component *C* (the first aspect of a component distinguished in this thesis) is composed of elements of the sets $\mathcal{S}_{C,in}$, $\mathcal{S}_{C,int}$ and $\mathcal{S}_{C,out}$ as follows:

Definition 5.1. (Component information state). *Let C be a component. An* information state *of C is an element of* $\mathcal{S}_{C,in} \times \mathcal{S}_{C,int} \times \mathcal{S}_{C,out}$. *The set of all component information states of C is denoted* $\mathcal{S}_C$, *i.e.,* $\mathcal{S}_C = \mathcal{S}_{C,in} \times \mathcal{S}_{C,int} \times \mathcal{S}_{C,out}$.

Example 5.2. In the running example, the following sets are used for the user and broker agents. (As explained in Section 4.2.2, the user agents and the broker agent use different ontologies to describe resources. Therefore, two sets of ontology terms OT_1 and OT_2 are assumed to be given. (These set can, for instance, be sets of resource descriptions in the RDF (Lassila, 1998) format.) The set Q is a set of query terms, *Users*={user_1,user_2} and *Providers*={provider_1,provider_2}. The user agents and the broker agent also maintain information about their own processes. For instance, the input interface of the user agents can be in a state in which it is ready to receive information. This is taken into account in the definition of $\mathcal{S}_{\text{user\_1},in}$ and $\mathcal{S}_{\text{user\_2},in}$ below. User agents and the broker agent may also internally maintain beliefs. For instance, the broker agent maintains beliefs about matches between user queries and information available via the providers.)

- $\mathcal{S}_{\text{user\_1},in} = \mathcal{S}_{\text{user\_2},in} = \{\text{communicated\_by}(t,\text{broker}) \mid t \in OT_1\} \cup \{\text{ready\_for\_information}\}$,
  $\mathcal{S}_{\text{user\_1},int} = \mathcal{S}_{\text{user\_2},int} = \{\varnothing\}$
  $\mathcal{S}_{\text{user\_1},out} = \mathcal{S}_{\text{user\_2},out} = \{\text{to\_be\_communicated\_to}(q,\text{broker}) \mid q \in Q\}$.
- $\mathcal{S}_{\text{broker},in} = \{\text{communicated\_by}(q,u) \mid q \in Q \text{ and } u \in Users\} \cup \{\text{communicated\_by}(t,p) \mid t \in OT_2 \text{ and } p \in Providers\}$,
  $\mathcal{S}_{\text{broker},int} = \{\text{belief(match}(t,q)) \mid t \in OT_2 \text{ and } q \in Q\}$
  $\mathcal{S}_{\text{broker},out} = \{\text{to\_be\_communicated\_to}(t,u) \mid t \in OT_2 \text{ and } u \in Users\} \cup \{\text{just\_communicated\_to}(t,u) \mid t \in OT_2 \text{ and } u \in Users\}$.

In this example, states are identified by elements of e.g. $\mathcal{S}_{\text{user\_1},in}$ such as communicated_by(t,broker), where t is an element from the set of ontology terms OT_1. The elements of sets such as $\mathcal{S}_{\text{user\_1},in}$ resemble propositions about states. In fact, in Chapter 9, a logical language for propositions that describe states is presented. However, in this example, elements of sets such as $\mathcal{S}_{\text{user\_1},in}$ are (unique) names of states. These names have no internal structure. ■

As stated in Section 2.1, two kinds of components are distinguished: composed components and primitive components. The component information state definition given above applies to both kinds of components. In this thesis, the term 'internal (sub)state' is used in a restricted way: it refers to the state of the part of a component's information contents that is not visible to other components. In particular, the terms 'internal (sub)state' and 'internal information' do not refer to, and are thus independent of, the (input and output) states of the subcomponents and links of a composed component. In other words, the contents of a composed component consists of subcomponents, links and the internal information contents as defined above.

5.1.2 Information Links

The second construct provided by the semantic structure is the information link. An information link transmits information from one component (called source component or *domain* of the link) to another, or possibly the same component (called the destination component or *co-domain* of the link). In this section, static aspects of this construct are formalised. As explained in Section 2.2.3.1, in the semantic structure, information links are first-class citizens; they are of the same standing as components.

It is assumed that a set of links *Lnk* (or, more precisely, link identifiers or names) is given. An element of this set is typically denoted with the capital *I*. The first static aspect identified is the *state* of the link (See Section 2.2.3.2). The state of a link is determined by (1) the state of information transmission as an activity: for instance, a link can be busy exchanging information, it can be waiting for new information to exchange, it can be enabled or disabled, and (2) the contents of the link, e.g. messages in transit. The semantic structure does not enforce a commitment with respect to the contents of link states, nor to the way in which the state of the link is described. Instead, for each information link *I*, a non-empty set of information link states $\mathcal{S}_I$ is assumed to be given.

Definition 5.3. (Link information state). *Let I be an information link. An* information state *of I is an element of a set* $\mathcal{S}_I$.

Example 5.4. In the running example, a link called broker_to_user_1 exists between broker and user_1. The set of link information states of this link is defined as follows (where *t* is a set of ontology terms):

$$\mathcal{S}_{\text{broker\_to\_user\_1}} = \{\text{awake\_and\_empty}, \text{active\_and\_contents}(t) \mid t \in OT_2\}$$ ■

The second static aspect of an information link is its *compositional relation* with components in a compositional system. In this section, the connection of a link with two components or links, one at each ‘end point’, is formalised. A more extensive definition of the compositional relation of a link is given in the next section.

Definition 5.5. (Domain and co-domain). *Let I be an information link. Two components or links, called the* domain *and* co-domain, *are relative to I. This is denoted by two functions, dom,cdom: Lnk→Comp∪Lnk. A link I with dom(I)=*S_1 *and cdom(I)=*S_2 *is called a link from* S_1 *to* S_2.

The third static aspect of an information link is the *information link mapping.* As explained in Section 4.1.2, an information link mapping is a relation between states from the state sets of the domain and co-domain of the link. In fact, an information link mapping is defined as an 8-ary relation to fulfil the commitments made in

Section 2.2.3.3, Section 2.2.5 and Section 2.2.6 (this is further explained below). As stated in Section 2.2.1, the semantic structure distinguishes six kinds of links. The information link mapping differs for each of the six kinds of links as follows:

Definition 5.6. (Information link mapping). *Let I be an information link. An information link mapping for I is a relation defined as follows:*

- $\lambda_I \subseteq (\mathcal{S}_{dom(I),out} \times \mathcal{S}_{dom(I),out}) \times \mathcal{S}_I^4 \times (\mathcal{S}_{cdom(I),in} \times \mathcal{S}_{cdom(I),in})$, *if I is a private link, or*
- $\lambda_I \subseteq (\mathcal{S}_{dom(I),in} \times \mathcal{S}_{dom(I),in}) \times \mathcal{S}_I^4 \times (\mathcal{S}_{cdom(I),in} \times \mathcal{S}_{cdom(I),in})$, *if I is an import mediating link, or*
- $\lambda_I \subseteq (\mathcal{S}_{dom(I),out} \times \mathcal{S}_{dom(I),out}) \times \mathcal{S}_I^4 \times (\mathcal{S}_{cdom(I),out} \times \mathcal{S}_{cdom(I),out})$, *if I is an export mediating link, or*
- $\lambda_I \subseteq (\mathcal{S}_{dom(I),in} \times \mathcal{S}_{dom(I),in}) \times \mathcal{S}_I^4 \times (\mathcal{S}_{cdom(I),out} \times \mathcal{S}_{cdom(I),out})$, *if I is a cross-mediating link, or*
- $\lambda_I \subseteq (\mathcal{S}_{dom(I),out} \times \mathcal{S}_{dom(I),out}) \times \mathcal{S}_I^4 \times (\mathcal{S}_{cdom(I)} \times \mathcal{S}_{cdom(I)})$, *if I is a link modifier link, or*
- $\lambda_I \subseteq (\mathcal{S}_{dom(I)} \times \mathcal{S}_{dom(I)}) \times \mathcal{S}_I^4 \times (\mathcal{S}_{cdom(I),in} \times \mathcal{S}_{cdom(I),in})$, *if I is a link monitoring link.*

The intended meaning of an information link mapping is explained with reference to Figure 2.7. In Figure 2.7, information transmission by a link *L* from a component *A* to a component *B* is depicted by eight states. (Thus, *dom(L)=A* and *cdom(L)=B.*) These eight states correspond to an element $\langle\langle v_{A,i}; v_{A,j}\rangle; \langle v_{L,i''}; v_{L,j''}; v_{L,k}; v_{L,l}\rangle; \langle v_{B,i'}; v_{B,j'}\rangle\rangle \in \lambda_L$. (In this thesis, tuples are delimited by angular brackets and their elements are separated by semicolons.) This element states that (the numbers between parentheses refer to the explanation below):

If component *A* reaches state $v_{A,i}$ (1), *and* link *L* is in state $v_{L,i''}$ (2), *and* component *B* is in state $v_{B,i'}$ (3),

then component *B* should reach state $v_{B,j'}$ (4) as one of the successors of $v_{B,i'}$ (5), and one of the following states of *A* should be $v_{A,j}$ (6),

and state $v_{L,j''}$ should be the first state of *L* in which the information is in transit, *and* state $v_{L,k}$ should be the last state of *L* in which the information is in transit, *and* state $v_{L,l}$ should be the first state of *L* in which the information is no longer in transit (7).

Parts (1) and (4) form the most important part of this expression and as such are part of the informal explanation in Section 4.1.2. Parts (1) and (4) show that in the semantic structure, information transmission is characterised in terms of states of the components that exchange information. Condition (2) is related to the commitment presented in Section 2.2.3.2: state $v_{L,i}$ enables the expression of a requirement on the state of the information link to enable transmission, e.g., the expression that the link must be in an enabled state. Condition (3) is related to the commitment presented in Section 2.2.6 and enables expression of an enabling condition for receipt imposed by the destination component. Parts (5) and (6) reflect the commitments to non-blocking receive (Section 2.2.6) and send (Section 2.2.5), respectively. Part (7) enables expression of the dynamics of transmission as a process as explained in Section 2.2.3.3. The sequence of states can, for instance, be used to describe that a message is put in the link (change from $\nu_{L,i''}$ to $\nu_{L,j''}$) and later taken from it (change from $\nu_{L,k}$ to $\nu_{L,l}$). An information link mapping does not, in general, need to be functional in all of its arguments, and is therefore formalised using a relation.

Example 5.7. The information link mapping of the link broker_to_user_1 is defined as follows (where *trans* is a function from OT_2 to OT_1 that translates ontology terms in OT_2 to OT_1, which is assumed to be given):

$$\lambda_{\text{broker\_to\_user\_1}} = \{\langle\langle \text{to\_be\_communicated\_to}(t,\text{user\_1}); \text{just\_communicated\_to}(t,\text{user\_1})\rangle; \langle \text{awake\_and\_empty}; \text{active\_and\_contents}(t); \text{active\_and\_contents}(t); \text{awake\_and\_empty}\rangle; \langle \text{ready\_for\_information}; \text{communicated\_by}(t',\text{broker})\rangle\rangle \mid t \in OT_2,\ t' \in OT_1 \text{ and } t'=trans(t)\}.$$

This information mapping specifies that if in broker's behaviour, there is a state to_be_communicated_to(*t*,user_1), and the state of link broker_to_user_1 is awake_and_empty, and in user_1's behaviour, there is a state ready_for_information, then a transition of user_1's input interface state to the state communicated_by(*t'*,broker) exists, for resource description terms such that *t'=trans(t)*. Moreover, in broker's behaviour, one of the successor states is the state just_communicated_to(*t*,user_1), and the state of link broker_to_user_1 changes to active_and_contents(*t*) and then back to awake_and_empty. (To keep the example simple, it is assumed that link broker_to_user_1 can only transmit a message if no other messages are in transit.) ■

An information link mapping provides a static description of the relation between two components and a link as a result of information transmission. The (informal) meaning of an information link description is made more precise in Chapter 6. In Chapter 6, the actual behaviour of two components and a link is related to the intended behaviour as described by the information link mapping. In other words, Chapter 6 discusses the relation between states that *actually occur* in the behaviour of the destination and source components of a link and the link itself, and the relation between *possible states* of these components described by an information link mapping. In Chapter 5, information link mappings are only used in examples of compatibility relations.

Information links provide flexibility in information transmission in three ways. In the first place, it is possible to specify how states of the two components and the link are connected. As the state of components and links is determined by their information contents, expressed in their 'own' terms, information transmission enables translation of the terms used for different components. (In the running example, this facility is used to translate the ontology used by the broker agent to the ontology used by the user agents, and vice versa.) In the second place, using the link state, the process of information transmission can be modelled in great detail. In the third place, it is possible to specify results of information transmission (in the example: state just_communicated_to(*t*,user_1)), and enabling conditions in the destination component of a link (in the example: state ready_for_information) and in the link itself.

5.1.3 Compositional Structures

Section 2.1 names the composition structure of a component as its third aspect. This aspect describes compositions of components in terms of the subcomponents and information links that constitute the components. The structural aspect is very general: starting from a set of components *Comp* (or, more precisely, component names) and a set of links *Lnk* (or, link names), arbitrary recursive composition structures, called *structure hierarchies,* can be described. It is not assumed that for a given set of components and a given set of links, there is only one structure hierarchy, nor is there any commitment with respect to whether components are composed or primitive. Thus, different perspectives on a set of components and a set of links can be described. For instance, in a certain stage in the analysis or development of a multi-agent system, certain components may be considered primitive. (They do not have subcomponents). In a later stage, these subcomponents may become composed. The qualifiers 'composed' and 'primitive' are thus related to a given structure hierarchy, as is the notion of being a subcomponent.

