Streamlining the Acquisition Process for Large-Scale COTS Middleware Components

Ian Gorton[1] and Anna Liu

Software Architectures and Component Technologies Group,
CSIRO Mathematical and Information Sciences,
Locked Bag 17, North Ryde, Sydney 2113, Australia
{Ian.Gorton;Anna.Liu}@cmis.csiro.au

Abstract: CSIRO's i-MATE process is an established approach to help IT organizations in the acquisition of large-scale COTS middleware components. It aims to minimize technical risk by matching detailed application and infrastructure requirements to the capabilities of COTS middleware products. This paper describes a case study on the use of i-MATE in a project that required the selection of appropriate components in a compressed timeline and from a broad range of candidate COTS technologies. The steps and tools in i-MATE are briefly explained, along with the characteristics of COTS middleware components that make them a unique challenge in terms of acquisition and adoption. The case study project is then outlined, and the key business and technical issues explained. Finally, we describe and evaluate the enhancements made to the i-MATE approach to successfully respond to the challenges encountered.

1. Introduction

Middleware is a term that refers to a broad class of software infrastructure technologies for building distributed systems. Over the past decade, the adoption of COTS middleware products across the software industry has gathered huge momentum. It is slowly becoming a rarity to find major organizations whose business does not rely on some form of COTS middleware components. For example, the use of middleware in Fortune 500 companies is pervasive.

The two key reasons for this growth are Internet usage, and the need to integrate heterogeneous legacy systems to streamline business processes [1]. As organizations do more and more business on-line, they need to provide scalable, high-performance infrastructures to handle business transactions and provide access to core business back-end systems. The latter requires controlled and managed integration between disparate systems that were never designed to interoperate.

COTS middleware components help solve these problems [2]. They provide core software infrastructures that make it relatively straightforward to build distributed

[1] Adjunct Professor, Basser Department of Computer Science, University of Sydney, Sydney, Australia

applications that are high-performance and scalable. They provide sets of services that support, for example off-the-shelf distributed transaction processing, security features, and directory and naming services. They also provide specialized components for integration with a massive variety of legacy systems, and the ability to design and deploy new business processes that integrate multiple distributed applications. There are numerous COTS middleware products available. These are based on some standard infrastructures such as CORBA or J2EE, or proprietary technologies such as COM+ or MQSeries.

This paper firstly describes the challenges that organizations face when they wish to acquire new COTS middleware technologies. A brief classification of middleware products is then given, and the steps in CSIRO's i-MATE approach are explained. i-MATE has evolved over a 2-year period and has been successfully applied in five projects with major industrial clients including the Australian Stock Exchange, the Australian Tax Office and the Defense Housing Association. These organizations all required help in the acquisition of COTS middleware technology to integrate their core business applications. Finally, we describe a case study that forced us to streamline the i-MATE process in order to meet tight deadlines and rapidly evaluate a broad range of candidate middleware products.

2. The Challenges of COTS Middleware Acquisition

COTS middleware components form the infrastructure, or *plumbing* of integrated enterprise information systems. They provide a distributed environment for deploying application level components that carry out business-specific processing.

This distinction between infrastructure level components and application level components is crucial. Application level components rely on the infrastructure components to manage their lifecycle and execution, and to provide them off-the-shelf services such as transactions and security. Hence, an application level component cannot execute outside of a suitable middleware environment. The two are extremely tightly coupled.

An important implication of this is that the behavior of application components is completely dependent upon the behavior of the infrastructure components. The two cannot be divorced in any meaningful way – the entire application's behavior is the combination of the behavior of the application and infrastructure components. This scheme is depicted in Figure 1.

All this has profound implications upon component certification and engineering. No matter how high the quality of the application components, the COTS middleware infrastructure becomes the most crucial component in most systems. If the middleware is naively architected or implemented, has subtle errors in some services, or is simply inefficient and lacking in features, then the application components inevitably pay the price.

Interestingly, open standard COTS middleware infrastructures such as CORBA and J2EE actually exacerbate this problem. With CORBA and J2EE technologies, many vendors sell their own versions of the middleware infrastructure. These are all implemented differently, and hence behave and perform differently [3]. This means,

for example, a J2EE component's performance is dependent upon the actual J2EE product that it runs on. The same component may perform very differently indeed on two different J2EE implementations [4], depending on the quality and features of the product. This of course is not the case with single-source component technologies such as COM+ or MQSeries.

