their systems are well suited to this as the meanings of fields in a data level system are easy to use in locally defined ways; of course this further complicates the problem, as these sorts of local modifications require local tweaks to the interfaces and ultimately produce an interface that marginally accomplishes the intended purpose, is not well understood, and is brittle and difficult to maintain as the corresponding systems evolve.

Information Level Interface

An information level interface differs from a data level interface in several regards. Primary among these is the requirement for the systems being interfaced to be information centric rather than data centric. Information centric systems are based on explicit ontologies that model the underlying semantic entities of the domain rather than the data crunched by the currently favored domain processes or displayed on the screens of particular applications. The developers of an information level interface consider all the information to be exchanged (parts and shipments in this case) in a singular context which not only relates the entities to be exchanged to each other but to the context in which the entities are related at both ends of the interface. This is show in Semantic Layer of the Information Level System Interface depicted in Figure 4 by an Interface Ontology that overlaps into the Supply and Maintenance Domains. The Interface Ontology is marked up in terms of the shared (public) Supply and Maintenance ontologies and vice versa.

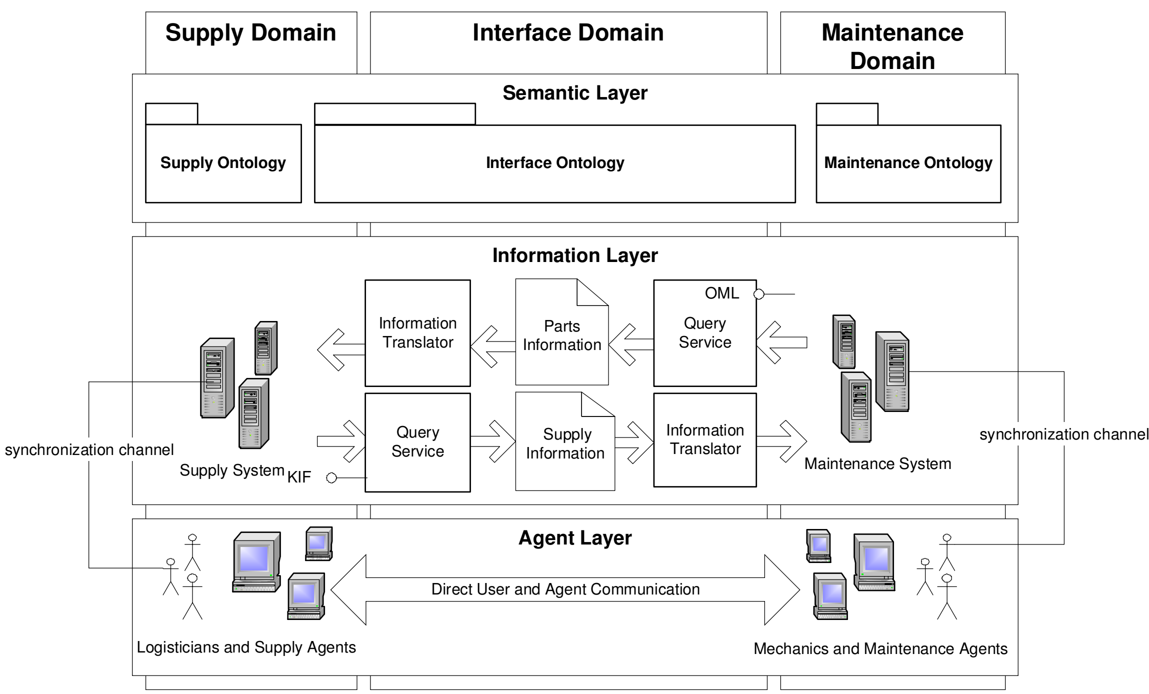

Figure 4: Information Level System Interface

The interface ontology itself will now consist of multiple interrelated entities derived from a Upper Level Interface Domain Ontology that provides higher level semantic context to each entity type concretely defined and used in the interface proper. The Interface Ontology should also define the entities required to pull the interface information from the context of one system and to place it into the context of another, although these constructs may not be present in the physical implementations that transport the information from one system to another it is important that they are defined in the ontology to fully capture the semantic context of the information. The Upper Level Interface Domain Ontology may have existed prior to the development of the interface or may have been developed in conjunction with it. In either case it is important that the application level ontologies specific to the individual Supply and Maintenance Domains in turn utilize it directly or at least reference it by semantically marking up the entities in the ontologies of these domains to correlate them to the concepts defined by the Upper Level Interface Domain Ontology.