While the semantic structure developed in this thesis does not assume that the structure hierarchy of a component is unique, it is assumed that many applications of the semantic structure have a certain unique structure for each component. For instance, a specification framework for multi-agent systems may provide facilities to describe the compositional structure of a component. Such a description may induce a unique structure hierarchy for the component. Even if this is the case, different refinements of specifications most often result in different structure hierarchies for the same component. The formal definition of a structure hierarchy is as follows:

Definition 5.8. (Structure hierarchy). *A* structure hierarchy *SH is a tuple* $\langle Comp;Lnk;\prec;dom;cdom\rangle$, *where:*

- *Comp is a finite set of component names;*
- *Lnk is a finite set of information link names such that* $Comp \cap Lnk = \emptyset$;
- $\prec \subseteq (Comp \cup Lnk) \times Comp$ *(the hierarchy relation) such that* $\prec$ *defines a forest: a finite, non-empty collection of trees. For all pairs* $\langle I;C\rangle \in \prec$ *such that* $I \in Lnk$, *I must be a leaf. The reflexive closure of* $\prec$ *is denoted* $\preceq$;

- *dom,cdom: $Lnk \rightarrow Comp \cup Lnk$ are total functions (the domain and co-domain functions) such that for all $I \in Lnk$, if $dom(I)=S_1$ and $cdom(I)=S_2$, then either*
  - *$S_1,S_2 \in Comp$ and there is a $P \in Comp$ such that $S_1,S_2 \prec P$ (private link), or*
  - *$S_1,S_2 \in Comp$ and $S_2 \prec S_1$ (import mediating link), or*
  - *$S_1,S_2 \in Comp$ and $S_1 \prec S_2$ (export mediating link), or*
  - *$S_1,S_2 \in Comp$ and there is a $P \in Comp$ such that $I,S_1,S_2 \prec P$ and $S_1=S_2$ (cross-mediating link), or*
  - *$S_1 \in Comp$ and $S_2 \in Lnk$ and there are $S_3,S_4,P \in Comp$ such that $S_3 \neq S_1$, $S_4 \neq S_1$, $P \neq S_i$ and $S_i \prec P$ for $i=1,\ldots,4$ and $dom(S_2)=S_3$ and $cdom(S_2)=S_4$, (link modifier link), or.*
  - *$S_1 \in Lnk$ and $S_2 \in Comp$ and there are $S_3,S_4,P \in Comp$ such that $S_3 \neq S_2$, $S_4 \neq S_2$, $P \neq S_i$ and $S_i \prec P$ for $i=1,..,4$ and $dom(S_1)=S_3$ and $cdom(S_1)=S_4$, (link monitoring link).*

A structure hierarchy is called a structure hierarchy for a component $C \in Comp$ iff $\prec$ defines a collection of exactly one tree (i.e., $\prec$ is connected, formally $\forall C_1,C_2 \in Comp$: $C_1 \prec C_2 \vee C_2 \prec C_1$) and C is the root of the tree defined by $\prec$, thus $\neg\exists C' \in Comp$: $C \prec C'$. If $Comp=\emptyset$, then the structure hierarchy is called empty.

In the subcomponent relation, $C_1 \prec C_2$ denotes that C_1 is a subcomponent of C_2 in *SH*. A component *C* is called primitive in $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ iff there is no $C' \in Comp$ such that $C' \prec C$. Otherwise, it is called composed in *SH*. Components which are leaves in a structure hierarchy (that is, components *C* in the structure hierarchy for which there is no *C'* such that $C' \prec C$) are by definition primitive in *SH*.

In general, a structure hierarchy for a component *C* not only contains subcomponents of *C*, but also subcomponents of these subcomponents, and so on. Therefore, a structure hierarchy itself comprises more than the compositional structure (the third aspect of a component distinguished by the semantic structure) of a component *C*. A structure hierarchy consisting of *C*, its subcomponents and links is called the *composition structure* of *C*. Formally:

Definition 5.9. (Composition structure).

- *A* composition structure *for a component C is a structure hierarchy $CS=\langle Comp;Lnk;\prec;dom;cdom\rangle$ for C such that for all $S \in Comp \cup Lnk$, $S \prec C$.*
- *Let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a structure hierarchy and let $C \in Comp$ be a component. The* composition structure *CS(C,SH)* of *C* with respect to *SH is the structure hierarchy $\langle Comp';Lnk';\prec';dom';cdom'\rangle$ where:*
  - *$Comp'=\{ C' \in Comp \mid C' \preceq C \}$;*
  - *$Lnk'=\{ I \in Lnk \mid I \prec C \}$;*
  - *$S \prec' C \Leftrightarrow S \prec C$, $S \in Comp' \cup Lnk'$ and $C \in Comp'$;*
  - *For all $I \in Lnk'$, $dom'(I)=dom(I)$;*
  - *For all $I \in Lnk'$, $cdom'(I)=cdom(I)$.*

If, for a structure hierarchy *SH* for *C* it holds that *SH=CS(C,SH)*, then *SH* itself is a composition structure for *C*. For a given component *C* and structure hierarchy *SH*, the composition structure $CS(C,SH)=\langle Comp';Lnk';\prec';dom';cdom'\rangle$ is unique. The set of subcomponents of *C* with respect to $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ is denoted $Subc(C,SH)=\{C'\in Comp\,|\,C'\prec C\}$. The set of links of *C* with respect to *SH* is denoted $Lnk(C,SH)=\{I\in Lnk\,|\,I\prec C\}$. The set of subcomponents of *C* with respect to *SH* and links '*inside*' *C* is denoted *SLC(C,SH)*: $SLC(C,SH)=\{C\}\cup Subc(C,SH)\cup Lnk(C,SH)$. (The abbreviation 'SLC' stands for '*Subc*, *Lnk* and the Component itself', but also for 'slice'.) If a component *C* is a primitive component according to a structure hierarchy *SH* for *C*, then, according to the definition, $Subc(C,SH)=\varnothing$.

Example 5.10. To illustrate structure hierarchies, a structure hierarchy for the compositional system depicted in Figure 4.5 is given. For conciseness, the structure hierarchy only contains the components toplevel, user_1, broker, ASP and OPC. The following structure hierarchy can be used to analyse user_1 together with agent broker: $sh=\langle Comp;Lnk;\prec;dom;cdom\rangle$, with:

- *Comp*={toplevel,user_1,broker,ASP,OPC};
- *Lnk*={user_1_to_broker,broker_to_user_1};
- ≺={⟨ASP;broker⟩,⟨OPC;broker⟩,⟨user_1;toplevel⟩,⟨broker;toplevel⟩};
- *dom*={⟨user_1_to_broker;user_1⟩,⟨broker_to_user_1;broker⟩};
- *cdom*={⟨user_1_to_broker;broker⟩,⟨broker_to_user_1;user_1⟩}. ■

5.2 Dynamics

This section focuses on how the dynamics of a compositional system is described using the constructs provided by the semantic structure. First, Section 5.2.1 discusses the description of the local dynamics of components. Section 5.2.2 discusses the description of the dynamics of composition structures.

5.2.1 Local Dynamics

The information state of a component changes over time, which is modelled using sequences of information states called traces. Separate local traces of the input, internal and output substates are distinguished, making it possible for applications of the semantic structure to focus on one or more of the interfaces of a component.

Definition 5.11. (Time frame). *A* time frame *is a pair* $TF=\langle T;<\rangle$, *where T is a set of time points and < is a strict partial order on T. Moreover, < is connected, i.e.* $\forall t\in T\!:\ \exists t'\in T\!:\ t<t' \vee t'<t$. *There is one element,* $\bot\in T$, *for which there is no* $t\in T$ *such that* $t<\bot$.

This definition enables various types of time frames to be used in the semantic structure, such as, for instance, branching time frames and dense time frames. In the remainder of this thesis, linear time frames are assumed, unless explicitly states otherwise. (For example, the discussion of dense time frames in Chapter 7). It is, however, possible to use different time frames with different properties for different components in a compositional system.

Definition 5.12. (Local trace). *Local traces are defined as follows:*

- *A* local input trace *of a component C for a time frame* $TF=\langle T;<\rangle$ *is a pair* $LT_{C,in}=\langle TF;V\rangle$, *where V is a total function* $V: T \rightarrow \mathcal{S}_{C,in}$. *The* internal and output traces *are defined analogously*.
- *A* local component trace *of a component C for a time frame* $TF=\langle T;<\rangle$ *is a pair* $LT_C=\langle TF;V\rangle$, *where V is a function* $V: T \rightarrow \mathcal{S}_C$.
- *The set of all local input traces of a component C is denoted* $\mathcal{LT}_{C,in}$. *The sets of all internal and output traces of a component are denoted analogously. The set of all local component traces of a component C is denoted* $\mathcal{LT}_C$.

In the previous definitions, four different notions of traces are distinguished (input, internal, output and local component traces). In the semantic structure developed in this thesis, all four traces can be associated with each component. These four traces are not independent: a local component trace consists of local component states, each of which itself consists of the input, internal and output traces of the same component. The dependencies between traces can be of several types. For instance, assume that for the input, internal and output trace of a component a discrete totally ordered time frame is used. In this case, at least three choices for the local component traces can be distinguished:

- For a local component trace, a discrete totally ordered time frame is chosen. Consider a point in time *t* and its immediate successor *t'* (which is assumed to exist here). The state at time point *t'* differs from the state at time point *t* for *all three substates* input, internal and output;
- For a local component trace, a discrete totally ordered time frame is chosen. Consider a point in time *t* and its immediate successor *t'* (which is assumed to exist here). The state at time point *t'* differs from the state at time point *t* for *two of the three substates* input, internal and output: the input and output substate differ, while the internal state remains the same;
- For a local component trace, a discrete totally ordered time frame is chosen. Consider a point in time *t* and its immediate successor *t'* (which is assumed to exist here). The state at time point *t'* differs from the state at time point *t* for *one of the three substates* input, internal and output;

If the restriction to totally ordered time frames is dropped, and instead, a local component trace with a partially ordered time frame is chosen, another alternative is possible. Consider a point in time *t* and *one* of its immediate successors *t'* (which is assumed to exist here). The state at time point *t'* differs from the state at time point *t* for *one of the three substates* input, internal and output.

These different possibilities for local component traces model different choices with respect to synchronisation of the different (input, internal and output) substates of a component. In an application of the semantic structure, a time frame and a dependence relation between the traces is chosen. (It is possible to vary these choices for different components.) The flexibility with respect to these choices is a feature of the semantic structure.

Local component traces are used to model the behaviour of components. For each component *C*, a subset of $\mathcal{LT}_C$ is distinguished which contains all local component traces that are possible behaviours of *C* from a local point of view. (It is assumed that the possible behaviours of *C* from a local point of view are given, for instance using the specification language presented in Chapter 9.) Formally:

Definition 5.13. (Local component behaviour). *Let C be a component. A* local component behaviour *$Beh_{loc}(C)$ of C is a set of local component traces of C.*

To clarify the use of the qualification 'local' in the name of the notion defined above, a distinction is made between two views on the concept of location. On the one hand, a component has a specific location in a structure hierarchy. This concept of location refers to the structure of a compositional system in terms of components and subcomponents. This structure is determined by how processes relate to one another in terms of the function of the subprocesses, as explained in Chapter 3. In this conception of the location of a component, a subcomponent is 'close', or 'local' to its parent component. On the other hand, there is also a 'physical' distribution of components (and the processes they represent) over processes, which are spatially divided. It is *not* assumed that the subcomponents of a component all exist at the same physical location. Thus, subcomponents of a component are not necessarily local to their parent component with respect to the 'physical' location of a compositional system. In this thesis, the qualifications 'local' and 'global' always refer to the physical distribution of a compositional system.

The notion of local component behaviour is called 'local' because the set $Beh_{loc}(C)$ is not constrained by non-local phenomena such as information transmission (not even with its subcomponents). A set $Beh_{loc}(C)$ is independent of any structure hierarchy in which *C* occurs. In other words, the set $Beh_{loc}(C)$ can contain local component traces that are not possible behaviour when information exchange is taken into account, for instance because these local component traces depend on information residing in other components. This is made more precise in Section 5.2.2.

Example 5.14. In the running example, the set of natural numbers together with the usual ordering is used as a time frame. A local component trace is represented as a sequence of component states with the sets of input, internal and output propositions separated by bars. To present some example traces, concrete elements of $\mathcal{S}_{\text{broker}}$ have to be given. (In a previous example, elements of $\mathcal{S}_{\text{broker}}$ were specified in an abstract way, because the sets of ontology terms OT_1 and OT_2 were not specified.) In this example, the sets OT_1, OT_2 and Q are partially specified by a few sample elements. First, the set of ontology terms OT_2 used by the information providers and the broker agent contains elements res_1 and res_2, which stand for specific resource descriptions. Thus, {res_1,res_2}$\subseteq OT_2$. Second, the set OT_1 of ontology terms used by the broker agent to transmit resource descriptions to user agents contains elements res_3 and location_of_holy_grail. Thus, {res_3,location_of_holy_grail}$\subseteq OT_1$. Third, the set of query terms used by user agents contains query terms query_1 and where_is_holy_grail. Formally: {query_1,where_is_holy_grail}$\subseteq Q$. The query about the location of the Holy Grail and the ontology term describing a resource that reveals the location of the Holy Grail serve to illustrate incorrect or formally correct, but impossible traces.

First, two example traces for broker are presented. Both traces only contain states of $\mathcal{S}_{\text{broker}}$, so both traces are elements of $\mathcal{LT}_{\text{broker}}$. Local traces of broker, or, in other words, elements of Beh_{loc}(broker), should satisfy the requirement mentioned in Section 4.2. The first trace presented below satisfies the requirement presented in Section 4.2 and is therefore an element of Beh_{loc}(broker). The second trace does not satisfy the requirement, as, after receipt of a query for which it knows a matching resource, it does not transmit information on this resource to user_1. Instead, it transmits information on another resource, res_2. Therefore, this trace is not an element of Beh_{loc}(broker).

$lt_{\text{broker},1}$ = ∅ | belief(match(res_1,query_1)) | ∅ →
communicated_by(query_1,user_1) | belief(match(res_1,query_1)) | ∅ →
communicated_by(res_1,provider_1) | belief(match(res_1,query_1)) | ∅ →
∅ | belief(match(res_1,query_1)) | to_be_communicated_to(res_1,user_1) →
∅ | belief(match(res_1,query_1)) | just_communicated_to(res_1,user_1)

$lt_{\text{broker},2}$ = ∅ | belief(match(res_1,query_1)) | ∅ →
communicated_by(query_1,user_1) | belief(match(res_1,query_1)) | ∅ →
∅ | belief(match(res_1,query_1)) | to_be_communicated_to(res_2,user_1) →
∅ | belief(match(res_1,query_1)) | just_communicated_to(res_2,user_1)

Second, two traces for user_1 are presented. Both traces only contain states of $\mathcal{S}_{\text{user\_1}}$, so both traces are elements of $\mathcal{LT}_{\text{user\_1}}$. Moreover, both traces are elements of Beh_{loc}(user_1), because both comply with the intended functionality of a user agent in the running example. However, the second trace cannot be an actual trace for behaviour of user_1, because in the example, there is no information provider that provides information on the location of the Holy Grail.

