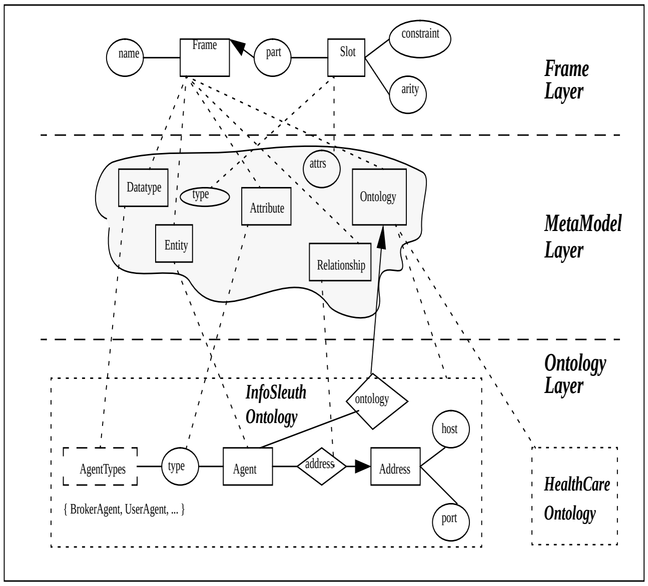
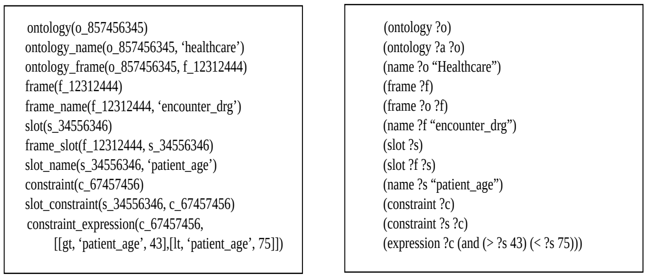


Figure 3: The three-layer ontology model



```
ontology(o_857456345)
ontology_name(o_857456345, ‘healthcare’)
ontology_frame(o_857456345, f_12312444)
frame(f_12312444)
frame_name(f_12312444, ‘encounter_drg’)
slot(s_34556346)
frame_slot(f_12312444, s_34556346)
slot_name(s_34556346, ‘patient_age’)
constraint(c_67457456)
slot_constraint(s_34556346, c_67457456)
constraint_expression(c_67457456,
        [[gt, ‘patient_age’, 43],[lt, ‘patient_age’, 75]])
```

```
(ontology ?o)
(ontology ?a ?o)
(name ?o “Healthcare”)
(frame ?f)
(frame ?o ?f)
(name ?f “encounter_drg”)
(slot ?s)
(slot ?f ?s)
(name ?s “patient_age”)
(constraint ?c)
(constraint ?s ?c)
(expression ?c (and (> ?s 43) (< ?s 75)))
```

Figure 4: Multiple representation of same ontology

of the system are its instantiations. However in the case of the InfoSleuth ontology, the instantiation “InfoSleuth” of the ontology object is also a part of the InfoSleuth ontology. This is required as the InfoSleuth ontology is the ontology associated with the broker agent.

4.2 Utilization of Multiple Representations of Ontologies

One of the reasons for representing ontologies is the ability to reason about them. For this purpose, different agents might represent them in different languages depending on the type of inferences to be made. Figure 4 shows an example of the same piece of ontology represented by the resource agent in KIF and by the broker agent in LDL. The broker agent uses this representation to determine whether a resource agent is relevant for a particular query.

The Broker Agent utilizes a representation of the ontology exported by the Resource Agent (shown in Figure 4) in LDL [38]. The deductive mechanisms of LDL help determine the consistency of the constraints in the user query and those exported by the Resource Agent which in turn determines the relevance of the information managed by Resource Agent. The Resource Agent, on the other hand, translates this information into KIF expressions (as shown in Figure 4), and sends them to the Broker Agent.

5 Brokering in InfoSleuth

One of the valuable new features of the InfoSleuth technology is an intelligent brokering system that performs semantic as well as syntactic brokering of resources. Each agent in the system advertises its capabilities to the Broker Agent. The advertisements specify the agent’s capabilities in terms of one or more of the ontologies. From the user’s perspective, semantic brokering enables requests to be specified in terms of the concepts in an ontology, and matches those semantic concepts to the resources that are currently best suited to handle those specific requests.

5.1 Capabilities Enabled by Semantic Brokering

Semantic brokering helps expand the functionality of InfoSleuth in the following ways.

Intelligent Routing. Through the use of brokering, InfoSleuth offers the ability to route information requests based on content, through the use of constraint matching on the ontology a resource claims expertise over. For instance, a resource may have access to information only about doctors in Houston and Austin. It would be fruitless to query this resource about doctors in Dallas and the use of constraints rules this resource out.

Currently constraint matching is an intersection function between the user query and the data resource constraints. If the conjunction of all the user constraints with all the resource constraints is satisfiable, then the resource contains data relevant to the user request. We should mention here that, following “the open world assumption”, the Broker Agent always matches a query with unconstrained, yet relevant data sources, regardless of the constraints imposed by the query.

Note that the constraints for both the user request and the resource data profiles are specified in terms of some common ontology. It is the use of this common vocabulary that enables the dynamic matching of requests to applicable resources.

Dynamic Binding of Resources. An InfoSleuth broker accepts advertisements from new resources and notifications of resource unavailability at any time. Thus, InfoSleuth is able to keep up with an ever changing set of resources, which is not easily accomplished in a federated database. As resources come and go, the broker is made aware of this through KQML advertisements, and will thus only recommend appropriate resources to the agents doing the query planning. This means that the same user request may produce different results at different times, depending on which resources are available. Also, neither the user nor any agents acting on his behalf needs to know where or what resources are available when building a query plan, i.e. the user can query an open information space.

Scalability. There are several ways in which our approach to brokering impacts system scalability. First, decisions on which resources are likely to be relevant to specific user requests are made without actually accessing the resource. This greatly reduces the time and effort required to route a request. Secondly, the ease with which new resources may be added to the system makes scalability much less of an issue. To add a resource to the system it need only have a KQML/KIF interface for advertising its services; then other agents can make use of them immediately. Thirdly, as the