

Session 1: Middleware Selection

Stanley M. Sutton Jr.

IBM T. J. Watson Research Center
30 Saw Mill River Road, Hawthorne, NY 10532 USA
(suttonsm@us.ibm.com)

1. Introduction

An increasing variety of middleware systems is available for use in enterprises today. Two widely used but very different middleware styles are object-oriented and message-oriented. Within each of these styles, there are multiple products to choose from. Moreover, any of these products may be used alone or in combination with other products. Thus the problem of middleware selection is increasingly important in the engineering of enterprise software systems.

Middleware selection, construed broadly, is the determination of middleware to be used in a software development or integration project. The middleware may already exist, in which case selection reflects an intention to acquire it, or the middleware may not yet exist, in which case selection implies an intention to develop it.

Middleware selection is important for a number of reasons. It is an essential part of the way in which distributed systems get built, both by new development and by integration of existing applications and services. Moreover, middleware is a key enabling technology: it provides services, supports application functions and features, separates concerns, and integrates components. In these roles, middleware interacts with and may impact many other kinds of technologies, such as database systems, workflow engines, web servers, and applications. It further affects system architecture and development processes.

Middleware selection is also challenging, for these same reasons. Middleware selection can be an involved software (and systems) engineering process in its own right, with all the technological, organizational, economic, and political aspects that this may imply. Because of the central position and critical function of middleware, if it is selected or applied inappropriately, it can become a key *disabling* technology.

The two papers in this session provide broad views on middleware selection for enterprise-scale distributed systems. “A Key Technology Evaluation Case Study: Applying a New Middleware Architecture on the Enterprise Scale”, by Michael Goedicke and Uwe Zdun of

the University of Essen, Germany, describes a method for evaluating and selecting middleware in an enterprise context. It emphasizes the importance of understanding and communication among all stakeholders in the enterprise system, notably management and engineers. It also emphasizes the enterprise-specific nature of the middleware evaluation and selection process.

“An Architecture Proposal for Enterprise Message Brokers”, by Jörn Guy Süß of the Technische Universität and Michael Mewes of the Fraunhofer ISST, both in Berlin, Germany, defines and motivates an enterprise architecture based on message brokers. This work also represents a kind of middleware selection process, since the motivation for the message broker architecture arises from failure of other kinds of considered middleware to effectively address enterprise integration needs.

Both of the papers focus on communication issues as key concerns in middleware selection. Interestingly, each paper arrives at a different primary communication style: synchronous in the former case, asynchronous in the latter. This divergence of results, both of which are well justified, only begins to suggest the diversity of concerns, issues, and approaches that may characterize middleware selection.

To begin to organize these topics, we considered the following general questions:

- What is the role of middleware selection in the engineering of distributed object systems?
- What is the process of middleware selection in the engineering of distributed object systems?

2. What is the Role of Middleware Selection?

When selecting middleware we ultimately want to know what process we should use. However, the most appropriate selection process depends on the role of middleware selection in the overall systems development or integration process.

The role of middleware selection depends on the current and projected shape of the “enterprise system landscape.” The main elements shaping this landscape are:

- The proportion of integration of existing applications and systems versus development of new applications and systems
- The level and scope of the organization or enterprise over which middleware is to be applied. This may vary from individual projects and work groups, up through individual departments, to span departments within an enterprise, and even to integrate distinct enterprises

- The time frames for middleware selection, implementation, and use
- The risks, costs, and feasibility of alternative approaches

Integration projects may present more or fewer challenges for middleware selection, depending on the diversity of systems and applications to be integrated. For new development, the constraints on middleware may be initially less restrictive, but middleware selection must be closely integrated with the design of other elements of the system, so selection can still be complex.

Typically, the larger the scale of the effort for which middleware is being selected, the more numerous and complex the issues affecting middleware and thus the middleware selection process. However, complicating and simplifying factors may occur at any level or scale of project.

The time frames for middleware selection, implementation, and use affect the potential selection process in several ways. If the period for selection is short, then the role of selection may be relatively limited, for example, to identify an approach based on a restricted criterion (such as flexibility, familiarity, or cost) or to consider just a small number of alternatives. If the period for selection is longer, then the role of the selection process may be to compare many technologies, identify and consider a wide range of selection criteria, and so on. If the implementation period is short, then the selection process and its results may be constrained, and an approach oriented toward acquisition may be favored over one oriented toward implementation. If the implementation period is longer, then the selection process can be open to considering and recommending a wider range of technologies and processes. If the period of use (over which the cost of middleware will be amortized) is short, then the selection process should focus on short term concerns and rapid gains and the selection and implementation processes should be constrained accordingly. If the period of use is to be long, then the selection process must focus on long term concerns and benefits, and the selection and implementation processes may be opened up accordingly.

Risks, costs, and feasibility affect all systems development and integration efforts to some degree, but any of these may be especially important in particular cases. Consequently, the role of middleware selection may emphasize risk reduction, cost control, or feasibility assessment.

The above concerns primarily reflect a technical viewpoint. However, organizational factors (including social factors) can be just as important in determining the role of the middleware selection process. Such factors include economics, politics, business processes, business relationships, organizational history, personnel, and so on. A primary function of the middleware selection process may be to identify stakeholders in the

middleware selection process and to educate or inform them regarding technical and other concerns relevant to the selection process. Stakeholders may include project managers, development engineers, business partners, customers, third-party systems integrators, consultants, and others, with very different interests, goals, and levels or areas of expertise. According to their stakes, each of these groups may see middleware selection in a different role and have different objectives for it.