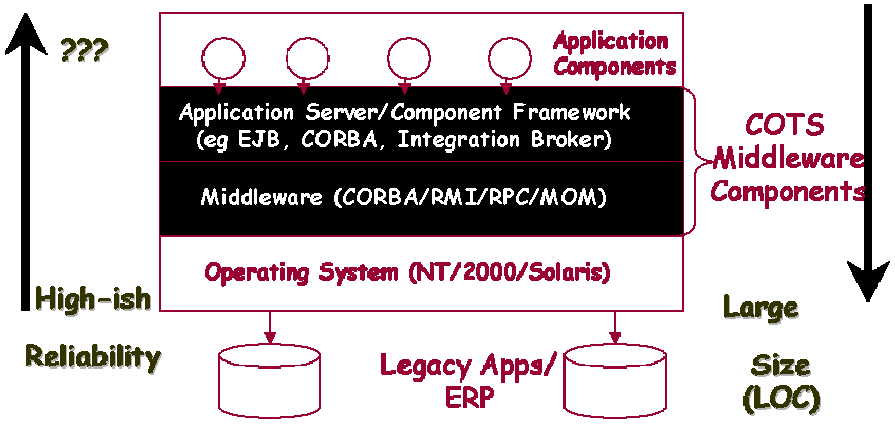

Fig. 1. Anatomy of a COTS Middleware Application

When acquiring a new COTS middleware product, most organizations do not have any pre-built application level components to run. The application components will be built when the middleware has been selected. This makes the selection of COTS middleware components absolutely crucial. If the middleware infrastructure is not 'up to the job', then the application is doomed to failure before it has even been designed.

Unfortunately, COTS middleware products are large and complex components. They will typically have well over 1000 API calls, and comprise a collection of integrated services, that are of varying levels of importance to different applications. There is rarely, if ever, a 'one size fits all' scenario in the acquisition of these technologies.

Apart from size and complexity, other issues make the acquisition of COTS middleware a challenge:

- There are numerous competing products in the middleware space. Narrowing the competition down to a manageable shortlist is a challenge.
- Vendor specifications of their COTS products are vague, and all claim superior features and functionality. This makes it difficult to do any serious comparison of the technologies at more than a very superficial level.
- Product pricing varies significantly, and rarely are acquisitions less than $500K. They are often considerably more.
- The whole middleware area is fast moving and evolving. It is often difficult to discern tested from promised functionality.

2. The i-MATE Process

CSIRO's Middleware Technology Evaluation (MTE) project commenced in 1999. Its aim is to help organizations understand, acquire, and successfully deploy large-scale enterprise information systems based on COTS middleware technologies. To this end, two main themes exist, namely:

1. **Rigorous Technology Evaluation:** This work has devised a rigorous, repeatable approach for performing qualitative and quantitative evaluations of COTS middleware components. Experiments and benchmarks are carried out in our lab, and the results are published in independent technology evaluation reports.[2] This work has so far focused on application servers and COTS message-oriented middleware (MOM) components (see Figure 2).
2. **Middleware Technology Acquisition:** This theme focuses on working with businesses to help them evaluate and select appropriate COTS middleware components for their specific applications.

(2) has evolved over several projects [5] with major clients in to the i-MATE (**M**iddleware **A**rchitecture and **T**echnology **E**valuation) approach. In essence, i-MATE is a specialized process for evaluating COTS middleware components, and is equivalent to the *Evaluation* set of activities defined in [6]. With i-MATE, CSIRO provides *just-in-time* technical expertise [7] and works with the business architects to match their technical and business requirements to a suitable infrastructure product.

The basic steps in i-MATE are as follows:

1. **Elaborate customer requirements:** The first step produces a document that captures the customer's requirements. It details the business and technical requirements that are specific to this application environment. Each requirement represents a single item that can be evaluated against a COTS middleware technology.
2. **Augment with generic requirements:** CSIRO has developed a knowledge base of currently around 150 generic, broadly applicable requirements for COTS middleware technologies. These are organized in categories such as message management, transaction services, and security features, and capture the key differentiating areas in middleware products. They focus across a range of functional and quality requirements, and collectively embody considerable accumulated experience from previous projects. These are merged with the application-specific requirements.
3. **Rank overall requirements:** Working with the key stakeholders, the overall set of requirements is ranked. At a coarse level, each requirement is deemed as *mandatory*, *desirable* or *low priority*. Within each of these 3 categories, importance weightings are assigned to give fine grain control over requirement rankings, in a fashion similar to [8].
4. **Identify candidate products:** This step identifies the 3-5 COTS middleware products that are most likely to be applicable to the overall application

[2] See http://www.cmis.csiro.au/adsat/mte.htm

requirements. In some cases, the customer has already identified a shortlist. In others, we use our experience to work with the customer to identify the most likely candidates.