With a semantic layer thus defined the information level interface can do much more than generate simple fixed reports. Each system can expose a much more generalized query interface. The queries are formulated and the responses returned in terms of the entities defined in the interface domain ontology. This allows for a much more flexible interface that is more likely to survive evolving interface and system requirements over time. Note that in order to support a generalized query interface at least one additional interface ontology must be defined that defines the semantics of the queries, or commands that in turn uses the interface domain ontology as logical arguments. For this purpose, many well defined standards exist such as Structure Query Language (SQL) and Knowledge Interchange Format (KIF), or systems may expose their own proprietary but publicly defined interface such as the Object Management Layer (OML) employed by many of the systems developed by CDM Technologies.

Between the information and agent layers in Figure 4 are depicted synchronization channels. In this example the Maintenance and Supply systems are information centric systems that provide for the development of software agents by providing subscription services to client applications. A subscription service is key to agent development as it lets agents register for the ontological patterns that trigger it to action. In this manner agents can always be operating in support of their users, as they are always ready to act in fulfillment of their responsibilities without having to perform needless busy work querying the information store for conditions that may never arise.

Information Level Interoperability

The Information level interface of the previous section has a shortcoming in that the Supply System and Maintenance system must both explicitly query each other to receive new information. Whether or not any new information is available a query must still be run just to find out. On the flip side, immediately after a query has been run information could change in the source system that would not be reflected in the querying system until after processing the next query which may take awhile depending on the polling scheduled employed. This situation can be remedied by employing the same sort of synchronization channel used between the individual information centric systems and the agents they support that is show in Figure 5. The addition of a synchronization channel for the interface allows for the development of interface brokers. Interface brokers serve as agents in the systems they support by automatically synchronizing the state of the system to the state of interest in an external system via the defined interface between the systems.