$lt_{\text{user\_1},1}$ = ∅ | ∅ | to_be_communicated_to(query_1,broker) →
ready_for_information | ∅ | ∅ →
communicated_by(res_3,broker) | ∅ | ∅

$lt_{\text{user\_1},2}$ = ∅ | ∅ | to_be_communicated_to(query_where_is_holy_grail,broker) →
ready_for_information | ∅ | ∅ →
communicated_by(location_of_holy_grail,broker) | ∅ | ∅ ■

These example traces are used in several examples below. The information state of a link also changes over time, and this is likewise modelled by traces of link states as follows:

Definition 5.15. (Information link trace). *An* information link trace *of an information link I for a time frame* $TF=\langle T;<\rangle$ *is a pair* $LT_I=\langle TF;V\rangle$, *where V is a total function* $V: T \rightarrow \mathcal{S}_I$. *The set of all link traces of a link from D to C is denoted* $\mathcal{LT}_I$.

The way in which the behaviour of a link is described by a set of information link traces, as witnessed by the following definition, is similar to the way in which local component behaviour is defined by a set of local component traces:

Definition 5.16. (Local link behaviour). *Let I be an information link. The* local link behaviour *of I is a set* $Beh_{loc}(I)$ *of information link traces.*

Example 5.17. A possible information link trace for the link broker_to_user_1 is:

$lt_{\text{broker\_to\_user\_1}}$ = awake_and_empty → active_and_contents(res_1) → awake_and_empty →

In this trace, the link is first awake (ready to transmit information) and there are no messages in transit. At the second time point, the link is busy (actively transmitting information, and a message t is in transit. At the third point in time, the message is delivered. The link is empty again and ready to transmit information. ■

5.2.2 Compositional Dynamics

In Section 5.2.1, local component traces are defined as a means to model the behaviour of a single component. In Section 5.1.3, the notion of a structure hierarchy is introduced, which enables the description of compositions consisting of components and links. This section defines structures for describing the behaviour of such compositions, in terms of the local component and link traces defined in Section 5.2.1.

Local component and link traces, the elements of the set $Beh_{loc}(S)$ given for *S*, as defined in Section 5.2.1, are the basic notions for describing the behaviour of a single component or link *S*. A local component trace of a composed component *C* is, by itself, not not sufficient to describe the behaviour of the subcomponents and links of *C*, because local component traces only record local information. However, the behaviour of a composed component *C* and its subcomponents and links can be described by a *structure* consisting of *a number of* local component and link traces: a local component trace of *C*, a local component trace of each of its subcomponents and a local link trace of each link in *C*. Indeed, so-called *multitraces*, which are defined in Section 5.2.2.1 to describe the behaviour of (composed) components, are structures of local component and link traces.

Multitraces cannot be *arbitrary* combinations of elements of the sets $Beh_{loc}(S)$, where $S \in \{C\} \cup Subc(C)$. The set $Beh_{loc}(C)$ as defined in Section 5.2.1 is intended, as stated, to describe the *local* behaviour of *C*. In other words, the behaviour of *C* as described by the set $Beh_{loc}(C)$ is independent of the composition structure in which

C is a part and the information exchange with other components in such a structure. However, to describe the behaviour of a composed component, dependencies imposed by information exchange between subcomponents, and between subcomponents and the composed component, have to be taken into account. For example, assume that no information provider provides information on a resource describing the location of the holy grail. If the behaviour of the running example compositional system were to be described by arbitrary combinations of local component and link traces, the example trace in which user_1 receives information on a resource describing the location of the holy grail (given in Example 5.14) would be included in at least some of the combinations of local component traces. The intended meaning of this fact is that, at least in some actual behaviours of the example compositional system, component user_1 receives information on a resource describing the location of the holy grail. However, such behaviour is impossible because user_1 can only receive such information if one of the information providers can provide such information. By assumption, this is not possible in the running example.

5.2.2.1 Compatibility and Multitraces

Dependencies between components imposed by information transmission are represented by compatibility relations. The principle underlying compatibility relations is as follows. For each link, a compatibility relation for that link describes which local component or link traces of the domain of the information link are compatible with which local component or link traces of the co-domain of the link. A local component or link trace of the domain is compatible with a local component or link trace of the co-domain if the information transmission as defined by the information link mapping, and specific general properties of information transmission, are taken into account (this is defined formally and in detail in Section 6.1). The behaviour of a component not only depends on the behaviour of other components with which it exchanges information, but also on the state of the links used for this information exchange (commitment presented in Section 2.2.3.2). For instance, in the running example, if the information link from broker to user_1 is continuously disabled, then the traces given for broker and user_1 cannot be considered to be compatible. Therefore, a compatibility relation for an information link *I* is defined as a *ternary* relation on the set of local component or link traces of the domain of *I*, the set of link traces of *I*, and the set of local component or link traces of the co-domain of *I*.

The focus of this section is on how compatibility relations can be employed to determine which combinations of local component traces of a component, its subcomponents and its links, constitute its actual behaviour. As a consequence, although compatibility relations are defined in this section, the discussion of various properties of compatibility relations is postponed until Chapter 6.

Definition 5.18. (Compatibility relation). *A* compatibility relation *for a link I is a relation* $\mathcal{CR}_I \subseteq \mathcal{LT}_{dom(I)} \times \mathcal{LT}_I \times \mathcal{LT}_{cdom(I)}$.

Example 5.19. Compatibility relations vary widely in their appearance due to the flexibility offered by the semantic structure. For instance, properties of the local component or link traces of the components and links related by a compatibility relation influence how a concrete compatibility relation in an application of the semantic structure is defined. In this example, a compatibility relation for a link *I* is defined, based on the following assumptions:

- Local component and link traces of *I*, its domain and co-domain are discrete, totally ordered and infinite (thus, the traces are isomorphic to the natural numbers);
- The link *I* is a private link;
- For link *I*, the facilities offered by the semantic structure for a detailed representation of information transmission as a process and enabling conditions on the state of *I* itself are not used.

Moreover, the compatibility relation defined in this example has the input persistence and lossless transmission properties (which are introduced in Chapter 6). In this example, the following notation is used. Let $LT_S = \langle\langle T, < \rangle; V\rangle \in \mathcal{LT}_S$ be a local component or link trace. The state $V(t)$ in LT_S at time point $t \in T$ is denoted $v_{S,t}$. The compatibility relation, $\mathcal{CR}_I$, is defined as follows: $\mathcal{CR}_I$ is the set of tuples $\langle LT_{dom(I)}; LT_I; LT_{cdom(I)}\rangle$ such that:

- $LT_{dom(I)} = \langle\langle T_{dom(I)}; <_{dom(I)}\rangle; V_{dom(I)}\rangle \in \mathcal{LT}_{dom(I)}$;
- $LT_I = \langle\langle T_I; <_I\rangle; V_I\rangle \in \mathcal{LT}_I$;
- $LT_{cdom(I)} = \langle\langle T_{cdom(I)}; <_{cdom(I)}\rangle; V_{cdom(I)}\rangle \in \mathcal{LT}_{cdom(I)}$;
- For all $i \in T_{dom(I)}$, either:
  - There are $j \in T_{dom(I)}$, $i" <_I j" <_I k <_I l \in T_I$ and $i' <_{cdom(I)} j' \in T_{cdom(I)}$ such that $\langle\langle out(v_{dom(I),i}); out(v_{dom(I),j})\rangle; \langle v_{I,i"}; v_{I,j"}; v_{I,k}; v_{I,l}\rangle; \langle in(v_{cdom(I),i'}); in(v_{cdom(I),j'})\rangle\rangle \in \lambda_I$,
  - Or there is no $\langle\langle out(v_{dom(I),i}); \sigma_2\rangle; \langle \sigma_3; \sigma_4; \sigma_5; \sigma_6\rangle; \langle \sigma_7; \sigma_8\rangle\rangle \in \lambda_I$ for any $\sigma_2, \ldots, \sigma_8$.

This definition shows the role of information link mappings: only those traces in which states occur in specific sequences as specified by the information link mapping, are compatible. ∎

Example 5.20. In the example, a possible compatibility relation for the link broker_to_user_1 is given. A compatibility relation for this link consists of triples of local component and link traces of broker_to_user_1, *dom(*broker_to_user_1*)*=broker and *cdom(*broker_to_user_1*)*=user_1. Traces for broker and user_1 are given in Example 5.14. A trace for the link broker_to_user_1 is given in Example 5.17. A compatibility relation for broker_to_user_1 relates traces of broker, user_1 and broker_to_user_1 according to the information link mapping of broker_to_user_1 given in Example 5.7, together with general properties of information transmission presented in Chapter 6. (In this example, such properties are briefly sketched where necessary.) The triple $\langle lt_{\text{broker},1}; lt_{\text{broker\_to\_user\_1}}; lt_{\text{user\_1},1}\rangle$ is an element of a compatibility relation for broker_to_user_1, for the following reasons. First, the information link mapping presented in Example 5.7 states that if a state to_be_communicated_to(t,user_1) occurs, for t an ontology term, and the state of link broker_to_user_1 is awake_and_empty, then user_1 should reach the state communicated_by(t',broker), for $t'=trans(t)$ another ontology term that is the translation of t. Moreover, before user_1 can reach this state, it must have reached the state ready_for_information, and after state to_be_communicated_to(t,user_1) occurs in broker, the state just_communicated_to(t,user_1) should have occurred. Furthermore, the link broker_to_user_1 should reach the state active_and_contents(t) and after that, awake_and_empty. A compatibility relation for broker_to_user_1 relates local component and link traces in which the situation called for by the information link mapping actually occurs. The traces in the triple $\langle lt_{\text{broker},1}; lt_{\text{broker\_to\_user\_1}}; lt_{\text{user\_1},1}\rangle$ are traces that agree with the requirement represented by the information link mapping. At the end of trace $lt_{\text{broker\_1},1}$, the output substate first is to_be_communicated_to(t,user_1) and after that, just_communicated_to(t,user_1) as required. At the end of trace $lt_{\text{user\_1},1}$, first the state ready_for_information occurs, followed by the state communicated_by(t',broker). Trace $lt_{\text{broker\_to\_user\_1}}$ also fulfils the requirement: its first state is awake_and_empty, then the state active_and_contents(term_1) occurs, followed by the state awake_and_empty. Second, the traces in the triple $\langle lt_{\text{broker},1}; lt_{\text{broker\_to\_user\_1}}; lt_{\text{user\_1},1}\rangle$ fulfil general properties of information transmission. In fact, the three traces fulfil the following properties: information transmitted by broker arrives at user_1, and it is not garbled. The three traces also comply with the most basic property of information transmission (information does not arrive before it is sent). However, the example is too simple to prove or disprove this claim. ■

As stated in Section 4.1.1.2, in fact three views on the behaviour of a compositional system are defined: the glass box view, the white box view and the black box view. In all three views, the behaviour of a component is defined in terms of *multitraces*. A multitrace is a collection of local component traces and link traces of a set of components and links, indexed by a structure hierarchy as defined in Section 5.1.3. With compatibility relations imposing additional structure and subject to certain other requirements, these collections of local component traces and link traces model behaviour as defined by the three views. An indexed set can also be seen as a function, which is the view taken in the formal definition below:

Definition 5.21. (Multitrace). *Let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a structure hierarchy. A* multitrace *$(mt_S)_{S\in Comp\cup Lnk}$ for SH is a total function $mt: Comp\cup Lnk \rightarrow \bigcup_{S\in Comp\cup Lnk} \mathcal{LT}_S$ such that for all $S\in Comp\cup Lnk$, $mt(S)\in\mathcal{LT}_S$. The set of all multitraces for SH is denoted MT_{SH}. A typical element of MT_{SH} is denoted μ. The element of a multitrace μ with index P is denoted μ_P (or sometimes $\mu(P)$ if this is more convenient).*

The hierarchy relation $\prec$ on the index set of a multitrace induces a hierarchical structure on the collection of local component and link traces that is indexed by this set, as depicted in Figure 5.1, which refers to the running example. In this figure, the left hand sides depicts the hierarchy relation of the structure hierarchy *sh* given in Example 5.10. The dashed lines show the multitrace as a mapping that maps to each component a local component trace depicted at the right hand side.

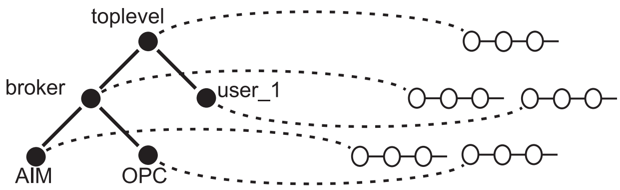

Figure 5.1: Hierarchical structure of a multitrace.

In the semantic structure developed in this thesis, multitraces are used to model possible behaviours of composed components. To model a possible behaviour, a multitrace, which is a collection of local component traces without additional structure apart from the indexing, has to take constraints imposed by non-local phenomena into account. Such constraints are represented by compatibility relations. Therefore, in the following definition, *compatible multitraces* are defined as multitraces in which the local component traces and link traces in the multitrace are related by compatibility relations.

Definition 5.22. (Compatible multitrace). *Let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a structure hierarchy and let $\gamma=(\gamma_I)_{I\in Lnk}$ be a collection of compatibility relations. A multitrace μ for SH is* compatible for γ *iff the following property holds:*

$$\forall I\in Lnk:\ \langle \mu_{dom(I)};\mu_I;\mu_{cdom(I)}\rangle\in\gamma_I.$$

This definition shows why multitraces are indexed by a set that not only includes components, but also links: compatibility requires that the middle element of a triple in the compatibility relation is present in the multitrace (denoted above by μ_I).

If a component C is primitive, according to a structure hierarchy SH for C, and there are no links from C to itself, (thus $SH=\langle\{C\};\emptyset;\emptyset;\emptyset;\emptyset\rangle$), then every multitrace for SH is compatible.

Before the definitions of the three views on behaviour can be given, two additional notions derived from the notion of a structure hierarchy are defined. In some definitions, only a subtree of (one of the trees occurring in) a structure hierarchy is used. Such a subtree is unique for a given structure hierarchy and is defined as follows:

Definition 5.23. (Substructure). *Let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a structure hierarchy and let $S\in Comp\cup Lnk$ be a component or link. If $S\in Comp$, then the* substructure *SS(S,SH) of SH for S is defined as the structure hierarchy* $\langle Comp';Lnk';\prec';dom';cdom'\rangle$ *where:*

- $Comp'=\{ C'\in Comp \mid C'\prec^{*}S \}\cup\{S\}$, *where is* $\prec^{*}$ *the transitive closure of* $\prec$;
- $Lnk'=\{ I\in Lnk \mid dom(I),cdom(I)\in Comp' \}$;
- $S\prec'C \Leftrightarrow S\prec C$, $S\in Comp'\cup Lnk'$ *and* $C\in Comp'$;
- *For all* $I\in Lnk'$, $dom'(I)=dom(I)$;
- *For all* $I\in Lnk'$, $cdom'(I)=cdom(I)$.