5. **Product Evaluations:** In workshops with the key customer stakeholders and product representatives, we evaluate each of the candidate products against the overall requirements. Scores are allocated against each requirement point for each product. This involves in-depth technical discussions, and stepping through relevant application scenarios to understand precisely how the COTS middleware products actually behave. In terms of effort, this stage typically consumes the majority of the project's effort.
6. **Product Selection:** The application requirements and product scores are captured through the process in a requirement management tool based on a spreadsheet. Once complete, the tool automatically compiles summary scores based on individual requirement point scores and requirement category weightings. Summary charts are also automatically created to support reporting purposes.

At the end of this process, it is usually clear which candidate technologies are capable of satisfying the application requirements. In some cases, two or more products may be close in their overall ranking. In such circumstances, the spreadsheet tool becomes extremely useful.

In the spreadsheet, it is a simple task to modify requirement rankings to see the effect on product selection. Typical strategies involve promoting some *desirable* requirements to *mandatory*, or relaxing some *mandatory* requirements that have proven difficult to satisfy due to some technology restrictions or trade-offs. The key customer stakeholders make these decisions in order to explore a number of *what-if* scenarios.

When a requirement ranking is changed in the spreadsheet, new product rankings are automatically charted. This makes it a trivial exercise to see the effects of varying individual requirement points on product selection. In this way, a leading product can quickly be identified.

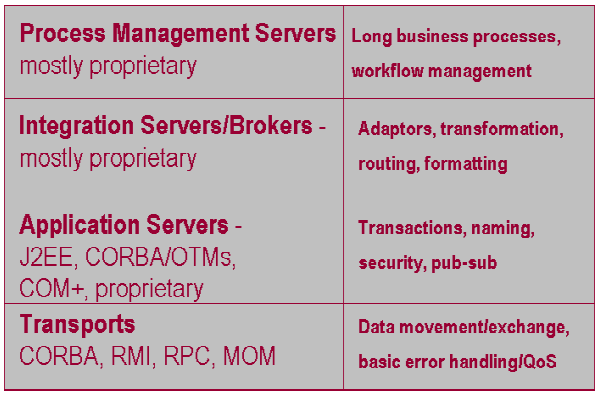

Figure 2 A Layered Taxonomy for COTS Middleware Technologies

i-MATE has been successfully applied in five major projects. The focus of all these projects has been the acquisition of COTS middleware components for application integration. Hence the technology focus has been the category of middleware known as integration brokers or message brokers. These basically provide collections of specialist integration components known as adapters or connectors, a messaging-based communications infrastructure, and development tools for describing business process and message transformations that span multiple legacy systems.

Figure 2 shows how integration brokers fit in to the overall spectrum of currently available middleware technologies. Integration brokers are typically layered upon lower level middleware technologies. As concrete examples, BEA System's WebLogic Process Integrator runs on top of their application server, WebLogic Server; IBM's MQSeries Integrator runs on top of MQSeries.

Consequently, a thorough understanding of integration brokers requires knowledge and experience with the underlying COTS middleware layers. In i-MATE, we are able to leverage the independent evaluations of these technologies that are published in our MTE reports. Hence the MTE reports contribute to the overall knowledge base used in i-MATE.

The unique attribute of the i-MATE approach is the inclusion of a comprehensive set of generic requirements applicable to COTS middleware technologies. This is valuable, reusable intellectual property, and saves customers from independently devising their own set. Based on these generic requirements, it is a relatively easy task to augment them with application-specific requirements. Also, once a COTS middleware product has been evaluated against the generic requirements, the evaluation scores can be reused in subsequent projects[3]. Again, we are able to reuse this knowledge, and only need to evaluate the product against the new application-specific needs.

This reuse of generic requirements and product evaluations makes it possible to apply i-MATE in relatively short project timeframes. A typical i-MATE assignment takes 3-4 weeks in elapsed time, and between 15-25 person days effort from the i-MATE team. Given the complexity of COTS middleware components and the scale of the problems they are used to solve, this is a small level of effort and cost. Importantly, it matches the compressed project schedules that are now common in Internet and e-business projects. Three-month product selection projects are rarely viable in these environments.

