Chapter 4

Overview and Running Example

Chapter 2 introduced the semantic structure, presenting a number of commitments that together determine which constructs are provided. After that, Chapter 3 presented a guideline for applying the semantic structure to model multi-agent systems. The next step in the development of the semantic structure, as stated in Section 1.4.2, is to formally describe the constructs and relations between constructs. Chapter 5 to Chapter 8 provide an elaborate and detailed mathematical description of the constructs that comprise the semantic structure. This chapter, Chapter 4, focuses on two topics that set the stage for this mathematical description in the next four chapters. First, Section 4.1 presents an overview of the mathematical description, which focuses on the mathematical concepts employed and on how these concepts are employed to meet the requirements put forward in Section 1.4.3. Second, Section 4.2 introduces a multi-agent system that is used as a running example in Chapter 5 to Chapter 8.

4.1 Overview of the Semantic Structure

Section 1.4.3 states a number of requirements for the semantic structure. The requirements can be summarised as follows. First, the semantic structure should support modelling the main activities of an agent, which are deliberation and interaction. Second, the semantic structure should be compositional, i.e. components are the most important construct provided by the semantic structure. Moreover, the dynamics of a compositional system (a system consisting of components) should be defined in terms of the semantics of the components that constitute the system together with a composition relation. Third, dynamics of a multi-agent system should be described in terms of the state of an agent. Fourth, the semantic structure should support a local view on dynamics. As many multi-agent systems are not centrally designed or managed, it should never be necessary to acquire a global picture of the structure or state of the system. However, it should be possible to compose a more global perspective from different local perspectives. (Thus, this requirement is also related to the compositionality requirement.) The way in which these requirements are addressed is explained in the next four subsections, which correspond to the next four chapters.

4.1.1 Constructs Provided by the Semantic Structure

Chapter 5 describes the main constructs provided by the semantic structure: components and information links. First, static aspects, and after that, dynamic aspects of these constructs are formalised.

4.1.1.1 Static Aspects

For a component, as stated in Chapter 2, three aspects are distinguished: the information state, or state, for short, of a component, its interfaces and its composition structure. The mathematical description of these three aspects together constitutes the component construct in the semantic structure.

A component is a locus of computation and information. The computations executed by a component modify the information contents of the component. The state of a component is determined by the information contents of the component. In a compositional system, components are likely to contain extensive data structures that together contain the information present in a component. These data structures enable the identification of substates, substates of substates, and so on, each of which is determined by a different data structure in the component. In fact, in many approaches, e.g., Troll (Jungclaus, Saake, Hartmann & Sernadas, 1996), the state of a component is defined in terms of the data structures it contains. However, the semantic structure developed in this thesis abstracts from the internal structure of a component's information (except for the identification of input, internal and output substates). Instead, it is assumed that at each moment in time, the state of a component can be identified and that for each component, a set of states (or, identifiers of states), is given. Moreover, in Chapter 5 to Chapter 8, in which the semantic structure is developed, no language is defined to specify sets of states. Instead, the semantic structure is developed without any reference to syntactic constructions. However, in Chapter 9, formal languages are developed that enable precise specification of sets of states and other sets that are assumed to be given, in the context of an application of the semantic structure.

As stated above, it is assumed that for *each* component, a set of states *of that component* is given. Such states are called *local states* to emphasise that these states are only determined by the information contents of a single component, and by nothing outside that component. To support locality, the semantic structure as defined in Chapter 5 does not refer to a notion a global state, that is, a state determined by the information contents of more than one component. Instead, a more global view of dynamics within a compositional system is defined in a compositional way, as compositions consisting of local states.

The other two aspects are formalised as follows. For each component, three substates of the state of the component are distinguished: the input, internal and output substates. These substates are determined by the information contents of the input and output interfaces of the component, and the component itself. The compositional structure of a component is formalised by a tuple, called a *structure* *hierarchy*, consisting of a set of component identifiers, a set of link identifiers, a relation that represents the hierarchical structure of components, their subcomponents, the subcomponents of subcomponents, and so on, and two functions that map each link to the component or link connected to each of its end points.