If $S\in Lnk$, *then* $SS(S,SH)=\langle\emptyset;\{S\};\emptyset;\emptyset;\emptyset\rangle$.

It is obvious that for arbitrary $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ and *C* in *Comp*, *SS(C,SH)* is a structure hierarchy for *C* (that is, it is a forest of exactly one tree with *C* as root). The notion of a substructure of *SH* for *S* with *S* a link is used in other definitions where a variable *S'* ranges over all components and links in *SLC(C,SH)*, such as the following definition:

Definition 5.24. (Primitive components). *Let C be a component and let SH be a structure hierarchy for C. The set of* primitive components *in SH is defined as follows:*

$$Prim(SH) = \bigcup_{C'\in Subc(C,SH)} Prim(SS(C',SH)),\ \text{if } SH=\langle Comp;Lnk;\prec;dom;cdom\rangle \text{ such that } Comp\supset^{6}\{C\}.$$

$$Prim(SH) = \{C\},\ \text{if } SH=\langle\{C\};Lnk;\emptyset;dom;cdom\rangle.$$

This inductive definition of the set of primitive components in a structure hierarchy is referenced in several definitions and propositions below. Moreover, it serves as a means for proofs by induction of these propositions. The stage is now set to define the three views on the behaviour of a component.

5.2.2.2 The White Box View

The first of the three views presented is the white box view. The three views on the behaviour of a component are defined relative to a structure hierarchy and the behaviour of specific other components in the structure hierarchy. Thus, the three views can be seen as composition operators that define behaviour of a composition of components in terms of the behaviour of the constituents of the composition. Consequentially, if the behaviour of the components in the structure hierarchy to which each view is relative, is not correct, then the behaviour defined by each of the views is also incorrect. The behaviour of a component *S* of *C* to which a view on the behaviour of *C* is relative need not be $Beh_{loc}(S)$. However, it is required that the behaviour of such a set *S* is a subset of $Beh_{loc}(S)$.

The white box view on the behaviour of a component differs from the other two views with respect to the kind of structure hierarchy considered. On the one hand,

[6] In this thesis, $\subset$ and $\supset$ denote *proper* subsets.

the black box and glass box views are both relative to an arbitrary structure hierarchy. The definitions of these two views refer to compatible multitraces for the relevant structure hierarchy. Consequently, the black box and glass box views may take the behaviour of arbitrary components into account by choosing an appropriate structure hierarchy. On the other hand, the white box view is relative to a composition structure *CS*, which is not an arbitrary structure hierarchy. As a consequence, the definition of the white box view on the behaviour of a composed component *C* can only refer to the behaviour of the subcomponents of *C*. (As stated in Chapter 2, in this thesis the term 'subcomponent' always refers to *direct* subcomponent of a component *C*.)

In addition to a composition structure *CS*, the white box view is relative to a collection of compatibility relations γ and to the behaviour of the subcomponents and links (the elements of $SLC(C,CS)\setminus\{C\}$). The white box view on the behaviour of a component *C* consists of a set of structures (multitraces) each of which consists of local component traces of *C*, its subcomponents and links. Because of nondeterminism, which gives rise to different alternative behaviours, a component can have more than one multitrace. (Each multitrace contains one behaviour alternative of a specific component.) Therefore, the white box view consists of a set of multitraces.

Definition 5.25. (Component behaviour, white box view). *Let C be a component, let* $CS=\langle Comp;Lnk;\prec;dom;cdom\rangle$ *be a non-empty composition structure for C, let* $\gamma=(\gamma_l)_{l\in Lnk}$ *be a collection of compatibility relations and let* $\Gamma=(\Gamma_S)_{S\in SLC(C,CS)}$ *be a collection of sets of traces such that for all* $S\in SLC(C,CS)$, $\Gamma_S\subseteq Beh_{loc}(S)$. *The* white box view *on the behaviour of C,* $Beh_{WB}(C,CS,\gamma,\Gamma)$, *with respect to CS,* γ *and* Γ *is the set of compatible multitraces* $\mu\in MT_{CS}$ *of C such that for each subcomponent or link S of C it holds that the local component trace of S in* μ *is an element of* Γ_S. *Formally:*

$$Beh_{WB}(C,CS,\gamma,\Gamma) = \{ \mu \mid \mu\in MT_{CS} \text{ is compatible for } \gamma \text{ and } \forall S\in SLC(C,CS){:}\ \mu_S\in\Gamma_S \}.$$

This definition of component behaviour provides a white box view on the behaviour of a component in the sense that each multitrace in $Beh_{WB}(C,CS,\gamma,\Gamma)$ not only contains a local component trace of *C*, but also local component and link traces of the subcomponents and links of *C*. However, these local component and link traces themselves do not contain information of their subcomponents and so on, recursively. Nevertheless, as is the case with the black box view on the behaviour of *C*, the requirement on the local component traces that constitute $Beh_{WB}(C,CS,\gamma,\Gamma)$ ensures that the combinations of local component traces and link traces that constitute the elements of $Beh_{WB}(C,CS,\gamma,\Gamma)$ take constraints imposed by information transmission into account.

If a component *C* is primitive according to a composition structure *CS* for *C*, (thus $CS=\langle\{C\};\emptyset;\emptyset;\emptyset;\emptyset\rangle$ and $Subc(C,CS)=Lnk(C,CS)=\emptyset$), then for every collection of compatibility relations γ, it holds that $Beh_{WB}(C,CS,\gamma,\Gamma)=\{\ \mu\in MT_{CS}\ |\ \mu_C\in Beh_{loc}(C)\ \}$ (with $\Gamma=\emptyset$).

5.2.2.3 The Black Box View

The second of the three views presented is the black box view. Similar to the other two views, the black box view is defined relative to a structure hierarchy, a collection of compatibility relations and the behaviour of specific other components in the structure hierarchy. In contrast to the white box view, but similar to the glass box view, the black box view is relative to an arbitrary structure hierarchy. (The white box view is relative to a specific type of structure hierarchy: a composition structure.)

In addition to an arbitrary structure hierarchy *SH*, the black box view on the behaviour of a component *C* is defined relative to a collection of compatibility relations γ and the behaviour of its subcomponents and links (the elements of $SLC(C,SH)\setminus\{C\}$).

Definition 5.26. (Component behaviour, black box view). *Let C be a component, let* $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ *be a non-empty structure hierarchy for C, let* $\gamma=(\gamma_I)_{I\in Lnk}$ *be a collection of compatibility relations and let* $\Gamma=(\Gamma_S)_{S\in SLC(C,SH)}$ *be a collection of sets of traces such that for all* $S\in SLC(C,SH)$, $\Gamma_S\subseteq Beh_{loc}(S)$. *The* black box view $Beh_{BB}(C,SH,\gamma,\Gamma)$ *for C with respect to SH,* γ *and* Γ_S *on the behaviour of C is the subset of* $Beh_{loc}(C)$ *such that each local component trace in this subset is part of a compatible multitrace for SH that is based on the given traces of the subcomponents and links of C. Formally:*

$$\begin{aligned} Beh_{BB}(C,SH,\gamma,\Gamma) = \{\ \mu_C\ |\ & \mu\in MT_{SH} \text{ is compatible for } \gamma, \\ & \forall S\in SLC(C,SH){:}\ \mu_S\in\Gamma_S \text{ and} \\ & \forall I\in Lnk \text{ such that } dom(I)=cdom(I)=C{:}\ \mu_I\in Beh_{loc}(I)\ \}. \end{aligned}$$

The definition given above provides a black box view in the sense that the behaviour of a component *C* is a set consisting of local component traces of *C* itself only, and not of its subcomponents and links. As the definition of local component traces given in Section 5.2.1 indicates, only local information is recorded in a local component trace of a component. In particular, information of subcomponents is *not* recorded in the local component trace of the encompassing component and is thus not visible in the black box view on component behaviour as defined above.

As stated in the beginning of Section 5.2.2, the behaviour of a composed component is, in general, not a set of *arbitrary* combinations of local component traces of its subcomponents and link traces, because only combinations that take constraints imposed by information transmission into account can be considered to represent behaviour of the component. The requirement on the local component traces that constitute $Beh_{BB}(C,SH,\gamma,\Gamma)$ ensures that the combinations of local component traces from which the elements of $Beh_{BB}(C,SH,\gamma,\Gamma)$ are taken, take constraints imposed by information transmission into account. This is ensured because $Beh_{BB}(C,SH,\gamma,\Gamma)$ depends on the existence of a compatible multitrace, and the elements of this compatible multitrace are themselves part of the behaviour of the subcomponents, to which the definition of the black box view is relative.

To determine the black box view on the behaviour of a component *C*, only the behaviour of the subcomponents and links of *C* needs to be given. This also holds if, according to the related structure hierarchy, these subcomponents are composed. It suffices to only take the behaviour of the subcomponents into account, (and not of their subcomponents), because it is assumed that the behaviour given for these subcomponents takes constraints imposed by their subcomponents into account. However, if this is not the case, the behaviour as defined by the black box view is incorrect.

If a component *C* is primitive, according to a structure hierarchy *SH* for *C*, and there are no links from *C* to itself, (thus $SH=\langle\{C\};\emptyset;\emptyset;\emptyset;\emptyset\rangle$, and $Subc(C,SH)=Lnk(C,SH)=\emptyset$), then for every collection of compatibility relations γ, it holds that $Beh_{BB}(C,SH,\gamma,\Gamma)=Beh_{loc}(C)$ (with $\Gamma=\emptyset$). However, $Beh_{BB}(C,SH,\gamma,\Gamma)$ is *not* equal to $Beh_{WB}(C,SH,\gamma,\Gamma)$ because $Beh_{WB}(C,SH,\gamma,\Gamma)$ is a set of multitraces, while $Beh_{BB}(C,SH,\gamma,\Gamma)$ is a set of local component traces.

5.2.2.4 The Glass Box View

As is the case for the black box view, the glass box view is relative to a structure hierarchy *SH*, a collection of compatibility relations γ and the behaviour specific subcomponents (in this case, the primitive subcomponents). The glass box view on the behaviour of a component *C* is defined by imposing three requirements on the set of multitraces μ for a structure hierarchy $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ for *C*: (i) they must be compatible, (ii) for all components *C'* in *Comp* that are primitive in *SH*, the local component trace $\mu_{C'}$ must be an element of the given sets of traces and (iii) for all components and links *I* in *Lnk*, the link trace μ_I must be an element of $Beh_{loc}(I)$. Formally:

Definition 5.27. (Component behaviour, glass box view). *Let C be a component, let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a non-empty structure hierarchy for C, let $\gamma=(\gamma_I)_{I\in Lnk}$ be a collection of compatibility relations and let $\Gamma=(\Gamma_S)_{S\in Prim(SH)}$ be a collection of sets of traces for the primitive components in SH. The* glass box view *$Beh_{GB}(C,SH,\gamma,\Gamma)$ for C with respect to SH, γ and Γ on the behaviour of C is the subset of the set MT_{SH} of multitraces for SH such that for all $\mu\in Beh_{GB}(C,SH,\gamma,\Gamma)$ it holds that:*

- *μ is compatible for γ,*
- *$\forall C'\in Prim(SH)\setminus\{C\}$: $\mu_{C'}\in\Gamma_{C'}$ and*
- *$\forall S\in Comp\cup Lnk$: $\mu_S\in Beh_{loc}(S)$.*

The glass box view is the most complete view on the behaviour of a component, as it consists of multitraces for a structure hierarchy *SH*. At first sight, the white box view may look like a special case of the glass box view. This is indeed *almost* the case for a structure hierarchy $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ for a component C if this structure hierarchy consists of only two levels. In this case $SH=SS(C,SH)=CS(C,SH)$ and $Prim(SH)\cup\{C\}=SLC(C,SH)=Comp\cup Lnk$, so the three requirements that a multitrace μ must meet to be an element of $Beh_{GB}(C,SH,\gamma,(\Gamma_S)_{S\in Prim(SH)})$ can be written as:

- μ is compatible for γ,
- $\forall C'\in SLC(C,SH)\setminus\{C\}$: $\mu_{C'}\in\Gamma_{C'}$ and
- $\forall S\in SLC(C,SH)$: $\mu_S\in Beh_{loc}(S)$.

The only difference, in this case, between the white box and the glass box view is that the white box view requires that $\mu_C\in\Gamma_S$, while the glass box views requires that $\mu_C\in Beh_{loc}(S)$. An alternative definition of the white box view can thus be given:

Definition (Component behaviour, white box view, alternative definition I). *Let C be a component, let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a structure hierarchy for C such that $SH=SS(C,SH)$, let $CS=SS(C,SH)$, let $\gamma=(\gamma_I)_{I\in Lnk}$ be a collection of compatibility relations and let $\Gamma=(\Gamma_S)_{S\in SLC(C,SH)}$ be a collection of sets of traces such that for all $S\in SLC(C,SH)$, $\Gamma_S\subseteq Beh_{loc}(S)$. The* white box view *$Beh_{WB}(C,SH,\gamma,\Gamma)$ for C with respect to CS, γ and Γ on the behaviour of C is the set of compatible multitraces in $Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S\in Prim(SH)})$ (with $\Delta_S=\Gamma_S$ for all $S\in Prim(SH)$) such that $\mu_C\in\Gamma_S$. Formally:*

$$Beh_{WB}(C,SH,\gamma,(\Gamma_S)_{S\in SLC(C,SH)}) = \{\mu\in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S\in Prim(SH)}) \mid \mu_C\in\Gamma_S\}.$$

In this case, for structure hierarchies with more than two levels, the white box view is not defined. (Note that Definition 5.25 of the white box view is defined for composition structures, which are two-level structure hierarchies, so the alternative definition and Definition 5.25 are similar in this respect.)

