Data Access and Integration Architectures

M. Cassandra Smith, Ph.D.
Elizabeth D. Zeisler
Michael W. Hooper
The MITRE Corporation
1820 Dolley Madison Blvd.
McLean, VA 22102
(703) 983-6703
mcsmith, ezeisler, mhooper@mitre.org

Abstract

This paper reports the results of a study to survey data access and integration architectures being used or considered for use within the domain of DOD Logistics. The goal of the study was to provide a management level briefing on current approaches to data access and integration. One similarity of the architectures surveyed is reliance on middleware to consolidate access to heterogeneous environments: for interoperability, distributed transactions, distributed components, WEB/internet integration, legacy integration, and mainframe qualities of service, which include the need for availability, integrity, scalability, and reliability. It was apparent that no single technology (architecture) will be sufficient.

Key words - Architecture, data access, data integration, heterogeneous, interoperability, logistics, mediation, middleware

1. Introduction

This paper reports the results of a study to survey data access and integration architectures being used or considered for use within the domain of DOD Logistics. The logistics components of the military Services and the Defense Logistics Agency (DLA) collectively own over a thousand automated information systems. Increasingly, there is a need to share information among the Services, DLA, and CINCs to support requirements for the Joint Task Force commander and requirements at the national level needed to sustain operations, e.g., demand planning for spare parts. This often requires that data be assembled from multiple disparate sources and processed (aggregated and integrated) into new information to support planning and execution of combat support missions. The goal of the study was to provide a management level briefing to the joint oversight boards within Logistics on current approaches to data access and integration in order to identify areas of joint concern. The briefing was to answer the following key questions:

- What is the data integration requirement implied by Joint Vision 2010 (JV2010) and Defense Science Board (DSB) studies?
- What are the pros and cons of architecture approaches being used in the combat support community?
- What are the community level risks?

This paper focuses on the second of these questions. In addition, the study had a motivation to provide the Logistics community situational awareness on current state of the art as background for future decisions.

2. Approach

A two-year outlook was taken, focusing on architectures supported by commercially available products or having apparent high interest in the commercial market place. Secondly, the study relied on the

significant expertise and in-house experience with the particular architectures under study. The following summarizes the steps used for the study:

- Describe high level information access and integration requirement
- Identify predominant approaches (not specific solutions)
- Identify strengths and weaknesses of each approach
    - Major capabilities and characteristics
    - Development and support considerations
    - Examples of where used in community
- Identify similarities and differences focusing on community impacts
- Develop conclusions

3. Technology Areas investigated

The study team brainstormed technology areas that might have a high impact on data integration and access. This led to a decision to include the following general areas in the study: data management architectures, messaging architectures, and component ware architectures. The team thought it beneficial to designate whether a specific technology was current technology or emerging for the future. The areas surveyed under data management architectures were distributed database architectures, data mediator architectures, and data warehouse architectures. Messaging architectures included message brokers, Extensible Markup Language (XML)/Electronic Data Interchange (EDI), and intelligent agents. The component ware architectures included distributed object and enterprise Java Beans architectures.

3.1 Data Management Architectures

Figure 1 provides a framework for discussing data management architectures. We considered all of the data management technologies current technologies. A distributed database management system (DDBMS) architecture is probably the baseline for providing access to data in different locations. In a DDBMS architecture the data may be distributed over a network. However, every data fragment must be in the same data model (e.g., relational or object-oriented), using the same language. The Logistics domain needs to provide access to logistics data that is managed by different Services and Agencies in DOD. Each may have an existing database and each database may be managed by a different DBMS. In such an environment a DDBMS would be applicable only if the Services and Agencies standardized on a single DBMS and redid each database for that DBMS. This would require each Service or Agency to give up autonomy over its database, and it might prove costly to redo the databases and purchase a new DBMS.

A data mediator architecture would be more appropriate where organizations involved desire autonomy and the ability to interoperate while avoiding rebuilding the database or purchasing a new DBMS. Figure 2 shows a generalized data mediator architecture (derived from [8]). Using database middleware and relying on an integrated schema, the data mediator architecture provides transparent access to heterogeneous data sources.

The data mediator architecture is appropriate for managing disparate databases on disparate systems for day to day operational needs. However, when decision support and large complex queries and analyses are needed, a data warehouse may be appropriate. As figure 3 indicates, a data warehouse is built by extracting data from the operational data stores, then transforming and cleansing it.