The formalisation of information links is largely determined by the commitments presented in Section 2.2. The basic idea is that information transmission from one component to another establishes a relation between possible states of the two components. This idea appears both in the static aspects of an information link as well as the dynamic aspects. From a dynamic point of view, information transmission establishes a relation between the behaviour of both components. This is discussed in Section 4.1.2. The static aspects of an information link consist of which states of the link itself are distinguished (commitment discussed in Section 2.2.3.2), the components to which it is connected (commitment discussed in Section 2.2.3.1) and how the link is intended to relate the behaviour of the two components it connects. Thus, similar to a component, for a link a set of local states is distinguished. The state of a link is determined by the state of information transmission as a process (e.g., a link can be busy transmitting information, or waiting for information to transmit) and possibly by the contents of the link (messages in transit). However, as for component states, the semantic structure abstracts from the internal structure of states. Instead, it is assumed that for each link, a set of states is given. In addition to this set, a relation, called the *information link mapping*, is distinguished. As the information link mapping describes the intended relation established between the two components it connects, the information link mapping is defined on the state sets of these two components. States in the state set of a component are states that may or may not occur in the behaviour of a component. Thus, an information link mapping may describe the relation between two components by reference to states that do not actually occur in each behaviour of the components.

As an example, consider a compositional system with a link I which transmits information from a component D to a component C. The state set of D contains a state identified by S_1. The state set of C contains a state identified by S_2. Suppose that the following requirement is imposed on the compositional system: if D reaches state S_1, then C should reach state S_2. This requirement is part of the static aspects of I and is described by its information link mapping. The formalisation of information link mappings is presented in Section 5.1.2.

4.1.1.2 Dynamic Aspects

After the formalisation of static aspects presented in the first half of Chapter 5, the second half presents the formalisation of the dynamic aspects. First, the local behaviour of each component and link is defined. The local behaviour of a component or link consists of a set of alternative behaviours that the component or link could exhibit if it were to exist in isolation. In other words, the local behaviour is the behaviour from a strictly local point of view, neglecting constraints imposed by a component's or link's relation with other components or links. The local behaviour of a single component or link is defined as a set of so-called local component or link traces, which are linear or branching structures of consecutive information states of the component.

Second, given the local behaviour of a set of components and links, the actual behaviour of a component or link in relation with other components and links is defined as a structure consisting of elements from the local behaviours of its constituents. The basic idea is that information transmission *constrains* the local behaviour of the components involved in the transmission, and that information transmission induces a *relation* between local traces of components that defines the constraints imposed by information transmission. Such a relation, called a *compatibility relation*, provides a more global view of co-operating components. (However, no global state and / or global language is assumed.) The basic idea is employed as follows.

For a component, three sets of traces can be distinguished, as depicted in the top half of Figure 4.1 below (in which traces are assumed to be linear). The first set is the set of traces consisting of all possible combinations of states of a component. This set includes traces that, in practice, can never be acquired, because they do not fulfil the specification of the behaviour of the component. For instance, consider a component B that is able to provide, upon request, a service 's' that, at some point in time, places new output in the output interface of B. The set of all possible traces of component B includes traces in which, after receipt of the request for a service 's' in the input interface, the result is never placed in the output interface of B.

A subset of the set of all possible traces is the set of local component traces: those traces that could, in principle, be acquired because they fulfil the specification of component behaviour. These traces are called local component traces to emphasise the fact that they are only part of a component's behaviour from a purely local point of view, in which constraints imposed by interaction with other components are not taken into account. For instance, a trace in which a service request 's' is visible in the input interface, and in a following point, the result of this service is visible in the output interface, is a local component trace. However, this trace might not be an actual behaviour of component B in a compositional system: it can only be an actual behaviour if another component, A, generates a request for service 's' and if this request is actually transferred to B. Therefore, in a structure that models the behaviour of the system consisting of component B, a component that requests B's service and interaction between A to B, such constraints should be represented.

