HETEROGENOUS DATABASE INTEGRATION USING AGENT-ORIENTED INFORMATION SYSTEMS [♦]

J. Todd McDonald, Michael L. Talbert, and Scott A. DeLoach
Air Force Institute of Technology
Graduate School of Engineering and Management
Department of Electrical and Computer Engineering
Wright-Patterson Air Force Base, OH 45433-7765
jeffrey.mcdonald@afit.af.mil, 937-256-1917 (Presenter)
michael.talbert@afit.af.mil., 937-255-6565 x4280
scott.deloach@afit.af.mil, 937-255-6565 x4284

Keywords: Agents, Military Applications, Modeling and Simulation, Data Integration

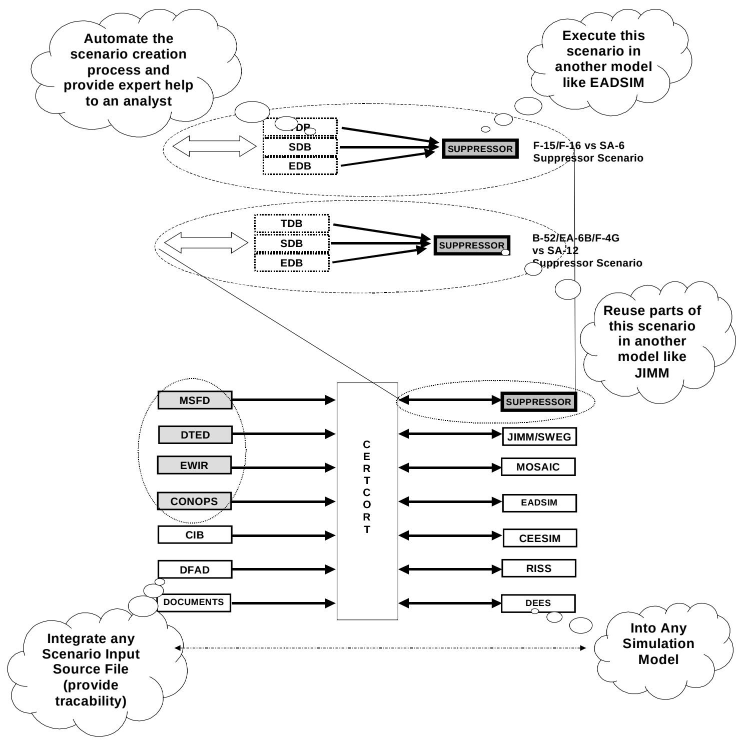
Abstract

The Department of Defense has an extensive family of models used to simulate the mission level interactions of weapon systems. Interoperability and reuse of the underlying data files used to create simulation scenarios are of particular interest to this community. Figure 1 presents a pictorial overview of the data integration problem domain itself and how both model to model and input-source-data to model correlation is required. Approaches to schema and data integration originate from the traditional field of federated database research, though these solutions tend to be data-oriented and not application-oriented. The emerging field of agent-oriented information systems (AOIS) views data as the central focus of an application while providing an overall framework for deriving system architecture. Combining object-oriented data modeling, a persistent programming language and an agent-oriented analysis and design methodology, we achieve key goals of simulation interoperability relevant to this problem domain. A multi-tiered agent framework is developed and its capabilities to handle reusability and integration of scenarios across simulation models are demonstrated by prototypical applications. Object-oriented modeling techniques are used to encapsulate and organize the syntactic information contained in simulation scenario database files while the semantic information of these objects is examined for data integration purposes. The agent architecture provides a communication layer to support collaborative development and a distributed environment for information brokering.

The reutilization problem across simulation models arises when the information contained in scenario database files needs to be re-used in another scenario of the same model or as entities in a domain-similar model. To compound this problem, no single input source file may be directly

[♦] The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government.

responsible for describing a scenario entity, and the scenario entity itself may be distributed over multiple data blocks in a set of input files. Currently, the process of mapping data input files into a scenario file (Figure 1) is done with minimal software support, is very tedious, and is done by one analyst at a time, with no facility for collaborative assistance from other domain specialists.



Automate the scenario creation process and provide expert help to an analyst
Execute this scenario in another model like EADSIM
OP
SDB
EDB
SUPPRESSOR
F-15/F-16 vs SA-6 Suppressor Scenario
TDB
SDB
EDB
SUPPRESSOR
B-52/EA-6B/F-4G vs SA-12 Suppressor Scenario
Reuse parts of this scenario in another model like JIMM
MSFD
DTED
EWIR
CONOPS
CIB
DFAD
DOCUMENTS
CERTCORT
SUPPRESSOR
JIMM/SWEG
MOSAIC
EADSIM
CEESIM
RISS
DEES
Integrate any Scenario Input Source File (provide tracability)
Into Any Simulation Model

Figure 1: Heterogeneous Database Problem Domain

Agent technology itself offers a different paradigm for looking at problems such as information reuse, data source combination, and intelligent construction of scenarios. Automation and reduced human effort can be achieved when intelligent user interfaces are combined into the framework as well. Information agents in our framework are used in terms of simulation model integration for two key purposes:

- Provide encapsulation, representation, and access to the information of any given set of scenario database files (from JIMM/SWEG, SUPPRESSOR, EADSIM, etc.)
- Provide encapsulation, representation, and access to the information contained in any given source input file (EWIR, MSFD, DTED, CONOPS, etc.)