4. Case Study

A large Australian organization has over 30 core business systems. These revolve around an Enterprise Resource Planning (ERP) system for sales and production planning, a data warehouse and a Customer Relationship Management (CRM) application. As the need for new, integrated business processes has grown, so has the number of application-specific point-to-point interfaces between applications.

[3] Until major product changes occur.

At the last count, there were over 150 such interfaces, with approximately 100 new ones planned within the next year. Most of these new interfaces were being driven by the demands to open up core business systems to the Internet. This was creating major development and maintenance problems. Figure 3 gives a high-level schematic of the relationship between these systems

The organization engaged CSIRO to help them select COTS middleware components for application integration. The eventual aim is to eliminate all point-to-point interfaces and use an integration broker to manage all application integration. The integration broker will provide a single repository for the definition of message types and formats, message transformations, and business rules used in integration. The broker must also provide a high-performance, scalable and reliable infrastructure to handle large volumes of transactions for business-critical applications.

Apart from this longer-term strategic aim of eliminating point-to-point application interfaces, there were two more immediate needs for the COTS middleware technology. These are:

1. A key interface between the CRM and ERP was causing operational problems for a key business process. Due to the nature of the business, the majority of the transactions across this interface took place in a 4 hour period at the start of the week. Current loads were around 60,000 transactions in that period, and this number was growing rapidly. The middleware product was required to eliminate the problems with this interface, and provide real-time turnaround for transactions.
2. There was an increasing need for the organization to exchange information electronically with business partners. Therefore, the COTS middleware technology adopted must have features that support a wide variety of business-to-business (B2B) interface types.

In addition, due to timing and budgetary constraints, the acquisition stage of the project was scheduled for a 1-month duration. This would then be followed by a proof-of-concept stage during which the selected COTS middleware product would be tested and validated.

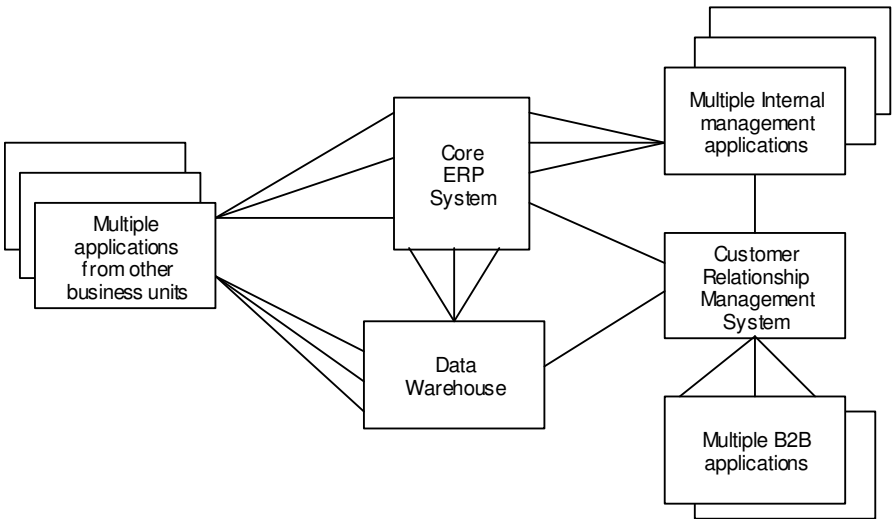

Figure 3 Enterprise Architecture

5. Streamlining i-MATE

At the start of the project, we began following the i-MATE process. The customer-specific requirements were quickly understood, reduced to specific requirement points and merged with the generic requirements that accompany the i-MATE process. In a series of workshops, the ranking process of the specific requirement points was commenced, and concurrently, the list of candidate technologies was drawn up.

It was at this stage where we began to encounter some problems. In past projects, i-MATE has evaluated between 3 and 5 candidate COTS middleware technologies. The product evaluations against the application requirements are time-consuming, taking at least 1 day for products that have already been evaluated in previous i-MATE projects, and typically 2-4 days for products that are new to the evaluation team.

However, for various non-technical reasons, the organization presented us with a list of 13 candidate products. This was clearly too many to fully evaluate in a one month project. As the project schedule was fixed, it was necessary to devise a strategy that allowed the number of candidate products to be reduced to a manageable size. In addition, strong justification was needed to eliminate a product, so that the organizations' upper management felt comfortable that the most appropriate technologies were being considered.