As stated above, the behaviour of a composition of components and links is defined in terms of the local behaviour of the constituents. To represent constraints imposed by information transmission, compatibility relations are defined between traces of components that exchange information. Only local component traces and link traces that are related by compatibility relations can constitute the dynamic structures associated with components. Thus, a local component trace of a component that transmits information to another component is part of the *actual overall behaviour* of the system only if it respects constraints imposed by information transmission. Only compatible local component and link traces are included in multitraces that model the behaviour of compositional systems. Only local component and link traces that respect information transmission are compatible.

As an example, consider a system with components A and B introduced above, and an information link between A and B. The information link automatically relays a request for B's service from A to B. In this example, a trace of component A in which event 's' is in A's output interface is compatible with traces in which event 's' is in B's input interface (and the result of service 's' is at some point in time generated at its output interface).

The bottom half of Figure 4.1 depicts similar sets of components of another component. As indicated in the picture, these sets are, in general, disjoint from the three sets of components of component A. An information link from component A to B establishes a compatibility relation between local component traces of both components. The two sets of compatible local component traces are the traces related by compatibility.

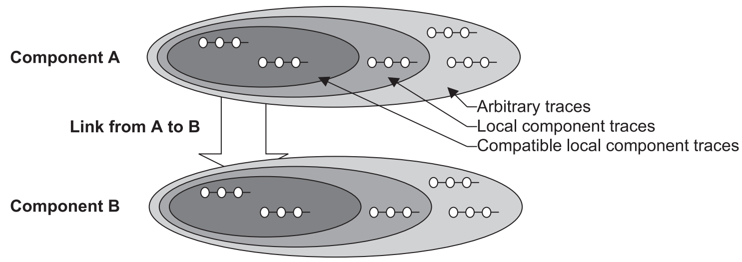

Figure 4.1: Sets of component traces.

In fact, three views on the behaviour of compositional systems are distinguished, called the black box, white box and glass box views. From a bird's-eye view, the structures that constitute these views consist of local component traces and link traces of a (possibly composed) component itself, possibly its subcomponents and its links, and possibly their subcomponents and links, and so on. All three views on behaviour developed in this section are relative for a given structure hierarchy and collection of compatibility relations. In a *black box view*, the behaviour of *C* is defined as a set of local component traces of *C* only. As a consequence, in the black box view the behaviour of subcomponents and links is not visible (although their behaviour is taken into account in the definition of the black box view to determine which local component traces of *C* constitute behaviour if information exchange is taken into account). In a *white box view*, the behaviour of a component not only consists of local component traces of this component, but also of local component traces of its subcomponents. In a *glass box view*, the behaviour of a component consists of local component traces of this component, of local component traces of its subcomponents and of local component traces of the subcomponents of these subcomponents, and so on.To summarise, the two most important formal notions are the structure hierarchy and the glass box view on the behaviour of a component. A structure hierarchy enables defining an entire compositional system, 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. Locality and compositionality play an important role in the formalisation of the dynamics of a compositional system. The starting point for describing the dynamics of a compositional system consists of structures (local component and link traces) that describe the behaviour of a component or link from a strictly local point of view. Three views on the dynamics of compositions of components and links are then defined as compositions of local component and link traces. Only local component and link traces that obey constraints enforced by information transmission can be part of these compositions. The constraints enforced by information transmission are represented by compatibility relations, which are briefly described in the next section, Section 4.1.2, and defined in Chapter 6

4.1.2 Interaction: Locality and Compatibility

The definition of properties of compatibility relations is the topic of Chapter 6. Compatibility relations are introduced in Chapter 5 as ternary relations on the set of traces of two components or links and a link. Any relation on the sets of local component traces of the domain of a link, the link and its co-domain is a possible compatibility relation. In Chapter 5, no properties of compatibility relations are defined.