A completely different definition of the white box view is to define the white box view for arbitrary structure hierarchies, but only focus on the two highest levels:

Definition (Component behaviour, white box view, alternative definition II). *Let C be a component, let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a structure hierarchy for C, let Lnk' be the set of links in SLC(C,SH), let $\gamma=(\gamma_I)_{I\in Lnk'}$ be a collection of compatibility relations for the links in SLC(C,SH) and let $\Gamma=(\Gamma_S)_{S\in SLC(C,SH)}$ be a collection of sets of traces such that for all $S\in SLC(C,SH)$, $\Gamma_S\subseteq Beh_{loc}(S)$. The* white box view *$Beh_{WB}(C,SH,\gamma,\Gamma)$ for C with respect to CS, γ and Γ on the behaviour of C is the set of multitraces $\mu\in MT_{CS(C,SH)}$ of C such that:*

- *$\forall I\in Lnk'$: $\langle\mu_{dom(I)};\mu_I;\mu_{cdom(I)}\rangle\in\gamma_I$, and*
- *$\forall S\in SLC(C,SH)$: $\mu_S\in\Gamma_S$.*

Note that in this second alternative definition, it is *not* required that μ is a compatible multitrace, because the definition of compatible multitraces requires that $\langle\mu_{dom(I)};\mu_I;\mu_{cdom(I)}\rangle\in\gamma_I$ for all $I\in Lnk$, not only for all I in the subset *Lnk'* of *Lnk*.

Of the three definitions of the white box view on the behaviour of a component *C*, Definition 5.25 is chosen because this definition puts most emphasis on the compositional nature of the semantic structure. A composition structure *CS* describes the structure of a component as an entity independent of other components with which it is related. The white box view relative to a composition structure *CS* describes the behaviour of such an independent component. The other two views on the behaviour of a component can then be employed to describe the behaviour of the component in a larger context. In the rest of this section, a proposition is presented that enables the definition of the behaviour of such a larger context as a composition of the white box views on the behaviour of its constituents.

In fact, in the rest of this section, three propositions are presented that relate the three views on the behaviour of a component. These propositions show how the white box view and the black box view can be expressed in terms of the glass box view by *restricting* the multitraces (in the usual sense of restricting the domain of a function) in the glass box view to a subset of the components and links in a structure hierarchy of *C*. A restriction of a multitrace to a set *S* of components and links is denoted $\mu_{|S}$ and is called a *restricted multitrace*. The set of all multitraces restricted by a set *S* is denoted $MT_{|S}$.

The first proposition shows how the white box view and the glass box view are related. On the one hand, the proposition shows how the white box view can be expressed in terms of the glass box view by restricting multitraces. On the other hand, the proposition shows how the glass box view on the behaviour of a composed component can be constructed from the white box views on the behaviour of this component, its subcomponents and their subcomponents, and so on. The proposition assumes that a structure hierarchy *SH* for a component *C* is given and states that, if a multitrace for this structure hierarchy satisfies a specific requirement for the restriction of this multitrace to the subcomponents of *C*, and their subcomponents, and so on, then this multitrace is an element of the glass box view on the behaviour of *C*. The definition of the proposition involves the following technical issues:

- Figure 5.2 shows a structure hierarchy for a composed component *C* consisting of six composed components (grey ovals), thirteen primitive components (white ovals) and four links (small white boxes). For each composed component *C'*, the composition structure *CS(C',SH)* is enclosed in a solid line. Within component S_1, as an example also $SLC(S_1,SH)\setminus\{S_1\}$ is depicted by a dashed line. The '=' between $CS(S_1,SH)$ and $SLC(S_1,SS(S_1,SH))$ is put between quotes because $CS(S_1,SH)$ is a *structure hierarchy* and $SLC(S_1,SS(S_1,SH))$ is a *set*, and therefore, they cannot be directly compared.

The proposition below expresses the white box view in terms of the glass box view on the behaviour of *C'* by restricting multitraces for *SH* to *SLC(C',CS(C',SH))*. To construct the glass box view from the white box views on the behaviour of each *C'*, the proposition assumes that for each component *C'* in *SH* (composed and primitive), the restriction of a multitrace for *SH* to *SLC(C',SS(C',SH))* is an element of the white box view on the behaviour of *C'*.

- The glass box view constructed from the white box views is relative to a collection of sets of local component and link traces of the primitive components in *SH*, as indicated by the definition of the glass box view on the behaviour of a composed component. Such a collection of sets $(\Delta_S)_{S \in Prim(SH)}$ is assumed to be given.
- Each of the white box views from which the glass box view is constructed, is itself relative to a collection of sets of local component and link traces, as indicated by the definition of the white box view on the behaviour of a component. These collections are taken from multitraces for *SH* as follows. Let μ be a multitrace for *SH*. For each component *C'* in *SH*, a collection of sets of local component and link traces $(\Gamma_S)_{S \in SLC(C',SS(C',SH))}$ is defined as follows: $\Gamma_S = \Delta_S$ if *S* is a primitive component in *SH*, or $\Gamma_S = \{\mu_S\}$ otherwise. The collections of sets of traces for each composed component in *SH* are themselves grouped as a collection of collections $\Gamma\Gamma$ indexed by the components in *SH*. (Figure 5.2 might help to obtain an overview of the index sets involved.)
- The collection of collections $\Gamma\Gamma$ is well-defined, although for each *composed* component *S* in *SH* that is a subcomponent of a component *C'*, $(\Gamma\Gamma_{C'})_S$ is defined twice: once because $S \in SLC(C',SS(C',SH))$ and once because $S \in SLC(S,SS(S,SH))$. In both cases, $(\Gamma\Gamma_{C'})_S = \{\mu_S\}$ because *S* is composed.
- Each of the white box views from which the glass box views is constructed, is itself also relative to a collection of compatibility relations indexed by the set of links in the composition structure to which the glass box view is relative. A collection of collection of compatibility relations $\gamma\gamma$, indexed by the set *Comp*, is defined in terms of the collection of compatibility relations γ to which the glass box view is relative and which is assumed to be given, as follows: $\gamma\gamma = ((\gamma_I)_{I \in Lnk'})_{C' \in Comp}$, with *Lnk'* the set of links in $CS(C',SH) = \langle Comp';Lnk';\prec';dom';cdom' \rangle$, such that for all $C' \in Comp$ and for all $I' \in Lnk$, $(\gamma\gamma_{C'})_I = \gamma_I$.
- It is straightforward to construct the glass box view on the behaviour of a primitive component from the white box view on its behaviour, as shown by Proposition 5.30 below. To avoid unnecessary extra cases in the proof of the

proposition below, the proposition is only applicable to composed components.

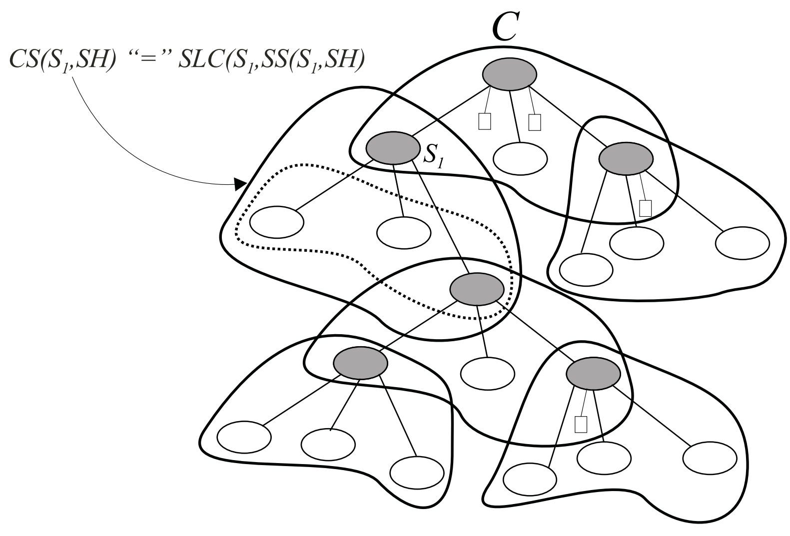

Figure 5.2: Composing the glass box view.

Proposition 5.28. *Let $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ be a structure hierarchy for a composed component C. Let $\mu\in MT_{SH}$ be a multitrace for SH such that for all $S\in Comp\cup Lnk$, $\mu_S\in Beh_{loc}(S)$. Let $\gamma=(\gamma_I)_{I\in Lnk}$ be a collection of compatibility relations and let $(\Delta_S)_{S\in Prim(SH)}$ be a collection of sets of local component traces for the primitive components in SH such that for all $S\in Prim(SH)$: $\Delta_S\subseteq Beh_{loc}(S)$. Define $\gamma\gamma=((\gamma_I)_{I\in Lnk'})_{C'\in Comp}$, with Lnk' the set of links in $CS(C',SH)=\langle Comp';Lnk';\prec';dom';cdom'\rangle$, such that for all $C'\in Comp$ and for all $I'\in Lnk$, $(\gamma\gamma_{C'})_I=\gamma_I$. Define $\Gamma\Gamma=((\Gamma_S)_{S\in SLC(C',CS(C',SH))})_{C'\in Comp}$ such that for all $C'\in Comp$ and for all $S\in SLC(C',CS(C',SH))$:*

$$(\Gamma\Gamma_{C'})_S=\begin{cases}\Delta_S & \text{if } S\in Prim(SH),\\ \{\mu_S\} & \text{otherwise.}\end{cases}$$

Then the following equivalence holds:

For all $C'\in Comp$: $\mu_{|SLC(C',CS(C',SH))}\in Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'})$,

$\Leftrightarrow$ $\mu\in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S\in Prim(SH)})$.

Proposition 5.28 is important for compositional verification of compositional systems (Engelfriet, Jonker & Treur, 1999), because this proposition shows that the semantic structure supports proving global properties of a system from local properties. This topic is not further investigated in this thesis.

The glass box view on the behaviour of a component *C* can be compared to the parallel composition operator ∥ in process algebra with communication (Bergstra & Klop, 1985) in the following way. In Process Algebra, the parallel composition operator takes a number of processes and returns a composed process. The behaviour of the composed process is the same as the concurrent execution of its constituent processes, taking non-local phenomena (information transmission) into account. Likewise, as indicated by Proposition 5.28, the glass box view on behaviour can be used to determine the behaviour of the concurrent execution of a set *Comp* of components given the white box views on the behaviour of each of the components in *Comp* individually. The semantic structure developed in this thesis has a very rich mechanism for composing components to form compositional structures. Therefore, this construction viewed as a composition operator is parameterised by a set of links with which the components are connected in *SH*.

The next proposition shows how the black box view can be expressed in terms of the glass box view. As stated before, all three views are, among others, relative to sets of traces of specific subcomponents. In the context of the propositions presented below, these sets of traces to which the black box view is relative, are defined in terms of the glass box view from which the black box view is generated. This is done by defining a collection of sets $(\Delta_S)_{S\in SLC(C,SH)\setminus\{C\}}$ such that for each $S\in SLC(C,SH)\setminus\{C\}$, $\Delta_S=\{\mu_S \mid \mu\in Beh_{GB}(C,SH,\gamma,\Gamma)\}$. In this proposition, the black box view is taken relative to (Δ_S), which (only) consists of sets of traces for the subcomponents of *C* in *SH* taken from the glass box view. The glass box view itself is defined relative to a collection of sets $(\Gamma_S)_{S\in Prim(SH)\setminus\{C\}}$ of traces of *the primitive components in SH*. In general, the primitive components in a structure hierarchy *SH* for *C* and the subcomponents of *C* are distinct.

Proposition 5.29. *Let C be a composed component, let* $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ *be a non-empty structure hierarchy for C, let* $\gamma=(\gamma_I)_{I\in Lnk}$ *be a collection of compatibility relations and let* $\Gamma=(\Gamma_S)_{S\in Prim(SH)}$ *be a collection of sets of traces for the primitive components in SH. Define* $\Delta=(\Delta_S)_{S\in SLC(C,SH)}$ *such that for all* $S\in SLC(C,SH)$, $\Delta_S=\{\mu_S \mid \mu\in Beh_{GB}(C,SH,\gamma,(\Gamma_S))\}$. *Then:*

$$Beh_{BB}(C,SH,\gamma,\Delta)=\{\ \mu_C\in Beh_{loc}(C) \mid \mu\in Beh_{GB}(C,SH,\gamma,\Gamma)\ \}.$$

The last proposition presented in this section shows how the three views relate to each other in the case of primitive components:

Proposition 5.30. *Let C be a primitive component, let* $SH=\langle\{C\};Lnk;\varnothing;dom;cdom\rangle$ *be a structure hierarchy for C, let* $\gamma=(\gamma_I)_{I\in Lnk}$ *be a collection of compatibility relations, let* $\Gamma=(\Gamma_S)_{S\in\{C\}}$ *be a collection consisting of one set of traces such that* $\Gamma_C\subseteq Beh_{loc}(C)$ *and let* μ *be a multitrace for SH such that for all* $I\in Lnk$, $\mu_I\in Beh_{loc}(I)$. *Then:*

$$\mu\in Beh_{GB}(C,SH,\gamma,\Gamma) \Leftrightarrow \mu\in Beh_{WB}(C,CS(C,SH),\gamma,\Gamma) \Leftrightarrow \mu_C\in Beh_{BB}(C,SH,\gamma,\Gamma).$$

5.2.2.5 Example

In the next example, the behaviour of the example compositional system is described to illustrate how the behaviour of components is modelled.

Example 5.31. In this example, three views on the behaviour of the compositional system in the running example are presented. This example is based on the following assumptions:

- The three views are developed relative to the structure hierarchy *SH* presented in Example 5.10;
- A collection of compatibility relations γ is given. In conformance with Example 5.20, γ is assumed to be defined such that traces $lt_{\text{broker},1}$, $lt_{\text{broker\_to\_user\_1}}$ and $lt_{\text{user\_1},1}$ are compatible;
- Component toplevel only serves as a demarcation component for the example compositional system. Its set of information states is defined as $\mathcal{S}_{\text{toplevel}}=\{\langle\emptyset;\emptyset;\emptyset\rangle\}$;
- The behaviour of OPC and ASP, the subcomponents of broker according to *SH* and Figure 4.5, is such that their collective behaviour fulfils the requirements put forward in Section 4.2.1. Moreover, sets of local component traces are given that reflect this behaviour.

As there is a component, toplevel, that represents the entire running example compositional system, the behaviour of the example compositional system is the behaviour of toplevel. So, in this example, three views on the behaviour of toplevel are developed. The first view on the behaviour of toplevel, the black box view, is obtained as follows. As for toplevel, only one state is distinguished (state $\langle\emptyset;\emptyset;\emptyset\rangle$), the only possible local component trace of toplevel is the trace $lt_{\text{toplevel}}=\emptyset|\emptyset|\emptyset\rightarrow\emptyset|\emptyset|\emptyset\rightarrow\ldots$ This trace is compatible because there are no links from toplevel to any of its subcomponents or vice versa, nor are there any links from toplevel to its parent (which does not exist in *SH*) or components at the same level (which do not exist in *SH* either). Thus, $Beh_{BB}(\text{toplevel},SH,\gamma,\Gamma)=\{\emptyset|\emptyset|\emptyset\rightarrow\emptyset|\emptyset|\emptyset\rightarrow\ldots\}$ for any collection Γ of component and link traces.

