Managing Data Integrity During Legacy Modernization
Table of Contents

1. Introduction 1

2. Need for Modernization 1

3. Modernization Considerations 1

4. Migration to the Modernized Platform 2
   4.1 Approaches for Migration 2
   4.2 Comparison of Phased Approaches 3
   4.3 Migration Approach Suitability Model 4
   4.4 iGATE Case Study 4

5. Conclusion 6
1. Introduction

Business needs are never static. Though applications are developed as per business requirements, they do not remain unchanged for a long time. Usually, applications need to be modified / updated to meet changing business needs. Initially, small modifications to the applications are sufficient to align them with the changed business environment. Over a period of time, limitations of the platform and the technology used for the applications result in creating a gap between what the business expects and what the applications are able to deliver. This gap may be created because the platform may have become obsolete and may not be able to provide adequate features. Additionally, as the applications and the platform grow old, skills required to keep the applications up-to-date may not be easily available.

When the gap between what the business expects and what is provided by the applications becomes large, the applications need to be “Modernized”, where modernization entails conversion, porting or re-writing of existing systems to a modern (typically Open Platform based) programming language, database, or hardware platform.

2. Need for Modernization

Some of the drivers for businesses choosing to modernize their IT infrastructure are given in Table 1.

<table>
<thead>
<tr>
<th>Business Need</th>
<th>Technology Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduced cost</td>
<td>Legacy system obsolescence (Mainframe, Server-Based Applications, Client-Server Applications, Early-Generation Web Applications)</td>
</tr>
<tr>
<td>Improved effectiveness by using business aligned IT applications</td>
<td>Leveraging IT advancements</td>
</tr>
<tr>
<td>Improved efficiency by increasing agility of business</td>
<td>Application standardization</td>
</tr>
<tr>
<td>Focus on commodity skills for applications</td>
<td></td>
</tr>
<tr>
<td>Better management of data using Master Data Management (MDM) concepts</td>
<td></td>
</tr>
</tbody>
</table>

3. Modernization Considerations

Any modernization initiative has to focus on two crucial elements that make up an IT system – applications and associated data. Each of the elements needs to be analyzed in detail, before arriving at a suitable modernization solution. Some of the typical modernization considerations are given in Table 2.
4. Migration to the Modernized Platform

When a modernized system is set to replace an existing system, care has to be taken to ensure that the most critical business asset – its data – is managed properly. Additionally, as legacy modernization typically involves critical applications, a large code base and huge amounts of data, the modernization effort usually spans multiple months. During this time frame, business does not remain static and data gets updated continuously. Hence, during modernization, the integrity and consistency of data needs to be maintained, as incorrect or inconsistent data will result in loss of business.

4.1 Approaches for Migration

During modernization, not only do we have to migrate applications, we also have to migrate the data to the Open Platform, using a well-defined approach. Based on the features of the approaches, they can be broadly classified (Figure 2) into two, namely:

i. Big-Bang Approach: In this approach, applications and data are migrated to the Open Platform in a single step, due to which, data integrity and consistency issues do not occur during migration.

ii. Phased Approach: In this approach, applications are migrated to the Open Platform in well-defined logical groups. Due to group-wise migration, till all applications and data have been migrated (i.e. while migration is in progress/during migration), some application groups will be on the Legacy platform while other application groups will be on the Open Platform. In such situations, it is critical to ensure there is no impact on the data integrity and consistency (while migration is in progress).

<table>
<thead>
<tr>
<th>Application Considerations</th>
<th>Data Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing business functiona</td>
<td>Need for data model</td>
</tr>
<tr>
<td>lity</td>
<td>changes</td>
</tr>
<tr>
<td>- Retained as-is</td>
<td>Mapping of source</td>
</tr>
<tr>
<td>- Enhanced</td>
<td>and target database</td>
</tr>
<tr>
<td>Mapping of source and</td>
<td>technology</td>
</tr>
<tr>
<td>target technology stacks</td>
<td>Handling live,</td>
</tr>
<tr>
<td></td>
<td>archive and</td>
</tr>
<tr>
<td>Managing application</td>
<td>historical data</td>
</tr>
<tr>
<td>security</td>
<td></td>
</tr>
<tr>
<td>Integration of 3rd party</td>
<td>Implementation of</td>
</tr>
<tr>
<td>tools</td>
<td>Master Data Management (MDM)</td>
</tr>
<tr>
<td>Training and transition</td>
<td>Implementation of</td>
</tr>
<tr>
<td></td>
<td>Information Lifecycle Management (ILM)</td>
</tr>
</tbody>
</table>

Table 2: Modernization Considerations

![Figure 2: Migration Approaches](image-url)
4.2 Comparison of Phased Approaches

Depending on the way the application and data migration is carried out, the Phased migration approach can be sub-divided into multiple approaches as given in Table 3.