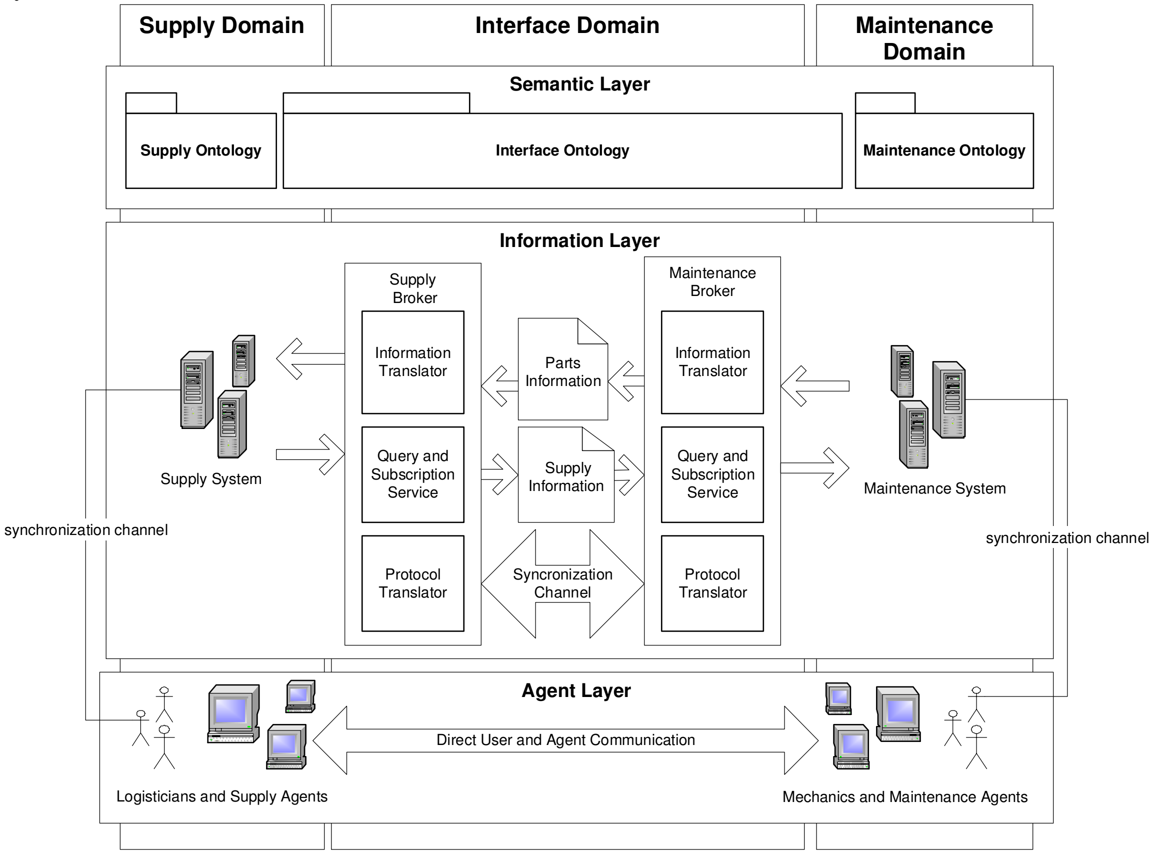

Figure 5: Information Level System Interoperability

This approach allows for true interoperability between the systems but is not without its own difficulties. Many of the entities exchanged between the systems correlate to items in the real world and thus have unique identities whose keys must be managed within the confines of a real system implementation. In this sort of information level interface this is typically accomplished by designating a single specific source for each type of unique entity. While this approach works well for interfaces in which only a few systems are participating in starts to break down in larger interoperability scenarios as each system broker must know about all the other systems participating and which system is designated to be the definitive source of which data. This approach requires much duplication of effort within the individual data brokers and introduces an undesirable coupling between the systems. One approach for dealing with this is the introduction of an interoperability server.

Interoperability Server

An interoperability server elevates the interfaces between systems to the level of information centric systems themselves. This approach provides one common implementation of the individual system data brokers that know in which system or combination of systems to find information defined within the Interface Domain

Ontology. It also provides specifically for the management of unique entities that are shared across two or more of the interfacing systems.

Employing the concept of an interoperability server leads to a system of systems architecture that groups collections of systems that need to regularly exchange information into loosely coupled federations whose central hub consists of an specific instance of an agent based interoperability server that is configured to address the specific needs of the federation. This concept views the interoperability server as just another system which allows one to layer a hierarchy on this system of systems architecture where higher level federations may include as systems zero or more interoperability servers typically from lower level federations. Within the defense domain, one could envision the proposed system of systems hierarchy following along the lines of existing unit hierarchies within the individual services with a the top level of the hierarchy operating at the level of the joint chiefs of staff, and commander in chief. Of course, crossties between the individual units and services at the lower levels of the hierarchy may also exist. The end state of this vision is a single, albeit large and distributed, system of systems that incorporates the entire information infrastructure of the DOD. This system of systems is tailored to meet the specific needs of user communities at all levels, by utilizing the systems specifically developed to meet their local needs, and is adaptable to change due to the loose coupling between systems.