The second view on the behaviour of toplevel, the white box view, is not relative to *SH*, but, according to Definition 5.25, to a composition structure. In this example, a composition structure *CS* is defined as follows: $CS=CS(\text{toplevel},SH)$. Moreover, the white box view on the behaviour of toplevel is defined relative to a collection $\Gamma=(\Gamma_S)_{S\in SLC(\text{toplevel},CS)\setminus\{\text{toplevel}\}}$ of sets of traces of the components and links in $SLC(\text{toplevel},CS)\setminus\{\text{toplevel}\}=\{\text{user\_1},\text{user\_2},\text{broker},\text{provider\_1},\text{provider\_2},\text{broker\_to\_user\_1}\}$.
Thus, before the white box view on the behaviour of toplevel can be defined, a relevant collection of component and link traces has to be defined. According to *SH*, only broker is a composed component. Therefore, a relevant collection of component and link traces can be defined by taking $Beh_{loc}(S)$ for the links and

primitive components in $SLC(\text{toplevel},SH)\setminus\{\text{toplevel}\}$ and the black box view on the behaviour of broker relative to its subcomponents. This is motivated as follows:

- Sets $Beh_{loc}(S)$ are assumed not to take any non-local constraint into account. Thus, these sets are, in general, too large: they contain traces that are not possible if non-local constraints were taken into account. However, requirements on the multitraces from which the black box and white box views on the behaviour of toplevel are taken, prevent local component and link traces in $Beh_{loc}(S)$ that do not take information transmission with components at the same level and with the parent component from appearing in these multitraces. Moreover, according to SH, there are no subcomponents of the primitive components that further constrain which local component or link traces from $Beh_{loc}(S)$ can appear in multitraces. Therefore, it is reasonable to take $Beh_{loc}(S)$ for the links and primitive components in SH. As an aside, if specific components or links are omitted from SH, and subcomponents of components that are primitive in SH can be distinguished, taking $Beh_{loc}(S)$ is not a good choice: constraints imposed by the omitted components and links are not taken into account.
- According to SH, broker is a composed component. In this case, it is not a good choice to take $Beh_{loc}(\text{broker})$. Instead, the black box view on the behaviour of broker relative to sets of local component traces of its subcomponents in SH is taken. This is a better choice for the following reason. The black box view, which is a set of component traces as required, is a subset of $Beh_{loc}(\text{broker})$ in which constraints imposed by information exchange with the subcomponents of broker is taken into account. (As an aside, it would have been possible to take the black box view not only for broker, but also for the other components in $SLC(\text{toplevel},SH)\setminus\{\text{toplevel}\}$. However, as these other components are primitive according to SH, it holds that $Beh_{BB}(S,SS(S,SH),\gamma,\varnothing)=Beh_{loc}(S)$, so formally there is no difference.)

Thus, the next step is to develop the black box view $Beh_{BB}(\text{broker},SH,\gamma,(\Delta_{S'})_{S'\in SLC(\text{broker},SH)\setminus\{\text{broker}\}})$ on the behaviour of broker. The black box view is defined relative to the behaviour of the subcomponents of broker according to SH, which are OPC and ASP. (Thus, $SLC(\text{broker},SH)\setminus\{\text{broker}\}=\{\text{APC},\text{OPC}\}$.) As OPC and ASP are primitive according to SH, $(\Delta_{S'})_{S'\in\{\text{OPC},\text{ASP}\}}$ is defined as $\Delta_{S'}=Beh_{loc}(S')$ for all $S'\in\{\text{OPC},\text{ASP}\}$. (The same motivation as for the primitive subcomponents of toplevel applies.) As stated above, it is assumed that the behaviour of OPC and ASP is such that their collective behaviour fulfils the requirements put forward in Section 4.2.1. Therefore, trace $lt_{\text{broker},1}$ presented in Example 5.14, is an example element of $Beh_{BB}(\text{broker},SH,\gamma,(\Delta_{S'})_{S'\in\{\text{OPC},\text{ASP}\}})$, which is a set of local component traces generated from compatible multitraces in $MT_{|SLC(\text{broker},SH)}$.

The white box view on the behaviour of toplevel can now be defined relative to the collection of sets $(\Gamma_S)_{S\in SLC(\text{toplevel},SH)}$ defined such that

$\Gamma_{\text{broker}}=Beh_{BB}(\text{broker},SH,\gamma,(\Delta_{S'})_{S'\in\{\text{OPC},\text{ASP}\}})$ and $\Gamma_S=Beh_{loc}(S)$ for $S\in(SLC(\text{toplevel},SH)\setminus\{\text{toplevel}\})\setminus\{\text{broker}\}$. The white box view is a set of compatible multitraces μ from MT_{CS} such that (among other requirements) for each $S\in SLC(\text{toplevel},CS)\setminus\{\text{toplevel}\}$, $\mu_S\in\Gamma_S$. This requirement holds for the following multitrace, as the traces $lt_{\text{user\_1},1}$, $lt_{\text{broker\_to\_user\_1}}$ and $lt_{\text{broker},1}$ are compatible:

$$mt_1 = \{\langle\text{toplevel};lt_{\text{toplevel}}\rangle,\langle\text{user\_1};lt_{\text{user\_1},1}\rangle,\langle\text{broker\_to\_user\_1};lt_{\text{broker\_to\_user\_1}}\rangle, \langle\text{broker};lt_{\text{toplevel}}\rangle,\langle\text{user\_2};lt_{\text{user\_2}}\rangle, \langle\text{provider\_1};lt_{\text{provider\_1}}\rangle, \langle\text{provider\_2};lt_{\text{provider\_1}}\rangle\},$$

where $lt_{\text{user\_2}}\in Beh_{loc}(\text{user\_2})$, $lt_{\text{provider\_1}}\in Beh_{loc}(\text{provider\_1})$ and $lt_{\text{provider\_2}}\in Beh_{loc}(\text{provider\_2})$. (The other traces referred to in mt_1 have been introduced before.) Other elements of $Beh_{WB}(\text{toplevel},SH,\gamma,\Gamma)$ can be found in a similar way.

To finalise the example, also the glass box view on the behaviour of toplevel is given. The glass box view is defined relative to a collection of sets of traces of the primitive components in SH, the elements of the set $Prim(SH)=\{\text{user\_1},\text{user\_2},\text{provider\_1},\text{provider\_2},\text{OPC},\text{ASP}\}$. This collection $\Delta'=(\Delta'_S)_{S\in Prim(SH)}$ is defined such that for all $S\in Prim(SH)$, $\Delta'_S=Beh_{loc}(S)$. (The same motivation as for the primitive subcomponents of toplevel applies.) An example element of $Beh_{GB}(\text{toplevel},SH,\gamma,\Delta')$ is given by the following multitrace to show how the behaviour of the example multi-agent system is modelled in the glass box view. However, it is left to the reader to check that the hierarchical multitrace given is indeed an element of $Beh_{GB}(\text{toplevel},SH,\gamma,\Delta')$. (This is quite straightforward.)

$$mt_2 = \{\langle\text{toplevel};lt_{\text{toplevel}}\rangle,\langle\text{user\_1};lt_{\text{user\_1},1}\rangle,\langle\text{broker\_to\_user\_1};lt_{\text{broker\_to\_user\_1}}\rangle, \langle\text{broker};lt_{\text{toplevel}}\rangle,\langle\text{OPC};lt_{\text{OPC}}\rangle,\langle\text{ASP};lt_{\text{ASP}}\rangle,\langle\text{user\_2};lt_{\text{user\_2},1}\rangle, \langle\text{provider\_1};lt_{\text{provider\_1}}\rangle,\langle\text{provider\_2};lt_{\text{provider\_1}}\rangle\},$$

This multitrace is an element of MT_{SH}. The hierarchical structure on mt_2 is imposed by the ordering $\prec$ of SH. This ordering is depicted in Figure 5.3 as a tree. ∎

The example presented in this section exhibits two aspects of compositionality. In the first place, the example compositional system consists of five components, one of which itself consists of two subcomponents. The composition of the system is reflected in the formal description of its structure by the structure hierarchy presented in Example 5.10. In the second place, the composition of the example system is also reflected in the behaviour of the system, in particular in the glass box view on its behaviour: the compositional structure of the example system induces the structure of the multitraces that comprises the glass box view on the behaviour of the example system. Moreover, a specific feature of the semantic structure developed in this thesis is that the behaviour of a composed component is not only determined by the behaviour of its subcomponents but also by the local behaviour of the composed component itself. (In the previous examples, $Beh_{loc}(\text{broker})$, which was assumed to be given, represented this factor in the total behaviour of the composed component broker.)

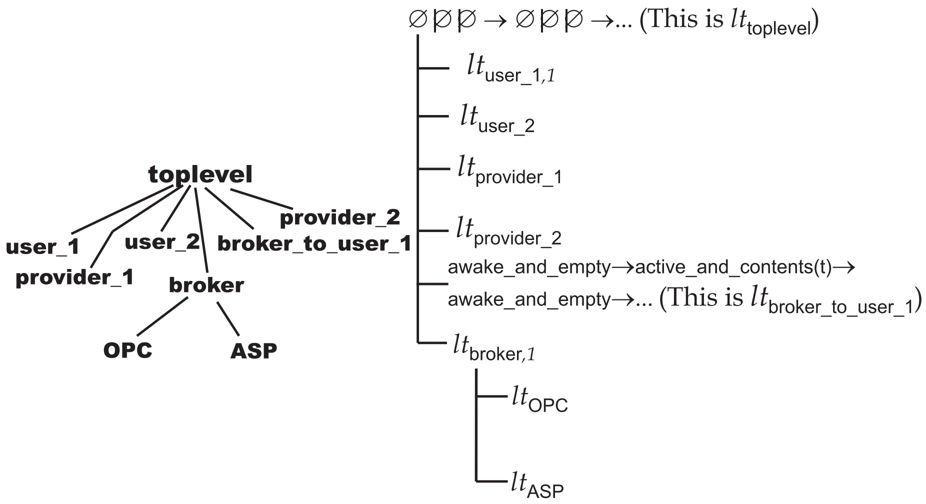

Figure 5.3: A structure hierarchy SH and an element of Beh_{GB}(toplevel,SH,γ,Δ').

5.3 Summary and Outlook

As explained in Chapter 1, this thesis aims at developing a formal, compositional semantic structure for multi-agent system dynamics. The semantic structure, which consists of constructs for building compositional systems, is presented in this chapter. The two most important definitions are the definition of a structure hierarchy and the definition of the glass box view on the behaviour of a component. A structure hierarchy enables an entire compositional system to be defined, while the glass box view defines the most complete picture of the behaviour of such a system. The definition of the glass box view, however, is relative to, among others, a collection of compatibility relations, which are discussed further in Chapter 6. Chapter 7 defines a more global notion of state as an extension of the semantic structure. After further development of the semantic structure in Chapter 6 and Chapter 7, facilities for separated, domain independent control are added in Chapter 8. Finally, in Chapter 9, the semantic structure is applied in the development of a semantics for DESIRE, an extensive compositional modelling framework for multi-agent systems.

5.4 Proofs

This thesis employs a notation for structuring proofs proposed by Lamport (1995). In Section 5.4.1, this notation is introduced. Proofs of the propositions and theorems presented in this chapter are provided in Section 5.4.2.

5.4.1 A Note on Proof Notation

Lamport (1995) advocates the use of a hierarchically structured proof style as a replacement for unstructured proofs presented in the form of (English) prose. The proof style he advocates is a refinement of natural deduction, in which proof steps are numbered according to a specific scheme, and the structure of the proof is represented by the indentation in its textual form. The following principles are applied:

- A proof has a hierarchical structure, which is represented in its textual form by indentation. Each level consists of proof steps, which are themselves proven at the next level. The proof steps at a specific level together prove the statement at the next higher level. A proof of a very simple statement can be given directly at the level of that statement. A proof can be read level by level.
- Proof steps are numbered by a decimal numbering system similar to the numbering of the section headings in this thesis. As step numbers at deeper levels of the proof can become quite long, and as it is easy to confuse e.g. 3.1.1.1.2 with 3.1.1.2, an abbreviation is used: a number such as 3.1.1.2 is written as ⟨4⟩2 (a four-part number ending in 2), while 3.1.1.1.2 is written as ⟨5⟩2 (a five-part number ending in 2). The laws of natural deduction guarantee that the abbreviated form suffices: a step such as 3.1.1.1.2 can only be used after it is proven, but because it is proven under the assumption of step 3.1.1.1, it can only be referred to in the proof of its parent. In the proof of its parent, step 3.1.1.1.2 is the only five-part number ending in 2. Parts of a proof step are referred to by appending the part number to the step number, separated by a semicolon e.g., ⟨3⟩:1. An assumption at the highest level is referred to by ⟨0⟩.
- A proof by cases is introduced by the word ‘Case’, followed by an expression characterising the case, which is an assumption in the proof of the case. A proof by cases ends with a proof that shows that all cases are covered.

According to Lamport, this proof style results in more rigorous and less error prone proofs that are easier to read, especially in the area of correctness proofs for algorithms, where proofs are seldom deep, but have considerable detail.

5.4.2 Proofs of Propositions and Theorems

In Section 5.2.2, three propositions are presented that relate the three views on behaviour developed in that section.

Proposition 5.28. *Let* $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ *be a structure hierarchy for a composed component C. Let* $\mu \in MT_{SH}$ *be a multitrace for SH such that for all*

$S\in Comp\cup Lnk$, $\mu_S\in Beh_{loc}(S)$. Let $\gamma=(\gamma_I)_{I\in Lnk}$ be a collection of compatibility relations and let $(\Delta_S)_{S\in Prim(SH)}$ be a collection of sets of local component traces for the primitive components in SH such that for all $S\in Prim(SH)$: $\Delta_S\subseteq Beh_{loc}(S)$. Define $\gamma\gamma=((\gamma_I)_{I\in Lnk'})_{C'\in Comp}$, with Lnk' the set of links in $CS(C',SH)=\langle Comp';Lnk';\prec';dom';cdom'\rangle$, such that for all $C'\in Comp$ and for all $I'\in Lnk$, $(\gamma\gamma_{C'})_{I'}=\gamma_{I'}$. Define $\Gamma\Gamma=((\Gamma_S)_{S\in SLC(C',CS(C',SH))})_{C'\in Comp}$ such that for all $C'\in Comp$ and for all $S\in SLC(C',CS(C',SH))$:

$$(\Gamma\Gamma_{C'})_S=\begin{cases}\Delta_S & \text{if } S\in Prim(SH),\\ \{\mu_S\} & \text{otherwise.}\end{cases}$$

Then the following equivalence holds:

For all $C'\in Comp$: $\mu_{|SLC(C',CS(C',SH))}\in Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'})$,

$\Leftrightarrow$ $\mu\in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S\in Prim(SH)})$.