In Chapter 6, specific classes of compatibility relations are defined in terms of the properties that compatibility relations in a specific class exhibit. These properties are expressed in terms of transmission octets. Transmission octets provide a bridge between, on the one hand, 8-tuples of states that occur in local component and link traces of a link I, its domain and co-domain, and, on the other hand, the (static) declaration for the information link I of how information transmission affects the states of the components and links involved. This (static) declaration itself is given by the information link mapping of I. An information link mapping does not refer to states that occur in component or link traces. Instead, as the following definition indicates, an information link mapping is directly defined in terms of the sets of all states of I, its domain and co-domain.

Figure 4.2 illustrates compatibility relations and transmission octets by zooming in on Figure 4.1. A triple consisting of a trace for a component A, for a link L from A to a component B and for B is shown. Within this triple of traces, octets of eight states (two from the trace for A, four from the trace from L, and two from the trace for *B*) can be found that respect information transmission as depicted in Figure 2.7. Such an octet is a transmission octet, and is depicted in Figure 4.2 by black elipses. The triple of traces is part of a compatibility relation with a specific property if transmission octets can be found for a specific subset of all states in the traces of the triple. (E.g., for all states in the trace for *A*, a transmission octet can be found.)

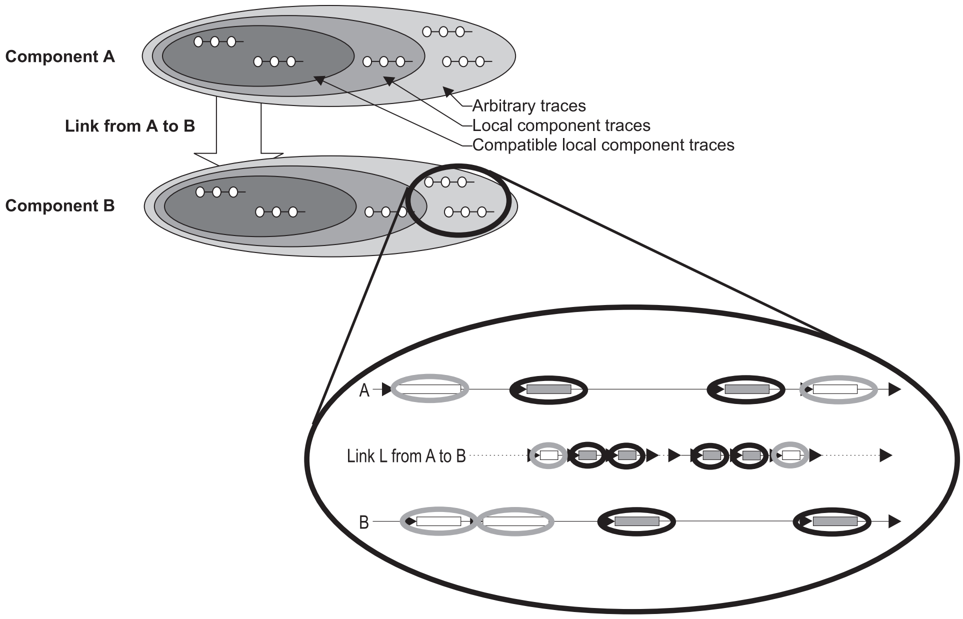

Figure 4.2: A transmission octet.

Transmission octets are employed to define properties that a compatibility relation, and thus information exchange, may exhibit. Four properties have been defined so far: the lossless transmission property, the order-preserving transmission property, the asynchronous transmission property and the logically instantaneous transmission property.

4.1.3 A Global Perspective

As explained in Section 4.1.1.2, the dynamics of a compositional system are described by compatible multitraces, which are indexed sets of local component and link traces. Three views on the dynamics of a compositional system are defined. Given a structure hierarchy that describes a compositional system, the glass box view provides the view with the most detail: the multitraces that constitute the glass box view contain local component and link traces for each component and link in the structure hierarchy. In this sense, the glass box view provides a global perspective on the behaviour of the compositional structure described by the structure hierarchy.