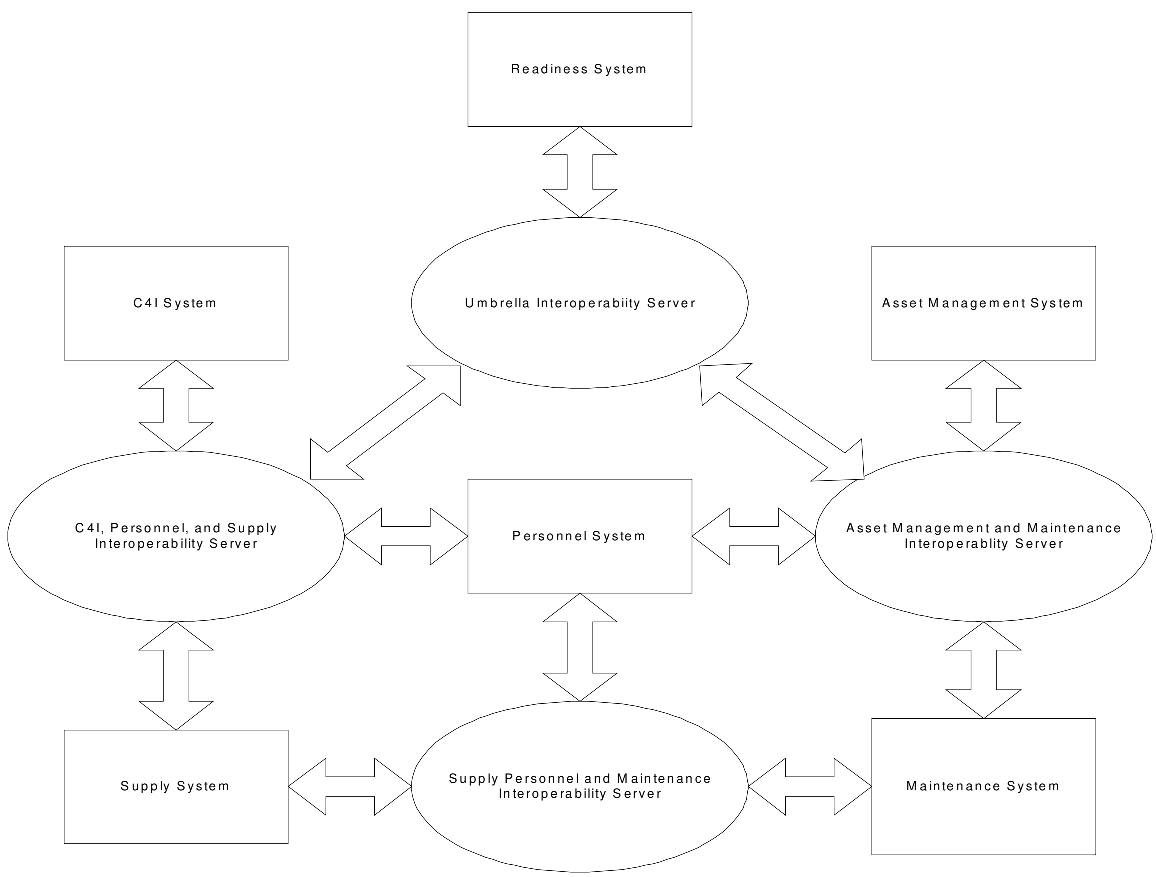

Figure 6: Interoperability Server

There will be several distinct ontologies associated with each Interoperability Server. System Interface Ontologies that are unique to each system participating in the federation will be used to define the interrelated logical constructs within the corresponding system that are targeted to participate in external interactions. For example, the System Interface Ontology for an air load planning system may include constructs to represent air transports, stow areas, and cargo items. Also associated with each participating system is an ontological map that defines the transformations required to translate information represented in the corresponding System Interface Ontology both to and from the Federation Interface Ontology. Federation Interface Ontologies that are unique to each federation will be used to define the interrelated logical constructs with which the client systems to the Interoperability Server may interact. This ontology defines all the information of common interest to the entire federation as opposed to the specialized interests of the individual systems that are participating in it. For example, the Federation Interface Ontology for a joint logistics transport federation, which interfaces specialized air, rail, and sea load planning systems may define a transport construct which is a generalization of the specialized air transport, rail transport, and sea transport constructs that may be defined by individual System Interface Ontologies of the participating systems. This ontology once established serves as a standard for the domain represented by the federation similar in concept to the enterprise models that were popularized in the late eighties and early nineties such as the DOD Logical Data Model but exist within a more manageable scope and are driven by the interoperability requirements of the federation. Finally, an Interoperability Ontology shared by all Interoperability Servers will be used to define the interrelated logical constructs associated with interoperating systems and the services provided by the interoperability server. These constructs are independent of the logical domain entities associated with a specific federation. For example, the Interoperability Ontology may define constructs such as query, constraint, system, and ontology.

Summary

The key to the interoperability between systems lies with well-defined system and interface ontologies. An ontology makes explicit the conceptualizations used and shared by the interoperating systems. The shared conceptualization is known as the interface domain ontology. The interface domain ontology and the individual system domain ontologies should both be well marked up in terms of each other and ideally share both an upper level and lower level interface domain ontology. In this manner, the mappings that determine the context of interfaced entities on either side of the exchange are made explicit and are more likely to endure evolutionary changes to the systems and local modifications or special case usages. For systems to truly interoperate rather than just interface some sort of synchronization channel must be provided. As the number of systems participating in an interface grows even a well-designed information level interface can become unmanageable and an interoperability server approach should be considered.