Proof. Proof sketch: the proof of the first implication (the implication from left to right) is as follows. For a multitrace μ that complies with a number of assumptions, it is proven that $\mu\in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S\in Prim(SH)})$. Thus, three requirements for elements of $Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S\in Prim(SH)})$ are proven. (Actually, one of the three, the requirement that for all $S\in Comp\cup Lnk$, $\mu_S\in Beh_{loc}(S)$, is an assumption, so only two requirements are proven.) The requirements are checked by induction to the structure of *SH*. The base case for induction is identified as $SH=CS(C,SH)$. Thus, the base case is formed of structure hierarchies consisting of precisely two levels. (It is assumed that *C* is composed.) Then, induction can be applied based on Definition 5.24 (the definition of *Prim(SH)*), which guarantees that for each primitive component *C'* in *SH*, there is a subcomponent *C"* of *C* such that *C'* is a primitive component in *SS(C",SH)*. Technically, the induction steps in the proof are identified as $\exists C'\in Subc(C,SH)$: $C'\notin Prim(SH)$. The second implication (the implication from right to left) consists of straightforward expansions of the definitions.

Assume:
1. $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ is a structure hierarchy for a composed component *C*.
2. μ is a multitrace for *SH* such that for all $S\in Comp\cup Lnk$, $\mu_S\in Beh_{loc}(S)$.
3. $\Gamma\Gamma=((\Gamma_S)_{S\in SLC(C',CS(C',SH))})_{C'\in Comp}$ such that for all $C'\in Comp$ and for all $S\in SLC(C',CS(C',SH))$:

$$(\Gamma\Gamma_{C'})_S=\begin{cases}\Delta_S & \text{if } S\in Prim(SH),\\ \{\mu_S\} & \text{otherwise.}\end{cases}$$

4. $\gamma\gamma=((\gamma_I)_{I\in Lnk'})_{C'\in Comp}$ such that for all $C'\in Comp$ and for all $I'\in Lnk$, $(\gamma\gamma_{C'})_{I'}=\gamma_{I'}$.

Prove: for all $C'\in Comp$: $\mu_{|SLC(C',CS(C',SH))}\in Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'})\Leftrightarrow$ $\mu\in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S\in Prim(SH)})$.

⟨1⟩1. $\forall C' \in Comp\colon \mu_{|SLC(C',CS(C',SH))} \in Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'}) \Rightarrow$
$\mu \in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S \in Prim(SH)})$.
Assume: $\forall C' \in Comp\colon \mu_{|SLC(C',CS(C',SH))} \in Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'})$.
Prove: $\mu \in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S \in Prim(SH)})$.

⟨2⟩1. For each $C' \in Prim(SH)$, $\exists C'' \in Subc(C,SH)\colon C' \in Prim(SS(C'',SH))$.
Proof: by assumption ⟨0⟩:1, C is a composed component. By Definition 5.24, $Prim(SH) = \bigcup_{C' \in Subc(C,SH)} Prim(SS(C',SH))$.

⟨2⟩2. μ is compatible for γ.

⟨3⟩1. Let $I \in Lnk$. There is a $C'' \in Comp$ such that $I \in SLC(C'',CS(C'',SH))$.
Proof: by the definition of a structure hierarchy.

⟨3⟩2. $\mu_{|SLC(C'',CS(C'',SH))} \in Beh_{WB}(C'',CS(C'',SH),\gamma\gamma_{C''},\Gamma\Gamma_{C''})$.
Proof: by assumption ⟨1⟩.

⟨3⟩3. $\mu_{|SLC(C'',CS(C'',SH))}$ is compatible for $\gamma\gamma_{C''}$.
Proof: by step ⟨3⟩2 and the definition of $Beh_{WB}(C'',CS(C'',SH),\gamma\gamma_{C''},\Gamma\Gamma_{C''})$.

⟨3⟩4. $\langle \mu_{dom(I)};\mu_I;\mu_{cdom(I)} \rangle \in (\gamma\gamma_{C''})_I$.
Proof: by step ⟨3⟩3 and the definition of compatible multitraces, for each link $I' \in Lnk'$ where Lnk' is the set of links in $CS(C'',SH)$, $\langle \mu_{dom(I')};\mu_{I'};\mu_{cdom(I')} \rangle \in (\gamma\gamma_{C''})_{I'}$. By step ⟨3⟩1, $I \in Lnk'$.

⟨3⟩5. For all $I \in Lnk$, $\langle \mu_{dom(I)};\mu_I;\mu_{cdom(I)} \rangle \in \gamma_I$.
Proof: by steps ⟨3⟩1 and ⟨3⟩4 and assumption ⟨0⟩:4.

⟨3⟩6. Q.E.D.
Proof: by step ⟨3⟩5 and the definition of compatible multitraces.

⟨2⟩3. $\forall C' \in Prim(SH) \setminus \{C\}\colon \mu_{C'} \in \Delta_{C'}$.

⟨3⟩1. Case: $SH = CS(C,SH)$.

⟨4⟩1. $Prim(SH) \setminus \{C\} \subseteq SLC(C,CS(C,SH)) \setminus \{C\}$.
Proof: by assumption ⟨3⟩.

⟨4⟩2. $\forall S \in SLC(C,SH) \setminus \{C\}\colon (\mu_{|SLC(C,CS(C,SH))})_S \in (\Gamma\Gamma_C)_S$.
Proof: by assumption ⟨1⟩, $\mu_{|SLC(C,CS(C,SH))} \in Beh_{WB}(C,CS(C,SH),\gamma\gamma_C,\Gamma\Gamma_C)$, and by the definition of $Beh_{WB}(C,CS(C,SH),\gamma\gamma_C,\Gamma\Gamma_C)$, $(\mu_{|SLC(C,CS(C,SH))})_S \in (\Gamma\Gamma_C)_S$.

⟨4⟩3. $\forall C' \in Prim(SH) \setminus \{C\}\colon \mu_{C'} \in (\Gamma\Gamma_C)_{C'}$.
Proof: by assumption ⟨3⟩, $\mu_{|SLC(C,CS(C,SH))} = \mu$, and by steps ⟨4⟩1 and ⟨4⟩2, $\mu_{C'} \in (\Gamma\Gamma_C)_{C'}$.

⟨4⟩4. $\forall C' \in Prim(SH) \setminus \{C\}\colon (\Gamma\Gamma_C)_{C'} = \Delta_{C'}$.
Proof: by assumption ⟨0⟩:3 (definition of $\Gamma\Gamma_{C'}$).

⟨4⟩5. Q.E.D.
Proof: by steps ⟨4⟩3 and ⟨4⟩4.

⟨3⟩2. Case: 1. $\exists C' \in Subc(C,SH)\colon C' \notin Prim(SH)$.
2. $\forall S \in Subc(C,SH)\colon \forall C' \in Prim(SS(S,SH)) \setminus \{C\}\colon \mu_{C'} \in \Delta_{C'}$.

⟨4⟩1. Let $C' \in Prim(SH) \setminus \{C\}$. $\exists S \in Subc(C,SH): C' \in Prim(SS(S,SH))$.
Proof: by assumption ⟨3⟩:1 and step ⟨2⟩1.

⟨4⟩2. Q.E.D.
Proof: by step ⟨4⟩1 and assumption ⟨3⟩:2.

⟨3⟩3. Q.E.D.
Proof: by steps ⟨3⟩1, ⟨3⟩2 and induction to the structure of *SH*.

⟨2⟩4. Q.E.D.
Proof: by steps ⟨2⟩2 and ⟨2⟩3, assumption ⟨0⟩:2 and the definition of $Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S \in Prim(SH)})$.

⟨1⟩2. $\mu \in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S \in Prim(SH)}) \Rightarrow$
$\forall C' \in Comp: \mu_{|SLC(C',CS(C',SH))} \in Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'})$.

Assume: 1. μ is a multitrace for *SH* such that $\mu \in Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S \in Prim(SH)})$.
2. $C' \in Comp$ and μ' is a multitrace for $CS(C',SH)$ such that $\mu' = \mu_{|SLC(C',CS(C',SH))}$.

Prove: for all $C' \in Comp: \mu_{|SLC(C',CS(C',SH))} \in Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'})$.

⟨2⟩1. $\mu' \in MT_{CS(C',SH)}$ is compatible for $\gamma\gamma_{C'}$.

⟨3⟩1. μ is compatible for γ.
Proof: by definition of $Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S \in Prim(SH)})$.

⟨3⟩2. For all $I \in Lnk$: $\langle \mu_{dom(I)};\mu_I;\mu_{cdom(I)} \rangle \in \gamma_I$.
Proof: by step ⟨3⟩1 and the definition of compatible multitraces.

⟨3⟩3. Let *Lnk'* be the set of links in $SLC(C',CS(C',SH))$. For all $I' \in Lnk'$: $\langle \mu'_{dom(I')};\mu'_{I'};\mu'_{cdom(I')} \rangle \in \gamma_{I'}$.
Proof: by step ⟨3⟩2 and $Lnk' \subseteq Lnk$, and by assumption ⟨1⟩:2, $\mu' = \mu_{|SLC(C',CS(C',SH))}$.

⟨3⟩4. $\forall I' \in Lnk': \langle \mu'_{dom(I')};\mu'_{I'};\mu'_{cdom(I')} \rangle \in (\gamma\gamma_{C'})_{I'}$.
Proof: by step ⟨3⟩3 and assumption ⟨0⟩:4.

⟨3⟩5. Q.E.D.
Proof: by step ⟨3⟩4 and the definition of compatible multitraces.

⟨2⟩2. For all $S \in SLC(C',CS(C',SH)): \mu'_S \in (\Gamma\Gamma_{C'})_S$.

⟨3⟩1. Case: $C' \in Prim(SH)$.
Proof: by assumption ⟨2⟩, $SLC(C',CS(C',SH))=\{C\}$, thus $S=C'$. By assumption ⟨0⟩:3, $(\Gamma\Gamma_{C'})_S=\Delta_S$. By the definition of $Beh_{GB}(C,SH,\gamma,(\Delta_S)_{S \in Prim(SH)})$, $\mu_S \in \Delta_S$ and by assumption ⟨1⟩:2, $\mu'_S=\mu_S \in \Delta_S=(\Gamma\Gamma_{C'})_S$.

⟨3⟩2. Case: $C' \notin Prim(SH)$.
Proof: by assumption ⟨0⟩:3, $(\Gamma\Gamma_{C'})_S=\{\mu_S\}$. By assumption ⟨1⟩:2, $\mu'_S=\mu_S$. Therefore, $\mu'_S=\mu_S \in \{\mu_S\}=(\Gamma\Gamma_{C'})_S$.

⟨3⟩3. Q.E.D.
Proof: steps ⟨3⟩1 and ⟨3⟩2 list all cases.

⟨2⟩3. Q.E.D.

Proof: by steps ⟨2⟩1 and ⟨2⟩2 and the definition of $Beh_{WB}(C',CS(C',SH),\gamma\gamma_{C'},\Gamma\Gamma_{C'})$.

⟨1⟩3. Q.E.D.
Proof: by steps ⟨1⟩2 and ⟨1⟩2.

Proposition 5.29. *Let C be a composed component, let* $SH=\langle Comp;Lnk;\prec;dom;cdom\rangle$ *be a non-empty structure hierarchy for C, let* $\gamma=(\gamma_I)_{I\in Lnk}$ *be a collection of compatibility relations and let* $\Gamma=(\Gamma_S)_{S\in Prim(SH)}$ *be a collection of sets of traces for the primitive components in SH. Define* $\Delta=(\Delta_S)_{S\in SLC(C,SH)}$ *such that for all* $S\in SLC(C,SH)$, $\Delta_S=\{\mu_S \mid \mu\in Beh_{GB}(C,SH,\gamma,(\Gamma_S))\}$. *Then:*

$$Beh_{BB}(C,SH,\gamma,\Delta)=\{\ \mu_C\in Beh_{loc}(C)\ \mid\ \mu\in Beh_{GB}(C,SH,\gamma,\Gamma)\ \}.$$

Proof. Proof sketch: at the highest level, the proof consists of two inclusions. The proof of the first inclusion is complicated. For an arbitrary element of $Beh_{BB}(C,SH,\gamma,\Delta)$ (a local component trace), it is proven that the multitrace from which this element is taken is an element of $Beh_{GB}(C,SH,\gamma,\Gamma)$. Induction on the structure of *SH* is applied twice to prove two requirements on elements of $Beh_{GB}(C,SH,\gamma,\Gamma)$. Intuitively, induction is needed to ensure that local component or link traces that cannot occur in elements of $Beh_{GB}(C,SH,\gamma,\Gamma)$, do not occur in the multitraces from which $Beh_{BB}(C,SH,\gamma,\Delta)$ is built. The proof of the second inclusion consists of straightforward expansions of definitions.

Assume: 1. *C* is a composed component,
2. $\Delta=(\Delta_S)_{S\in SLC(C,SH)}$ such that for all $S\in SLC(C,SH)$, $\Delta_S=\{\mu_S \mid \mu\in Beh_{GB}(C,SH,\gamma,\Gamma)\}$.

Prove: $Beh_{BB}(C,SH,\gamma,\Delta)=\{\ \mu_C\in Beh_{loc}(C)\ \mid\ \mu\in Beh_{GB}(C,SH,\gamma,\Gamma)\ \}$.

⟨1⟩1. $Beh_{BB}(C,SH,\gamma,\Delta)\subseteq\{\ \mu_C\in Beh_{loc}(C)\ \mid\ \mu\in Beh_{GB}(C,SH,\gamma,\Gamma)\ \}$.
Assume: $LT\in Beh_{BB}(C,SH,\gamma,\Delta)$.
Prove: $LT\in\{\ \mu_C\in Beh_{loc}(C)\ \mid\ \mu\in Beh_{GB}(C,SH,\gamma,\Gamma)\ \}$.