<table>
<thead>
<tr>
<th>Phased Approach</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| **Database on Legacy Platform** (Single database is maintained on the Legacy platform) | ● Single database during migration  
● Data integrity ensured by usage of one database | Migrated applications need to be changed:  
● As part of Phase 1 - to access Legacy database from Open Platform  
● After Phase 2 - to access Open Platform database |
| **Database On Open Platform** (Single database is maintained on the Open Platform) | ● Single database during migration  
● Data integrity ensured by usage of one database | Before application migration can begin, Legacy applications have to be modified to access Open Platform database |
| **Legacy Database Primary, Open Platform Database Secondary**  
(Two databases - one on the Legacy platform and one on the Open platform - are maintained during migration. Updates are allowed only on the Legacy database) | ● Legacy database is primary data source  
● Data integrity ensured as Legacy database is only database where updates happen directly  
● Non-invasive approach | Open Platform applications will not be able to update the database if network connectivity is not available |
| **Both Legacy Database & Open Platform Database Primary with 2-Way Synchronization**  
(Two databases - one on the Legacy platform and one on the Open platform - are maintained during migration. Both databases can be updated. Data integrity and consistency maintained using synchronization) | ● Open Platform applications read and update Open Platform database  
● Data integrity ensured by synchronizing changes to both databases  
● Legacy applications read and update Legacy database | Reconciliation between Legacy database and Open Platform database needed to recover from loss of network connectivity |
| **Both Legacy Database & Open Platform Database Primary with 2-Phase Commit**  
(Two databases - one on the Legacy platform and one on the Open platform - are maintained during migration. Both databases can be updated. Data integrity and consistency maintained using Two-Phase Commit) | ● Data integrity ensured as both databases are updated simultaneously  
● Read operations performed on local database | Legacy applications have to be modified to support Two-Phase Commit  
● After migration, Open Platform applications modified to remove Two-Phase Commit |
4.3 Migration Approach Suitability Model™

To help choose a suitable approach, we at iGATE, have created a model called ‘Migration Approach Suitability Model™’. The model is based on three main parameters:

i. Migration Drivers and Attributes – captures important business goals,

ii. Suitability Preference – indicates suitability of the approach for the migration driver (and its attribute),

iii. Customer Priority – assigned by the customer.

Using these parameters, the model assigns a value to each of the migration approaches. The approach with the maximum assigned value is the recommended approach. An illustrative example of the model, with suitable values is given in Table 4 and the corresponding recommendation is given in Figure 3.

Table 4: Method for Choosing Suitable Migration Approach

<table>
<thead>
<tr>
<th>Driver</th>
<th>Attribute</th>
<th>Customer Priority</th>
<th>Approaches</th>
<th>Database on Legacy</th>
<th>Legacy DB Primary, Open Platform DB Secondary</th>
<th>Both Legacy and Open Platform DB Primary (Two-Way Sync)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Transactions</td>
<td>High</td>
<td>Critical</td>
<td>Suitable</td>
<td>Suitable</td>
<td>Not Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Moderate</td>
<td>Low</td>
<td>Suitable</td>
<td>Most Suitable</td>
<td>Not Applicable</td>
<td></td>
</tr>
<tr>
<td>Number of Transactions</td>
<td>Online</td>
<td>Moderate</td>
<td>Suitable</td>
<td>Most Suitable</td>
<td>Least Applicable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Batch</td>
<td>Critical</td>
<td>Most Suitable</td>
<td>Suitable</td>
<td>Least Applicable</td>
<td></td>
</tr>
</tbody>
</table>

Typically, the model is used in conjunction with the modernization CTQs (Critical To Quality), application details and details about data. The model facilitates vetted decision-making while selecting the best-fit option for migration.

4.4 iGATE Case Study

One of our customers wanted to migrate their Mainframe-based application to the Open Platform. The application considered, was deployed across multiple geographical regions. In other words, there were multiple instances of the same application, with one instance per geography. The application code base was in COBOL, while the database was CA-IDMS (Hierarchical database).