Of course, organizational as well as technical concerns can affect practically any aspect of the enterprise system landscape: the degree of integration versus new development, the level and scale of software projects, the time scales on which projects operate, and the perspectives on risk, costs, and feasibility. Standards (or the lack of them) are also a critical consideration at the interface between the technical and non-technical elements of enterprise system engineering. There are choices involving standards that are open or closed, official or *de facto*. Just as standards help to achieve integration and interoperation on technical plane, they can also help to promote communication and understanding on the social plane. The role of middleware selection must be defined with respect to the use of standards in these ways.

Finally, the role of middleware selection must be defined with respect to the rest of the software life cycle. Middleware selection should generally be based on system requirements specifications (that account for both technological and organizational factors), but middleware selection may not (and probably should not) be wholly determined by those specifications. Depending on circumstances, middleware selection may occur prior to, along with, or subsequent to system architectural specifications. Consequently, middleware selection may constrain, or be constrained by, architectural design.

Generally, depending on the variety of technological and organizational factors outlined above, middleware selection can be a relatively independent or dependent process. The potential variety of interdependencies between middleware selection and other development or integration activities points out that what is needed is a more general methodology for the software engineering of distributed object systems. Such a methodology should enable all of the activities and technologies relevant to the development and integration of enterprise-scale systems to be defined, organized, and interrelated in a coherent, consistent, and effective way.

3. What is the Process of Middleware Selection?

Due to the variety of roles that middleware selection may play, and to the relative immaturity of software methodologies for distributed object systems, it is evident that there is not, and cannot be, one process for middleware selection. The role that middleware selection plays, and the context in which that role is determined, set important parameters that govern the particular selection process used in (or appropriate for) a particular enterprise or project. These parameters include, for example,

- The parties involved, their roles, the amount of communication needed between them, and the level of discourse required
- The applicability and use of standards
- The breadth of middleware technologies considered
- The criteria and means for technology evaluation

The level and scope of development or integration, and whether it is within a group (team, department, or enterprise) or crosses group boundaries, also has an effect on the complexity of the software process in general and of middleware selection in particular. A particular defined selection process may execute variously, and variously well, depending on the level and scope of the project to which it is applied.

Despite the potential variety of middleware selection processes, a number of common characteristics or recommendations regarding middleware selection processes were identified in this session.

Selection may be made based on a number of approaches to comparing and evaluating middleware technologies. One of the papers in this session describes an analytical approach, while the other is based on an experimental approach. It may also be possible to use simulation for comparison, but this requires the use of simulation technology, which may not be available yet in many cases.

The frameworks, taxonomies, and criteria that are used for comparing and evaluating middleware must reflect enterprise-specific and project-relevant conditions. General frameworks, taxonomies, and criteria may be useful to some extent, but they must be tailored for a particular organizational situation.

Any particular development or integration process is also likely to involve a number of kinds of stakeholders with diverse background and interests. Consequently, an effective middleware selection process will promote communication and understanding among these stakeholders.

Standards can be useful for enhancing communication among stakeholders as well as facilitating technical integration. For example, XML provides a means to define standard document types that facilitate information exchange between heterogeneous systems; the use of XML requires agreement among cooperating organizations on the structure of these documents, and a number of domain-specific XML standards are under development. Where official standards are not published, certain widely used middleware technologies, for better or worse, may implicitly define *de facto* standards. Even where standards are used, though, nonstandard extensions offered by particular technology providers may determine the selection of particular middleware products. These extensions may provide useful features or functionality, but at the cost of interoperability or flexibility with respect to other technologies or technology providers. Whether this tradeoff is worthwhile depends on the particular enterprise, project, and technology involved.

To afford an appropriate degree of flexibility in middleware selection, the selection process should allow for the adoption of multiple alternative (or complementary) technologies in the solution to any particular problem. Additionally, the selection process should allow for the refinement of a general development or integration problem into more specific problems, in different areas or on different levels, to which different solution constraints and opportunities may apply.

One of the most critical challenges in middleware selection is accommodating the various time frames over which middleware selection, implementation, and use are to occur. One concern is controlling costs through the stages of the middleware life cycle and assuring the amortization of those costs across the life-cycle period. Another concern is accommodating change during the life of the system, as unexpected changes may incur unexpected costs or prematurely cut short the effective life of the system.

Especially at the enterprise scale, the long-term view is important. Two general approaches to building long-lived systems were advocated in this session:

- Anticipate changes and plan for them (build in points or mechanisms of change)
- Don't try to anticipate changes but use simple, flexible constructs (that can accommodate change as it occurs)

Enterprise system builders find themselves in a somewhat paradoxical situation. The cost of enterprise systems development and integration, and the time required to complete such projects (which may be years), means that careful planning is necessary. However, the evolution of both enterprise needs and software technologies cannot be totally controlled. It is inevitable that plans will go astray. Therefore, the development and integration of enterprise systems must be based on a long-term strategy, but it must also be recognized that the object of that strategy will be a moving target. Indeed, middleware selection itself may be an ongoing activity that outlives individual application and integration projects in a long-term enterprise.

Acknowledgments

This paper is based in part on the comments of the following workshop attendees, who contributed to the discussion during this session: Premkumar Devanbu, Wolfgang Emmerich, Alfonso Fuggetta, Michael Goedicke, Cecilia Mascolo, Walter Schwarz, Antonio Rito Silva, Jörn Guy Süß, Stanley Sutton, and Stefan Tai. Thanks to Stefan Tai, Jörn Guy Süß, and Michael Goedicke for their review of a draft of this manuscript.