However, the global perspective provided by the glass box view consists of local component and link traces, each of which consists of local component and link states. Thus, this global perspective does not describe global dynamics in terms of a notion of a global state. However, in some cases, it is necessary to have available a description of the global dynamics in terms of the global state of a compositional system.

A common way to define a global state is as a composition of the local states of all components and links at the same moment. However, this definition relies on the availability of global time to determine 'the same moment' for all components and links. As stated in Chapter 1, this thesis assumes that global time is not available. In Chapter 7, a different way to define global states is developed.

In Chapter 7, a global state is defined in terms of *snapshots*. A snapshot is itself relative to a compatible multitraces for a structure hierarchy *SH* as introduced in Section 4.1.1.2. A snapshot is a function from the set of components and links in *SH* to the (disjoint union of) the sets of states of all components and links, such that the snapshot selects exactly one local component or link state from each local component or link trace in the multitrace. Figure 4.3 depicts a snapshot.

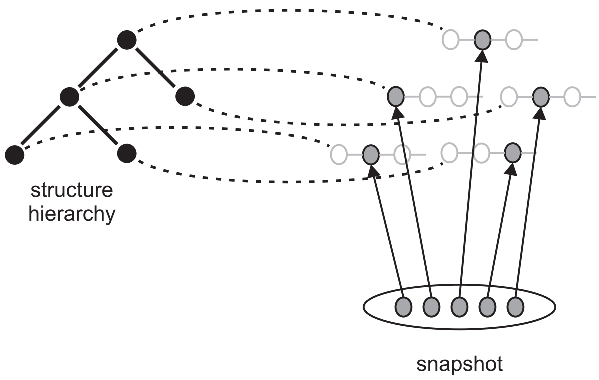

Figure 4.3: Structure hierarchy.

A snapshot is thus an arbitrary selection of states from a multitrace. Not every snapshot of a multitrace for *SH* represent a global state of the behaviour of the compositional system described by *SH*, because in an arbitrary selection of states, possibly for a component *A*, a state is selected in which information is available that is, according to the state of another component *B* in the same snapshot, not yet sent. In other words, the selected state of *A* depends on the occurrence of state *B*. Dependence was first introduced (in an event-based context) by Lamport (1978). In the semantic structure developed in this thesis, dependence is not a primitive notion. Instead, it is defined in terms of transmission octets. Loosely speaking, a state σ_A of component *A* depends on a state σ_B of component *B* if either (1) A=B and σ_B occurred at an earlier time point in the trace of *A* than σ_A, or (2) in σ_A, information is received that is, according to a transmission octet, made available in σ_B (in other words, σ_B and σ_A are the first and eighth state of a transmission octet, respectively), or (3) there is a state σ_C of a component C such that σ_A depends on σ_C and σ_C depends on σ_B.

Two states σ_S and $\sigma_{S'}$ are independent if neither σ_S depends on $\sigma_{S'}$ nor $\sigma_{S'}$ depends on σ_S. States that are independent can possibly occur simultaneously. In Chapter 7, a global state is defined as a snapshot such that for two different components or links S and S', the states σ_S and $\sigma_{S'}$ in the snapshot are independent. In this definition, it is not assumed that global time is available. Instead, the definition only refers (indirectly) to information transmission as represented by transmission octets.

Even if the traces in a compatible multitrace for a structure hierarchy *SH* are linear, the set of global states as introduced above has the structure of a partial order. This partial order of global states constitutes the global perspective on the behaviour of the compositional system represented by *SH*. The partial order of global states can also be viewed as a transition system. Starting from the global state consisting of all initial component and link states, the compositional system proceeds as follows. Each component or link is either involved in updating its internal state, transmitting information to another component or link, or receiving information. Each of these activities results in transitions from one local state to the next local state. In the transition system, the local states in the global state are updated accordingly.

