

In this article we are going to look at traditional and new challenges that face developers and enterprise integrators who work on Enterprise Integration solutions. Then we will see how Enterprise Integration Patterns can help us to address those challenges.

Traditional Challenges of EAI

EAI solutions are traditionally known to come with technical challenges like:

- Diversity of platforms
- Various transport protocols
- “Closed” applications

Software solutions like Microsoft BizTalk Server address most of those challenges by using standard protocols (e.g. HTTP, FTP, Email, etc). Even in the case of close applications, you can have application endpoints like a file sharing location when you can drop and collect messages.

Although these technical issues are still need to be addressed, but since there are products out there that help us to address those issues, many of the enterprise integrators are not much concerned about the technical challenges today. But not all of the EAI challenges are technical issues. Let’s look at EAI from another perspective, which has something to do with “Semantics”.

Newly Identified Challenges

Over time, enterprises have developed multiple applications and have purchased packaged software (this includes ERP applications from vendors like SAP, PeopleSoft, JDEdwards and Oracle). Not all of these solutions have been designed as part of the Enterprise Architecture of the organisation, either because a well-defined Enterprise Architecture did not exist at the time or it was not possible to fit that specific application into the Enterprise Architecture.

This introduces more challenges to the EAI solutions:

- **Functionality Overlap**
In some cases you can do the same job in two applications. For example in order to see the status of an order, you may be able to use either Sales or Delivery Tracking application. The problem is none of them does the job completely. The customer is either transferred to another department, or will be told that “Our computers are very slow today”. In the meanwhile the operator is launching another package to get a complete order status. Even they may come back to you and ask your account number or name again because they didn’t write it down the first time!

In an EAI project, normally we are not going to change any of the existing applications so addressing the functionality overlap issue is a challenge.
- **Data Duplication**
For example information for a customer may be held in Accounting software as well as the CRM package, which can become inconsistent. For example last week I had a call from the bank in which I hold an account. They were trying to convince me to upgrade my account to “Gold”, and I told them I have got a “Gold” account already. They were insisting that their system doesn’t show it.

- **Different Semantics**

Applications may have different interpretation of the same concept. For example in one application we may use the term “Property” for referring to a building, whereas in another application we may consider the same term for referring to a car as well. So when it comes to integrating applications, we may encounter inconsistencies in the semantics.

Since these challenges are related to semantics, we need to tackle them by using appropriate patterns, and not products. This introduces us to the “Enterprise Integration Patterns”. These patterns provide us with the best practices to solve common Enterprise Integration problems.

Now let’s have a quick look at some of the patterns that can be used to solve the problems mentioned above.

Problem: Functionality Overlap

Solution: Shared Business Function, Service-Oriented Architecture

“Shared Business Function” pattern suggests us to define a clear boundary for the shared functionality, implement it and then reuse it in multiple applications. Most of the time this shared functionality will be part of one of the existing systems, like the Delivery Tracking functionality in an Order Processing system.

Shared Business Functions can also be considered as “Services”. So we can use a Service-Oriented Architecture (SOA) to publish the Shared Business Function so that it can be used from other applications.

Problem: Data Duplication

Solution: Data Replication, Information Portal

To address the “Data Duplication” challenge, we can either use Data Replication or Information Portal pattern. “Data Replication” pattern talks about replicating data from one application to the other, so that they are in a synchronised and consistent state. When it comes to data synchronisation, there are various options available based on consistency and transactional requirements.

“Information Portal” suggests us to aggregate information from multiple sources (different applications) and then displays the result back to the user.

Problem: Different Semantics

Solution: Canonical Data Model

As mentioned earlier, different applications may have different interpretations for a single term. This makes the process of inter-connecting applications very hard because every application needs to have a data adapter to translate the terms when it wants to talk to other applications. So the total number of adapters needed for an enterprise grows exponentially when adding new applications to the Enterprise Architecture. To address this challenge, we can use the “Canonical Data Model” pattern, which talks about creating a canonical, enterprise-wide data model. This model doesn’t have to be implemented by all of the applications. The purpose of creating such a model is to have a conceptual view of the data within an enterprise, and then have a mapping between the canonical data model and the data model used within any specific application. This helps us to define the data adapters. Also we don’t need to create a separate data adapter for inter-connecting every two single application. What we need to do is to create an adapter to transform the canonical data model to the data model used by any other application, and vice versa. Using this pattern will reduce the number of needed adapters and we will have a linear relationship between the number of needed adapters and the number applications in the Enterprise Architecture.

Conclusion

In this article, we have seen some of the new challenges of Enterprise Application Integration solutions, and have shown how Enterprise Integration Patterns can help us to address those challenges. For more information, you can refer to the references to find out more about the Enterprise Integration challenges and patterns.

References

Enterprise Integration Patterns by Gregor Hohpe and Bobby Woolf, Addison Wesley, 2003
Patterns of Enterprise Application Architecture by Martin Fowler, Addison Wesley, 2002
<http://www.eaipatterns.com/>
<http://msdn.microsoft.com/practices/type/Patterns/Enterprise/>