⟨2⟩1. There is a $\mu\in MT_{SH}$ compatible for γ such that $\mu_C=LT\in\Delta_S\subseteq Beh_{loc}(C)$ and $\forall S\in SLC(C,SH),\ \mu_S\in\Delta_S$.
Proof: by definition of $Beh_{BB}(C,SH,\gamma,\Delta)$.

⟨2⟩2. $\forall C'\in Prim(SH)\setminus\{C\}$: $\mu_{C'}\in\Gamma_{C'}$.

⟨3⟩1. Case: $SH=CS(C,SH)$.

⟨4⟩1. $Prim(SH)=Comp\setminus\{C\}\subseteq SLC(C,SH)$.
Proof: by assumptions ⟨3⟩ and ⟨0⟩:1, all subcomponents of *C* are primitive.

⟨4⟩2. $\forall C'\in Prim(SH)$: $\mu_{C'}\in\Delta_{C'}$.
Proof: by definition of $Beh_{BB}(C,SH,\gamma,\Delta)$, $\forall S\in SLC(C,SH)$: $\mu_S\in\Delta_S$ and by step ⟨4⟩1, $Prim(SH)\subseteq SLC(C,SH)$.

⟨4⟩3. $\forall C'\in Prim(SH)$: $\exists\mu'\in MT_{SH}$ such that $\mu'\in Beh_{GB}(C,SH,\gamma,\Gamma)$ and $\mu_{C'}=\mu'_{C'}$.

Proof: by step ⟨4⟩2 and assumption ⟨0⟩:2 (definition of Δ_S).

⟨4⟩4. $\forall C' \in Prim(SH)$: $\exists \mu' \in MT_{SH}$: $\mu'_{C'} \in \Gamma_{C'}$ and $\mu_{C'} = \mu'_{C'}$.

Proof: by step ⟨4⟩3 and definition of $Beh_{GB}(C,SH,\gamma,\Gamma)$.

⟨4⟩5. Q.E.D.

Proof: by step ⟨4⟩4.

⟨3⟩2. Case: 1. $\exists C' \in Subc(C,SH)$: $C' \notin Prim(SH)$.

2. $\forall S \in SLC(C,SH) \setminus \{C\}$: $\forall C' \in Prim(SS(S,SH))$: $\mu_{C'} \in \Gamma_{C'}$.

⟨4⟩1. $\forall C' \in Prim(SH) \setminus \{C\}$: $\exists S \in SLC(C,SH) \setminus \{C\}$: $C' \in Prim(SS(S,SH))$.

Proof: by assumption ⟨3⟩:1 and the definition of $Prim(SH)$.

⟨4⟩2. Q.E.D.

Proof: by step ⟨4⟩1 and assumption ⟨3⟩:2.

⟨3⟩3. Q.E.D.

Proof: by step ⟨3⟩1, ⟨3⟩2 and induction to structure of SH.

⟨2⟩3. $\forall S \in Comp \cup Lnk$: $\mu_S \in Beh_{loc}(S)$.

⟨3⟩1. Case: $SH = CS(C,SH)$.

⟨4⟩1. Case: $S = C$.

Proof: $\mu_C \in \Delta_S \subseteq Beh_{loc}(C)$ by definition of $Beh_{BB}(C,CS(C,SH),\gamma,\Delta)$

⟨4⟩2. Case: $S \in (Comp \setminus \{C\}) \cup Lnk$.

Proof: $S \in SLC(C,SH) \setminus \{C\}$ because $\forall I \in Lnk$, $I \prec^* C$. Therefore, $\mu_S \in \Delta_S$. Thus, $\exists \mu' \in Beh_{GB}(C,SH,\gamma,\Gamma)$: $\mu_S = \mu'_S$ and $\forall S \in (Comp \setminus \{C\}) \cup Lnk$: $\mu'_S \in Beh_{loc}(S)$ by the definition of $Beh_{GB}(C,SH,\gamma,\Gamma)$.

⟨4⟩3. Q.E.D.

Proof: steps ⟨4⟩1 and ⟨4⟩2 list all cases.

⟨3⟩2. Case: 1. $\exists C' \in Subc(C,SH)$: $C' \notin Prim(SH)$.

2. $\forall S' \in SLC(C,SH) \setminus \{C\}$: $\forall S'' \in Comp' \cup Lnk'$: $\mu_{S''} \in Beh_{loc}(S'')$ with $SS(S,SH) = \langle Comp'; Lnk'; \prec'; dom'; cdom' \rangle$,

3. $S \in Comp \cup Lnk$.

⟨4⟩1. Case: $S = C$

Proof: by definition of $Beh_{BB}(C,SH,\gamma,\Delta)$, $\mu_C \in \Delta_C \subseteq Beh_{loc}(C)$.

⟨4⟩2. Case: $S \in Lnk$ such that $S \prec C''$ for $C'' \in Subc(C,SH)$, or $S \in Comp \setminus \{C\}$.

Proof: $\exists S' \in SLC(C,SH) \setminus \{C\}$: $S \in Comp' \cup Lnk'$ for $SS(S',SH) = \langle Comp'; Lnk'; \prec'; dom'; cdom' \rangle$. By assumption ⟨3⟩:2, $\mu_S \in Beh_{loc}(S)$ for $S \in Comp' \cup Lnk'$.

⟨4⟩3. Case: $S \in Lnk$ such that $S \prec C$.

Proof: $S \in SLC(C,SH)$. Therefore, $\mu_S \in \Delta_S$. Thus, $\exists \mu' \in Beh_{GB}(C,SH,\gamma,\Gamma)$: $\mu_S = \mu'_S$ and $\forall S \in (Comp \setminus \{C\}) \cup Lnk$: $\mu'_S \in Beh_{loc}(S)$ by the definition of $Beh_{GB}(C,SH,\gamma,\Gamma)$.

⟨4⟩4. Q.E.D.
Proof: step ⟨4⟩1, ⟨4⟩2 and ⟨4⟩3 list all cases.
⟨3⟩3. Q.E.D.
Proof: by step ⟨3⟩1, ⟨3⟩2 and induction to structure of *SH*.
⟨2⟩4. Q.E.D.
Proof: by step ⟨2⟩1, ⟨2⟩2 and ⟨2⟩3, $\mu \in Beh_{GB}(C,SH,\gamma,\Gamma)$. $\mu_C = LT \in Beh_{loc}(C)$, thus, $LT \in \{\mu_C \in Beh_{loc}(C) \mid \mu \in Beh_{GB}(C,SH,\gamma,\Gamma)\}$.
⟨1⟩2. $\{ \mu_C \in Beh_{loc}(C) \mid \mu \in Beh_{GB}(C,SH,\gamma,\Gamma) \} \subseteq Beh_{BB}(C,SH,\gamma,\Delta)$.
Assume: $LT \in \{ \mu_C \in Beh_{loc}(C) \mid \mu \in Beh_{GB}(C,SH,\gamma,\Gamma) \}$.
Prove: $LT \in Beh_{BB}(C,SH,\gamma,\Delta)$.
⟨2⟩1. $\exists \mu \in Beh_{GB}(C,SH,\gamma,\Gamma)$: $LT = \mu_C \in Beh_{loc}(C)$.
Proof: by assumption ⟨1⟩.
⟨2⟩2. μ is compatible for γ, $\forall C' \in Prim(SH) \setminus \{C\}$: $\mu_{C'} \in \Gamma_{C'}$ and $\forall S \in Lnk \cup Comp$: $\mu_S \in Beh_{loc}(S)$.
Proof: by step ⟨2⟩1 and definition of $Beh_{GB}(C,SH,\gamma,\Gamma)$.
⟨2⟩3. $\forall S \in SLC(C,SH)$: $\mu_S \in \Delta_S$.
Proof: by step ⟨2⟩2 and assumption ⟨0⟩ (definition of Δ).
⟨2⟩4. $\forall I \in Lnk$ such that $dom(I) = cdom(I) = \{C\}$: $\mu_I \in Beh_{loc}(I)$.
Proof: by step ⟨2⟩2.
⟨2⟩5. Q.E.D.
Proof: by step ⟨2⟩3, ⟨2⟩4, compatibility of μ and definition of $Beh_{BB}(C,SH,\gamma,\Delta)$.
⟨1⟩3. *Q.E.D*.
Proof: by steps ⟨1⟩1 and ⟨1⟩2.

Proposition 5.30. *Let C be a primitive component, let* $SH = \langle\{C\};Lnk;\emptyset;dom;cdom\rangle$ *be a structure hierarchy for C, let* $\gamma = (\gamma_I)_{I \in Lnk}$ *be a collection of compatibility relations, let* $\Gamma = (\Gamma_S)_{S \in \{C\}}$ *be a collection consisting of one set of traces such that* $\Gamma_C \subseteq Beh_{loc}(C)$ *and let* μ *be a multitrace for SH such that for all* $I \in Lnk$, $\mu_I \in Beh_{loc}(I)$. *Then:*

$$\mu \in Beh_{GB}(C,SH,\gamma,\Gamma) \Leftrightarrow \mu \in Beh_{WB}(C,CS(C,SH),\gamma,\Gamma) \Leftrightarrow \mu_C \in Beh_{BB}(C,SH,\gamma,\Gamma).$$

Proof. Proof sketch: the proof consists of straightforward expansions of the definitions. Note that, because *C* is primitive, *Lnk=Lnk'*, where *Lnk'* is the set of links in *CS(C,SH)*. Therefore, the collection of sets of link traces γ conforms to the requirements of the white box view as well as of the glass box and black box views.
Assume: 1. *C* is a primitive component,
2. $SH = \langle\{C\};Lnk;\emptyset;dom;cdom\rangle$ is a structure hierarchy for *C*,
3. $\Gamma = (\Gamma_S)_{S \in \{C\}}$ is a collection consisting of one set of traces such that $\Gamma_C \subseteq Beh_{loc}(C)$,
4. μ is a multitrace for *SH* such that for all $I \in Lnk$, $\mu_I \in Beh_{loc}(I)$.
Prove: $\mu \in Beh_{GB}(C,SH,\gamma,\Gamma) \Leftrightarrow \mu \in Beh_{WB}(C,CS(C,SH),\gamma,\Gamma) \Leftrightarrow \mu_C \in Beh_{BB}(C,SH,\gamma,\Gamma)$.
⟨1⟩1. $\mu \in Beh_{GB}(C,SH,\gamma,\Gamma) \Rightarrow \mu \in Beh_{WB}(C,CS(C,SH),\gamma,\Gamma)$.

Assume: $\mu \in Beh_{GB}(C,SH,\gamma,\Gamma)$.

Prove: $\mu \in Beh_{WB}(C,CS(C,SH),\gamma,\Gamma)$.

⟨2⟩1. $\mu \in MT_{SH}=MT_{|\{C\}}=MT_{|SLC(C,SH)}$ is compatible for γ, and $\mu_C \in \Gamma_C$.
Proof: by definition of $Beh_{GB}(C,SH,\gamma,\Gamma)$.

⟨2⟩2. $\forall S \in SLC(C,CS(C,SH))$: $\mu_S \in \Gamma_S$.
Proof: by assumption ⟨0⟩:1, $SLC(C,CS(C,SH))=\{C\}$. By step ⟨2⟩1, $\mu_C \in \Gamma_S$.

⟨2⟩3. Q.E.D.
Proof: by steps ⟨2⟩1 and ⟨2⟩2, and the definition of $Beh_{WB}(C,CS(C,SH),\gamma,\Gamma)$.

⟨1⟩2. $\mu \in Beh_{WB}(C,CS(C,SH),\gamma,\Gamma) \Rightarrow \mu_C \in Beh_{BB}(C,SH,\gamma,\Gamma)$.

Assume: $\mu \in Beh_{WB}(C,CS(C,SH),\gamma,\Gamma)$.

Prove: $\mu_C \in Beh_{BB}(C,SH,\gamma,\Gamma)$.

⟨2⟩1. $\mu \in MT_{CS(C,SH)}=MT_{|\{C\}}=MT_{SH}$ is compatible for γ and $\forall S \in SLC(C,SH) \setminus \{C\}$: $\mu_S \in \Gamma_S$.
Proof: by definition of $Beh_{WB}(C,CS(C,SH),\gamma,\Gamma)$ and by assumption ⟨0⟩:1, $CS(C,SH)=SH$.

⟨2⟩2. $\forall I \in Lnk$ such that $dom(I)=cdom(I)=\{C\}$, $\mu_I \in Beh_{loc}(I)$.
Proof: by assumption ⟨0⟩:4.

⟨2⟩3. Q.E.D.
Proof: by steps ⟨2⟩1 and ⟨2⟩2 and the definition of $Beh_{BB}(C,SH,\gamma,\Gamma)$.

⟨1⟩3. $\mu_C \in Beh_{BB}(C,SH,\gamma,\Gamma) \Rightarrow Beh_{GB}(C,SH,\gamma,\Gamma)$.

Assume: $\mu_C \in Beh_{BB}(C,SH,\gamma,\Gamma)$.

Prove: $\mu \in Beh_{GB}(C,SH,\gamma,\Gamma)$.

⟨2⟩1. $\exists \mu' \in MT_{SH}$ such that $\mu_C=\mu'_C \in \Gamma_C \subseteq Beh_{loc}(C)$, μ' is compatible for γ, and $\forall I \in Lnk$ such that $dom(I)=cdom(I)=\{C\}$, $\mu'_I \in Beh_{loc}(I)$.
Proof: by definition of $Beh_{BB}(C,SH,\gamma,\Gamma)$.

⟨2⟩2. $\forall C' \in Prim(SH) \setminus \{C\}$: $\mu_S \in \Gamma_S$.
Proof: by assumption ⟨0⟩:1, $SLC(C,SH)=\{C\}$, and $Prim(SH) \setminus \{C\}=\emptyset$.

⟨2⟩3. $\forall S \in Comp \cup Lnk$: $\mu_S \in Beh_{loc}(S)$
Proof: assumption ⟨0⟩:3 covers $S \in Lnk$ and by step ⟨2⟩1, $\mu_C \in Beh_{loc}(C)$.

⟨2⟩4. Q.E.D.
Proof: by steps ⟨2⟩1, ⟨2⟩2 and ⟨2⟩3 and definition of $Beh_{GB}(C,SH,\gamma,\Gamma)$.

⟨1⟩4. Q.E.D.

Proof: by steps ⟨1⟩1, ⟨1⟩2, and ⟨1⟩3 and transitivity of $\Rightarrow$.