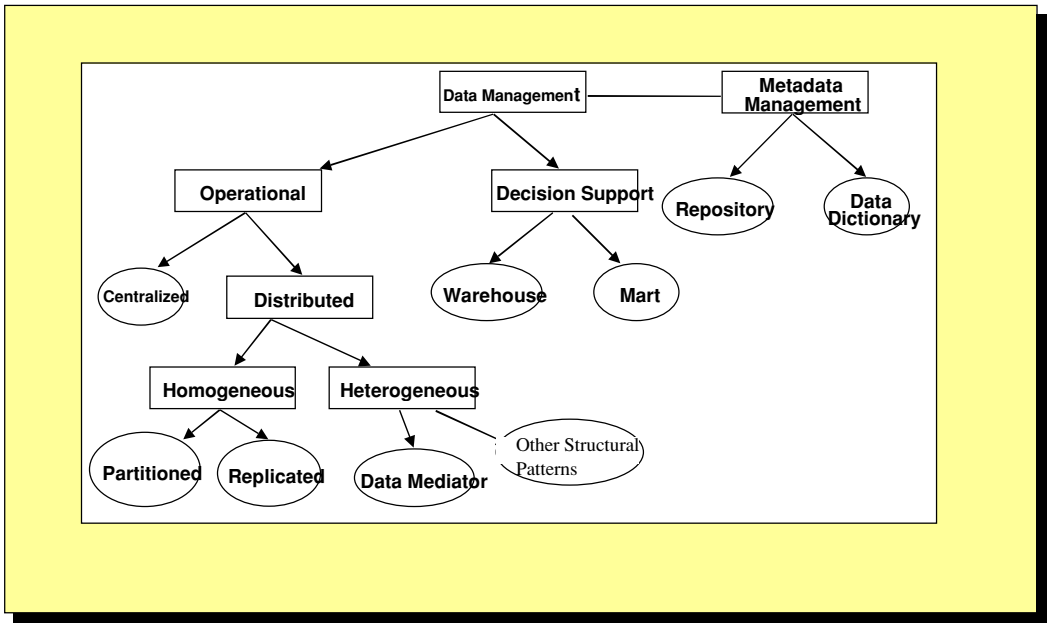

Figure 1. Data Management Framework

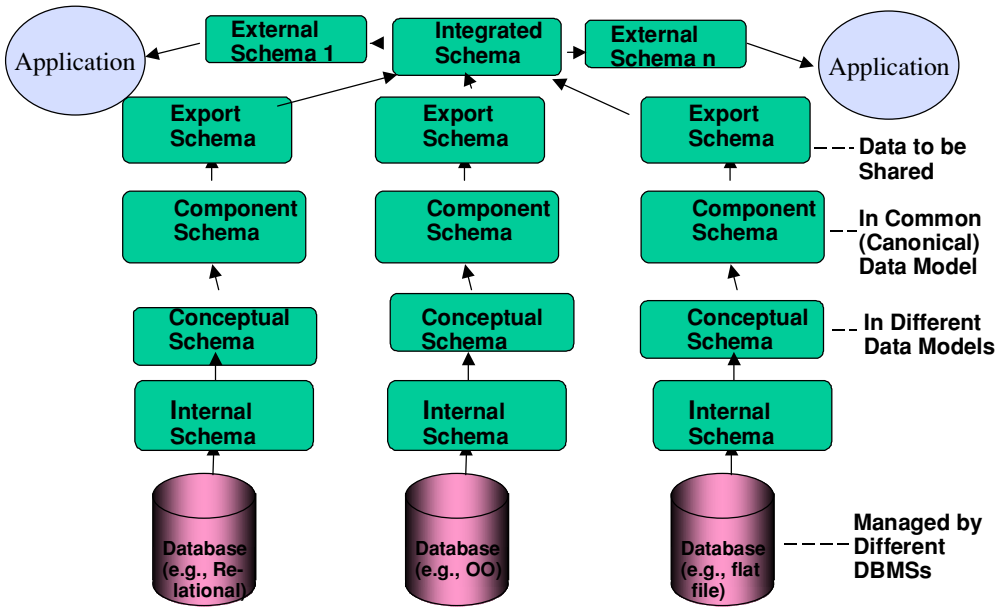

Figure 2. Generalized Data Mediator Architecture

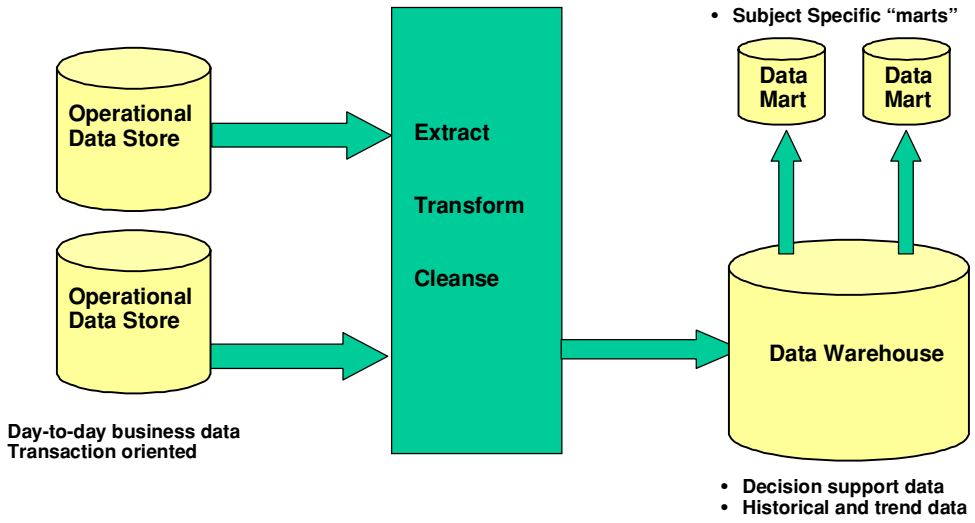

Figure 3. Data Warehouse Architecture

3.2 Messaging Architectures

Figure 4 is an overview of a messaging architecture. The messaging approach differs from direct data access in that data is accessed or created through applications that interact by passing messages between them. With messaging, access to data is indirect as user requests are serviced by an application, which in turn accesses the data. The application has the necessary business logic to effectively manage transactions that result in the updates of data. Applications can cooperate through sharing of messages. A message is a file (practical limit 2-4MB in size) sent electronically, either synchronously or asynchronously, between two computers on a network. Metadata includes the application (message) protocol, which is defined by two communicating entities. It defines the formats of messages exchanged between them, the states and which messages can be exchanged when the entities are in the respective states.

Architectures supported by current technologies include message brokers (message-oriented middleware or message-queuing software), Email, Electronic Data Interchange (e.g., ANSI X12), and batch or streaming file transfers, e.g. MPEG Video. Architectures supported by emerging technologies include Extensible Markup Language (XML) for publishing structured "documents" over the internet and Agent Mediation for providing mediated data access.