4.1.4 Control

As stated in Chapter 1, agents are socially able. Consequently, in almost every multi-agent system, agents try to influence other agents (by information transmission) to satisfy goals they cannot satisfy without help. Likewise, in a compositional system, some processes try to influence other processes. In other words, processes in a compositional system and agents in a multi-agent system try to exercise *control* over other components or agents, respectively. Chapter 8 is devoted to the representation of this phenomenon, the phenomenon of control, in the semantic structure.

Chapter 8 starts with a characterisation of the control phenomenon. Control (at least, phenomena with that name) is encountered in programming language design, but also in reactive Artificial Intelligence systems that assist in operating, for instance, a power plant. However, all these instances of the control phenomenon share a common characteristic: they repeatedly carry out a process that can be described conceptually as follows. A component that wants to exercise control over another component builds a descriptive model of the past and present behaviour of the component that is to be controlled. It does so on the basis of information obtained from the controlled component. The controlling component extends the model of the controlled component such that the extension prescribes the future of the controlled component. This prescriptive future part of the model presumably specifies the future as the controlling component would like it to be. On the basis of this prescriptive part, the controlling component determines which actions to take, or which information to transmit, to (try to) set the actual future of the controlled component as envisioned.

Chandrasekaran (1994) argues that everything that exercises control carries out a process to do so that can be described at a conceptual level as above. (Chandrasekaran does not imply that, e.g., a thermostat, which controls a heater, actually maintains a model of the heater and its behaviour. However, the process carried out by a thermostat can be conceptualised as if it does.)

The essential element in Chandrasekaran's characterisation of control is information transmission. A component is able to control another component (at least in principle) by the virtue of information transmission. Information transmission enables trying to set the future of a controlled component as well as observing the controlled component to evaluate the exercition of control. In terms of the semantic structure developed in this thesis, Chandrasekaran's characterisation of control shows that the constructs of the semantic structure as described in Section 4.1.1 and Section 4.1.2 suffice to represent control. However, there are some additional issues that have to be addressed.

First, the constructs presented in Section 4.1.1 and Section 4.1.2 do not enable the distinction of information transmission for control as a special form of information transmision. Chapter 8 introduces a number of refinements of the constructs presented in Section 4.1.1.1 that enable the separation of control processes (these processes are represented by components as usual) and information transmission for control. These refinements are technically straightforward and therefore not discussed in this section. It is important to note that Chapter 8 does not define which information is control information. Instead, Chapter 8 introduces facilities that enable applications of the semantic structure to designate specific information as control information.

Second, Chapter 8 discusses the relation between the behaviour of a control component and controlled components. Conform the general characterisation of control, a controlling component transmits information to a controlled component to (try to) set the future behaviour of the controlled component. Control processes differ with respect to the extent to which the future behaviour of the controlled component will be as specified. For instance, compare the thermostat and the president of the national bank. On the one hand, it may be expected that if a thermostat transmits information to a heater (by closing an electric circuit) stating that the heater has to start, the heater will indeed start (not taking into account malfunctions). On the other hand, if the president of a national bank tries to influence consumers' spendings, it may well be the case that his or her information is misunderstood, neglected or nullified by other information directed at the consumers. The semantic structure, nor the refinement presented in Chapter 8, commits itself to a specific extent to which received control information determines the future behaviour of a controlled component. Instead, applications of the semantic structure can choose a level of extent suitable for the application. The final section of Chapter 8 discusses a few possible relations.

The semantic structure is fully defined in Chapter 5, Chapter 6 and Chapter 7. Chapter 8 presents formal definitions of a number of refinements of constructs defined in Chapter 5. (Chapter 8 also presents a number of additional commitments with respect to control and discusses how control in a multi-agent systems is represented in a compositional system.) Chapter 9 also contains a number of formal definitions. However, the constructs defined in Chapter 9 are not part of the semantic structure. Instead, they constitute the semantics of the DESIRE modelling framework. The formal definitions of the constructs that constitute the semantic structure are illustrated with a running example, which is described in Section 4.2.