We use an agent-oriented analysis and design methodology (MaSE) to analyze this simulation specific problem domain and derive a set of agent classes with appropriate conversations to support goal-directed behavior. In terms of application demonstration, three primary agent types are implemented and defined further in terms of system capability. These three types form the core of an “information layer” and include the information requestor, provider, and broker agents. External "layers" of agent types that support *collaboration* and *intelligent construction* are envisioned. Traditionally, a *semantic* or *ontological* broker is used to provide the "point of reference" to terms and types of information in an agent brokering system because it makes judgments about information requests according to a set of predefined ontologies that an information gathering system is based upon. Our framework envisions scenario reuse and traceability to input data sources by finding semantic levels of relationships from the object models that encapsulate and represent these data sources. The "ontology" for scenario reuse is thus based on the definition of and relationships between various levels of these semantic objects (Figure 2).

We implement an architecture that allows persistent objects to be both created and retrieved from an ObjectStore object-oriented database management system (OODBMS). Persistence in regards to this framework is seen as orthogonal to the agent-oriented information system, yet support is provided by the framework itself for OODBMS access. Figure 2 illustrates the separation of concerns across these two architectures.

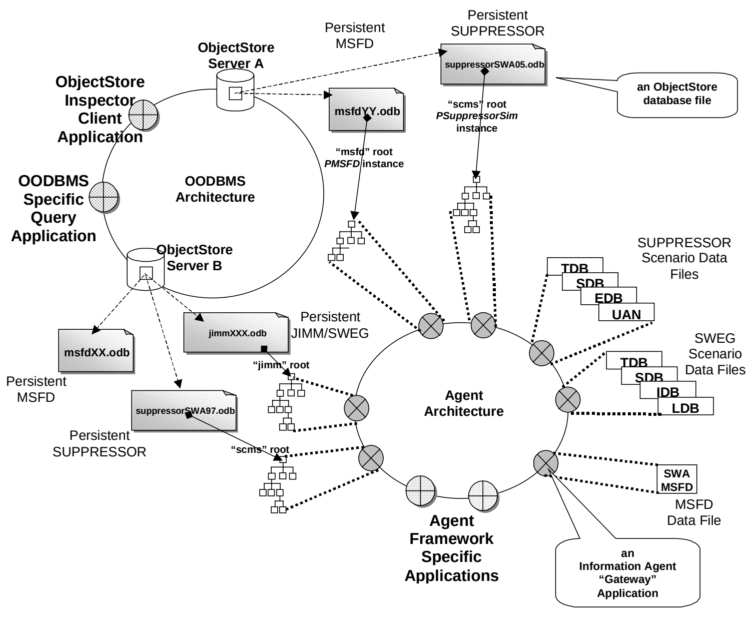


Figure 2: Orthogonality of OODBMS Architecture and Agent Framework

The OODBMS architecture only deals with persistence-capable objects (scenario component instances) stored in database files accessible by an OODBMS server. In contrast, the agent framework is based on the existence of common semantic objects (and their instances), the information content of which all participating agents agree. Support is explicitly built into *provider* agent applications to "provide" information in the context of the agent system, but the underlying source is either 1) the original textual scenario files; 2) a persistently stored object representation of a scenario or data input file; or 3) a relational data source. In terms of the agent architecture, the source of the information that an agent provides is unknown, and for ultimate reuse purposes, inconsequential. Agents can “broker” information on different levels, and our research uses common object representations such as instances of specific input source or scenario object models. Figure 3 illustrates the distributed, cross-platform nature of the framework along with representative applications functioning in different “roles” within the information layer of the agent architecture.

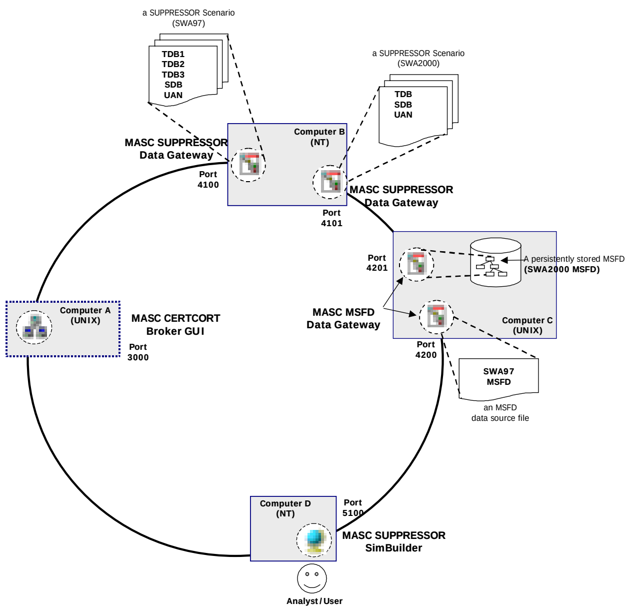


Figure 3: Application Demonstration

Agents provide unique benefits to information integration in this context above those provided by traditional heterogeneous database architectures.

- Semantic models in this domain require post-processing of instance data; this is best supported in the context of an “active” data source that information agents can provide.
- Federated databases tend to be “data” centric and not “application” centric. Multi-agent systems provide a life cycle approach that can provide direct traceability of user requirements into system components and agent classes.
- AOIS technology keeps the “focus” of system development on the data without binding to a particular data storage mechanism.
- Agents provide the ability to abstract away the underlying data representation of information sources within information systems.
- Agent based systems can be expanded to provide greater functionality without drastic architectural changes. Intelligent interfaces and the ability to achieve coordinated plan-based goals are not possible from a database-centered approach to systems development.

- Scenario model integration and construction has certain information retrieval aspects that are naturally suited to underlying information agent architecture. AOIS has implementation in terms of both information-gathering systems and the encapsulation of traditional data sources normally part of a database management system.