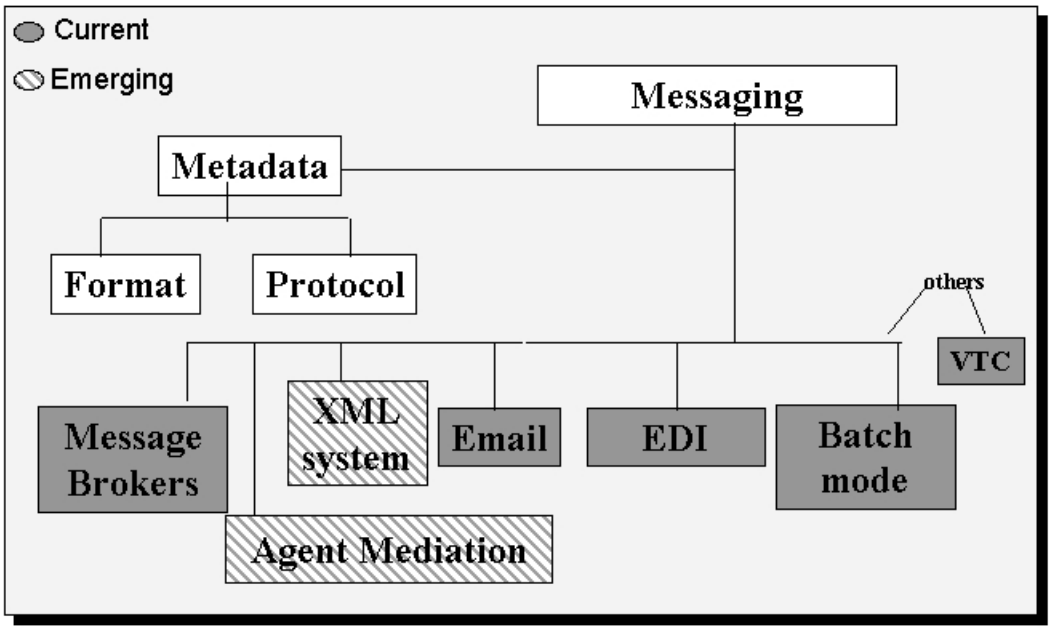

Figure 4. Messaging Architecture Overview

3.3 Component Ware Architectures

Figure 5 demonstrates a component-based architecture. According to [9] a software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties. According to [11] a component has the following properties:

- Configurable via properties/resources
- Connectable with other components via visual builder tools or scripting languages
- And (for purposes of this paper) it adheres to one of these evolving standards:
    1. JavaBeans, Enterprise JavaBeans
    2. The CORBA Component Model: "CORBABeans"
    3. Microsoft ActiveX, COM+

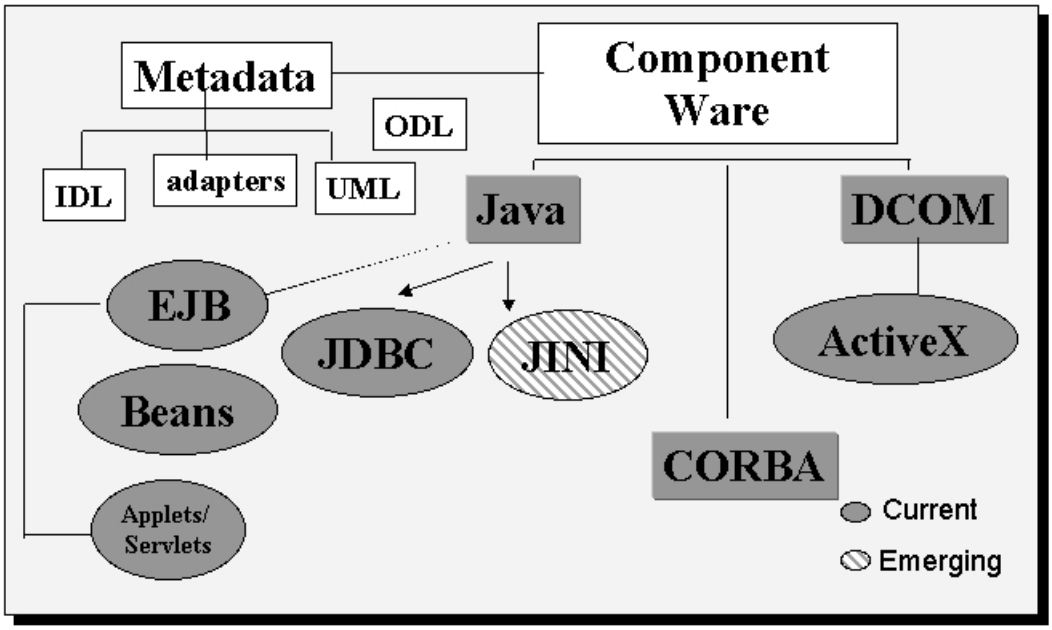

Figure 5. Component Ware Architecture Overview

4. Analysis

One similarity of the architectures surveyed is reliance on middleware to facilitate interoperation among heterogeneous systems: for interoperability, distributed transactions, distributed components, WEB/internet integration, and legacy integration. It is also apparent that no single technology (architecture) will be sufficient. Further, elements of a particular architecture style are being combined to form new architecture breeds, e.g., use of XML to communicate between databases and web servers.

We consider the data mediator architecture the most likely data management approach used today to provide interoperability. Accordingly, we found the following in contrasting data mediator and messaging architectures. Data mediator architectures demonstrate the following characteristics:

- Provide data integration
- Demonstrate slow performance
- Require less programming
- Resolve integrity issues
- Have semantic capability

Messaging architectures tend to have the following characteristics:

- Provide application integration
- Demonstrate good performance
- Require more programming
- Provide transaction update

- Enable business logic reuse

Table I summarizes the technologies surveyed and the motivations for selecting a particular technology as well as the general risks. Table 1 is based on [5].