4.2 Running Example

An example system in the area of intelligent Internet applications is used to illustrate the semantic structure developed in this thesis. This example system is a multi-agent system in which an information brokering agent serves user agents with information from information provider agents. Information brokering is introduced in Section 4.2.1. Section 4.2.2 presents a compositional system that models information brokering in a multi-agent system. This compositional system is the actual running example.

4.2.1 Information Brokering

The multi-agent system on which the example is based consists of broker agents that act as intermediaries between agents that provide information (on arbitrary resources) and agents that use this information. (Agents that use this information are customers of the broker agent. Provider agents can, however, also be viewed as customers: the broker agent presents information on their behalf. To make the distinction between the two clear, agents that use the information provided are called ‘user agents’.) Figure 4.4 depicts the three types of agents. To keep the example concise, only one broker agent is depicted (called ‘Broker’), together with two agents that provide information (called ‘Provider 1’ and ‘Provider 2’, respectively) and two users of the broker agent (called ‘User 1’ and ‘User 2’). As indicated in Figure 4.4, the user agents exchange information with both the broker agent and the provider agents. The reason for this is that the broker agent only provides information on a resource, and not the resource itself. To obtain the resource itself, user agents contact the provider agents directly.

In fact, the terms used to describe the agents in the example system indicate the *roles* of the agents distinguished, and not their types. In other words, in the domain of information brokering, one can distinguish agents that play the role of information provider and others that play the role of information users. These roles need not be the only roles played by the agents: for example, agents may also play roles that have nothing to do with the domain of information brokering and are thus not of importance in the context of this example. Moreover, one or more agents may have more than one role: a provider may present information obtained as a user of another broker agent. In the example system, however, an agent has one specific role and is named accordingly. Thus, for example, an agent with the role of a provider is called a provider agent.

The five agents depicted in Figure 4.4 are the only agents distinguished in the running example. Figure 4.4 deliberately depicts the agents as different forms of agents: the user agents appear to be humans, while the broker appears to be an automated system. The provider agents appear to be institutions or agencies (a store and a school). The running example, however, abstracts from the form of the agents: human agents are not distinguished from software agents, and subagents that comprise the agencies (e.g., store clerks, school teachers) are not identified. Moreover, the running example also abstracts from the means of communication used by the agent. Figure 4.4 suggests that the human agents access the broker agent via a public terminal. In this case, communication between the user agents and the broker agent consists of manipulations using the (graphical) user interface of the broker agent. However, it may also be the case that the users employ their own personal computers to access the broker agent.

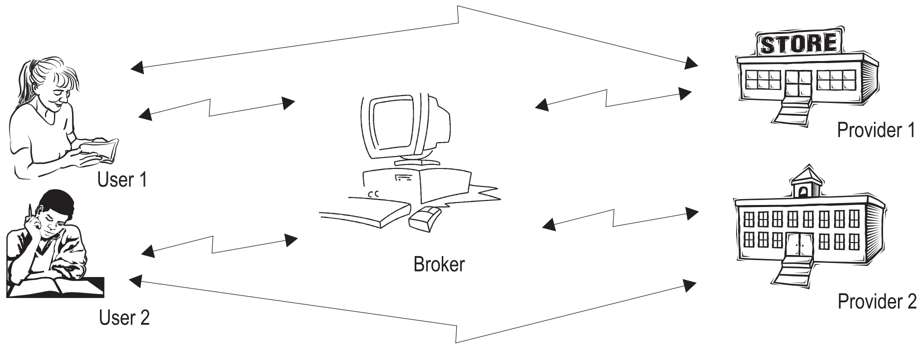

Figure 4.4: Information brokering.