To this end, we modified step 4 of i-MATE as follows:

1. In conjunction with the customer organization's technical team, 5 key requirements were identified. These were a combination of application-

specific (e.g. CRM integration components) and generic middleware (e.g. scalability) requirements.

2. The 13 candidate products were quickly categorized as *strong*, *weak* or *unknown* in terms of addressing these 5 key requirements. This was based on the previous experience of the i-MATE team.
3. An information gathering exercise was commenced to qualify the *weak* and *unknown* candidate products. In three cases, it was possible to eliminate a product based on a simple non-conformance issue with regards to a platform that the organization needed support for. In 2 cases, products were eliminated because they did not have strong local organizations to support their product – this was a mandatory requirement. Finally, one product was eliminated because, in discussions with the vendor, it became clear that the product's real strengths were not in application integration.
4. Half-day workshops were organized with each of the seven remaining vendors. In these workshops, the majority of the time was spent probing on how each product could support the five key requirements. In addition, issues related to costs were broadly addressed.

The seven workshops took place during the same week. At the end of this week, we had a set of scores on how each candidate COTS product was able to address the key application requirements. Using a spreadsheet tool and weighting the five requirements in various ways, we worked through various scenarios with the customer to explore the various product strengths and weaknesses. It was clear that three of the products were considerably stronger than the others in terms of their features and underlying architectures. Consequently, these three were selected as the candidate COTS middleware products to fully evaluate in step 5 of i-MATE.

6. Evaluation

The modified *Identify Candidate Products* step of i-MATE worked extremely well. It enabled us to rapidly reduce the number of COTS middleware products to a more manageable set, and then perform relatively detailed evaluations of the remaining products against the key application requirements. This process was successful in:

1. Rapidly identifying the strongest candidates to be evaluated in more detail
2. Providing clear justification to the customer's technical and managements teams as to why certain COTS middleware products were less suitable
3. Helping identify the highest priority requirements with the client at an early stage in the process.
4. Not expanding the overall project schedule – the i-MATE process was still completed in the four-week schedule for the project.

In fact, we see step (3) above as a valuable contribution to the whole process. It helps focus the client's thinking at an early stage of the acquisition project, and brings the highest priority requirements to the fore, for primary consideration.

7. Conclusions

The i-MATE process has been developed to help with the acquisition of COTS middleware technologies. These are complex, highly technical and diverse collections of products that typically operate in business-critical environments. i-MATE's key contributions, which have formed the basis for several successful projects are:

- A pre-fabricated, reusable set of generic requirements, based upon the practical analysis in business environments of application and COTS middleware components characteristics
- A defined process for incorporating application-specific requirements
- A process and tool for weighting requirements and ranking specific COTS middleware products against the requirements
- Tool support for rapidly exploring requirement trade-offs and generating reports showing how the COTS products compare against the requirements.

This paper has described some modifications that were introduced in to the core process to enable i-MATE to cover a broader range of candidate COTS middleware products. These modifications made it possible for the i-MATE team to evaluate 13 candidate COTS middleware products, and still complete the project in a 4 week schedule. The success of these modifications demonstrates the solid core foundations of i-MATE, and its inherent flexibility and adaptability to changing business requirements during the acquisition process.

References

1. Ron Zahavi, David S. Linthicum, Enterprise Application Integration with CORBA Component and Web-Based Solutions, John Wiley and Sons, November 1999
2. George T. Heineman (Editor), William T. Councill, Component-Based Software Engineering, Addison-Wesley June 2001
3. I.Gorton, A.Liu, P. Tran, The Devil is in the Detail, A Comparison of CORBA Object Transaction Services, the 6th International Conference on Object-Oriented Information Systems, pages 211-221, 18-20 December 2000, London
4. S.Ran, P. Brebner, I.Gorton, The Rigorous Evaluation of Enterprise Java Bean Technology, in 15th International Conference on Information Networking (ICOIN-15), Beppu City, Japan, Feb 2001, IEEE
5. Liu A., Gathering Middleware Requirements, Proceedings of the Australian Requirements Engineering Workshop, 8-9 November, 2000
6. Lisa Brownsword, Tricia Oberndorf, Carol A. Sledge, Developing New Processes for COTS-Based Systems, *IEEE Software,* Vol. 17, No. 4, July/August 2000
7. Kurt Wallnau, Scott Hissam, Robert Seacord, Building Systems from Commercial Components, Addison-Wesley, July 2001
8. Patricia K. Lawlis et al, A Formal Process for Evaluating COTS Software Products, *Computer,* Vol. 34, No. 5, May 2001