The modernization implementation solution suggested was Re-hosting, with COBOL being retained as-is on the Open Platform. As CA-IDMS is not supported on the open platform, it was suggested that the database (CA-IDMS) be migrated to Oracle.
After conducting an assessment of the existing application and understanding the customer requirements, the collected parameters were run through the model. The migration approach suggested was Phased migration and in particular, Legacy Database Primary, Open Platform Database Secondary.

**Key Features of the Solution**

Some features of the recommended solution are given below:

- **Approach**
  - Phased migration
  - Legacy Database Primary, Open Platform Database Secondary

- **Features**
  - Data integrity and consistency is maintained by ensuring that only one database is updated by all applications (while migration is in progress)
  - Migrated applications update a View on Oracle, instead of a Table (till all business regions are migrated to the Open Platform)
  - Updates by migrated applications, to Oracle database are applied to Mainframe IDMS database, via tcVISION, instead of the Oracle database. This is transparent to the migrated applications and non-invasive
  - Updates to the Mainframe IDMS database are replicated to Oracle using real-time replication feature of tcVISION
  - Updates by legacy applications, to Mainframe IDMS database, are replicated to Oracle using real-time replication feature of tcVISION
  - After all regions have been migrated, the Oracle View is removed. Real-time replication (using tcVISION) from Mainframe IDMS to Oracle is also discontinued and the IDMS database can be retired.

- **Software**
  - tcVISION, from Treehouse Software, Inc. used for data synchronization between CA-IDMS (on Mainframe) and Oracle (on Open Platform)

A schematic of the solution is given in Figure 4.
Conclusion

Legacy modernization involves migration of applications and data to a new platform (typically Open Platform). Usually, the size of data that needs to be migrated to the Open Platform is very large, due to which, modernization initiatives span multiple years. While business users and CIOs have to strive for minimum possible downtime when moving to the Open Platform, ensuring that data integrity and consistency are not compromised (while migration is in progress), is of prime importance.

To maintain data integrity during migration, access to the data (typically residing in the database) needs to be controlled. Data integrity can also be guaranteed, either by using a single database, or, by using a failure-resilient synchronization mechanism in case of multiple databases.

References

2. Taxonomy of Legacy Modernization Approaches; Erradi et al; 2006
3. tcVISION, Treehouse Software, Inc.; http://www.treehouse.com
About the Author

Bipin Patwardhan is a Technical Architect with more than 15 years of experience in the IT industry. A member of the Research & Innovation (R&I) – Mainframe Modernization (MM) CoE at iGATE, he is responsible for creating modernization solutions, with a focus on creating re-implementing solutions with Java as the target platform.

Mainframe Application Modernization CoE

With its vast experience in providing IT services to global clients across industry domains, iGATE’s Mainframe Application Modernization CoE has a deep understanding of mainframe environments and the mission-critical role of hosted legacy applications. iGATE’s customized approach to application modernization preserves the integrity of business operations during and after the modernization process. The result is improved flexibility and the ability to integrate legacy applications with emerging business needs of the enterprise.

The iGATE solution uses advanced tools that automate the ‘documentation’ and most of the ‘mainframe application migration’ process to open platforms, namely Linux, Unix, or Windows, using technologies such as Java/JEE, Microsoft .NET, Micro Focus COBOL, and ISCOBOL. Organizations migrating mainframe applications to open systems not only get one-time benefits, but also recurring benefits which results in optimal total cost of ownership of IT operations.

Over the years, iGATE has developed various methodologies, tools, and services to accelerate time-to-value for our clients. iGATE’s proven accelerators enable predictable outcome, on-time delivery within the budget, and improved return on investment (ROI).

To know more, visit our Website.

About iGATE

‘iGATE’ is the common brand identity of two organizations — iGATE Corporation and iGATE Computer Systems Limited (iGATE). With iGATE having acquired a majority stake in iGATE, the two companies, under the common brand iGATE, provide full-spectrum consulting, technology and business process outsourcing, and product engineering services on a Business Outcomes-based model.

Armed with over three decades of IT Services experience and powered by the iTOPS (Integrated Technology and Operations) platform, iGATE’s multi-location global organization with a talent pool of 26000+ people, consistently delivers effective solutions to over 360 Fortune 1000 clients spanning across verticals like: banking & financial services; insurance & healthcare; life sciences; manufacturing, retail, distribution & logistics; media, entertainment leisure & travel; communication, energy & utilities; public sector; and independent software vendors.

iGATE Corporation is listed on NASDAQ (IGTE), and iGATE Computer Systems Limited on BSE (532517), NSE (iGATE) and NYSE (PTI).