Table I
Application of Architectures

IF YOUR MOTVATION IS	CONSIDER THESE TECHNOLOGIES	BUT RECOGNIZE THESE RISKS
Global applications w/ distributed updates, scalable	• DDBMS • Message Broker	• System administration • Configuration management
Ad Hoc query of integrated data, near real-time	• Data Mediator	• Performance, scalability • Interface management
Ad hoc query of voluminous data for mining and forecasting	• Data Warehouse • Data Marts	• Costs to model, populate, and reorganize warehouse
Integration of COTS packages or legacy applications	• Message Broker • Distributed Objects	• System administration, CM • Cost to adapt applications
Migrate legacy applications to utilize common S/W infrastructure	• Distributed Objects • Application Frameworks	• High development costs • Performance
Electronic Commerce / Electronic Date Interchange	• Mediation Agents, XML • Application Frameworks	• Immature tools • High development costs
User assistance and information monitoring/control	• Mediation Agents • Messaging	• High development costs • Performance

5. Conclusion

As expected, multiple architectures and supporting technologies are being used across the logistics community. This confirms that no general solution exists and that architecture choice must be matched to specific requirements, e.g., ad hoc query versus updates. Weakness with a particular technology is not a problem if it can be used (integrated) with other technologies that fill that void. However, integration risks must be identified and managed. An architecture provides a blue print for identifying technical risks and for easing integration by lending focus to interfaces and identifying areas of commonality and heterogeneity. Further, while most architectures and supporting technologies provide general data management capabilities, data mediation and integration still require significant analysis and development effort. That is, no technology provides an "out of the box" solution for legacy data access and data integration. In addition, access architectures come with significant development and support considerations. Therefore, life cycle cost (total cost of ownership) must be one of the decision factors.

References

[1] Bosak, J. and T. Bray, "XML and the Second-Generation Web," *Scientific American,* http://www.sciam.com/1999/0599issue/0599bosak.html#link1, May 1999.

[2] Gartner Group, *Message Brokers: A Focused Approach to Application Integration, Part 1*, 1996.
[3] Glossary XML, http://www.xmledi.com/repository/xml-rep.htm.
[4] Hurwitz, J., "Sorting Out Middleware," *DBMS*., January 1998.
[5] Percy, A. and A. Cushman, *Data in Its Place*, Gartner Group, 1999.
[6] Percy, A, *The Dysfunctional Middleware Family*, Gartner Group., June 1996.
[7] Ritter, D, "The Middleware Muddle," *DBMS*., May 1998.
[8] Sheth, A. and J. Larson, "Federated Database Management Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys,* Vol. 22, #3, 1990.
[9] Szyperski, Clemens, "Software Components," European Conference on Object-Oriented Programming (ECOOP) , 1996.
[10] Thuraisingham, B., *Data Management Systems: Evolution and Interoperation*, CRC Press: Boca Raton, 1997.
[11] Vecellio, Gary, Internal Brief, The MITRE Corporation, 1999.
[12] Webber, D. and XML/EDI Group, "XML/EDI and the Universal Data Element Framework (UDEF), http://www.xmledi.com/repository/xml-rep.htm, April 1998.
[13] Wise, H, "Developing and Using Data Warehouse Metadata," *Proceedings DAMA/Metadata Conference*, April, 1999.

Author Biographies

Dr. M. Cassandra Smith is a senior information systems engineer at the MITRE Corporation in McLean VA. Her interests include database management systems, especially object-oriented and relational. She holds a Ph.D. degree in computational linguistics from Georgetown University.

Ms. Elizabeth D. Zeisler is a project leader and principal scientist at the MITRE Corporation in McLean, VA. She holds a BFA degree from Cornell University and MMS from American University. Ms. Zeisler has participated in numerous symposia and refereed journals, including March-April 1999, "Intelligent Interdomain Distributed Service Management " International Journal of Network Management," Volume 9, June 1998, "A Vision and Framework for CORBA/JAVA Beans Convergence", and presentation to Object Management Group Technical Committee.

Mr. Michael Hooper is a lead information systems engineer at the MITRE Corporation for the C2 and Combat Systems Engineering department. He has over 13 years of experience related to information technology management and application to defense logistics. He is currently providing technical support to the Defense Logistics Agency (DLA) and to the Deputy Under Secretary for Logistics DUSD(L). Mr. Hooper has a BS in Computer Science.