The intended function of the broker agent is as follows. A user agent communicates to the broker agent that he/she is interested in information on a resource (i.e., a research paper, a WWW page, a product sold by means of e-commerce). A provider agent communicates to the broker agent descriptions of resources available to the broker agent, preferably using an internet standard such as the Resource Description Framework (RDF), see (Lasilla, 1998). The broker agent matches interests of users with information provided by the provider agents, in compliance with a number of requirements. To keep the running example concise, only one requirement is given:

- Once a broker agent receives a query from a user, information matching the query has to be communicated to the user at the next moment in time if this information was already available to the broker, or some time in the future in all other cases.

Many more requirements can be placed in the design of the multi-agent system. However, as the study of information brokering is not the topic of this thesis, and to keep the example concise, no further requirements or detail are provided. See (Jonker & Treur, 1998c), for a more complete investigation of information brokering. (In Chapter 8, an additional requirement is presented, which is used to illustrate control.)

4.2.2 Design of a Compositional System for Information Brokering

The semantic structure illustrated by the running example provides constructs for building compositional systems, such as the multi-agent system presented in Section 4.2.1. Applying the guideline put forward in the previous chapter to the example multi-agent system results in the compositional system presented in Figure 4.5. The compositional system depicted in Figure 4.5 consists of eight components:

- One component, called toplevel, of which all other components are subcomponents, or subcomponents of subcomponents, and so on. This component represents the complete multi-agent system.
- The components labelled user_1 and user_2 represent the user agents. (As stated in Section 4.2.1, the precise nature of these components is not important: user_1 and user_2 can, for example, represent human agents, their personal digital agents, or web browser applications serving as intermediaries between the user and the broker.)
- The component labelled broker represents the broker agent. For the broker agent, two subcomponents are distinguished: Agent Specific Processes (ASP) and Own Process Control (OPC). The subcomponent Own Process Control represents all subprocesses of the broker agent that control its behaviour. All other subprocesses, which are specific to the task of the broker, are subprocesses of the component Agent Specific Processes. In the next chapter, subcomponents of a subcomponent of toplevel makes the examples more interesting. In Chapter 8, OPC and ASP are used to illustrate control in a multi-agent system.
- The components labelled provider_1 and provider_2 represent the information provider agents. (Similar to user_1 and user_2, the precise nature of these components is not important.)

As stated in Section 4.2.1, the user agents, the broker and the information provider agents preferably employ an Internet standard such as the Resource Description

Format (RDF, Lassila, 1998) for communication. However, while this standard defines the format of resource descriptions, it allows different ontologies to be used for descriptions. The running example, in fact, assumes that on the one hand the broker and information providers and, on the other hand, the users use different ontologies.

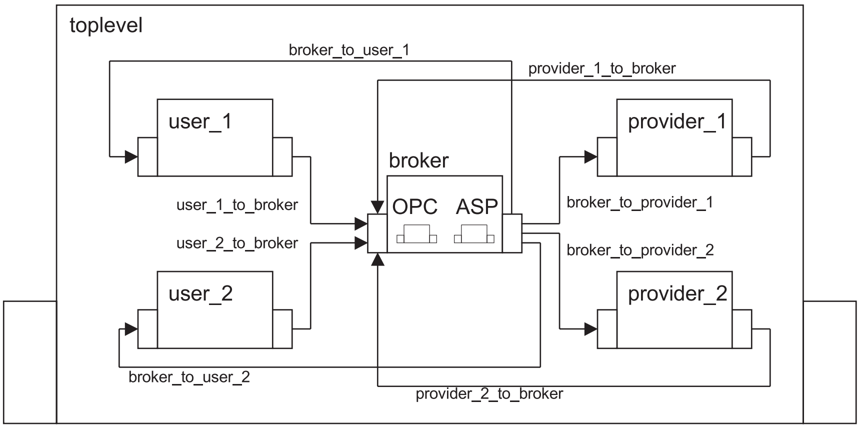

Figure 4.5: Information broker agent system.