An Overview of Legacy Information System Migration

Jesus Bisbal, Deirdre Lawless,
Bing Wu, Jane Grimson, Vincent Wade
Computer Science Department,
Trinity College, Dublin, Ireland.
Email: {*name.surname*}@cs.tcd.ie

Ray Richardson[1], D O'Sullivan[2],
Broadcom Éireann Research,
Dublin, Ireland.
Email: [1] : rr@broadcom.ie
[2] : dosullivan@ broadcom.ie;

Abstract

Legacy information systems typically form the backbone of the information flow within an organisation and are the main vehicle for consolidating information about the business. These systems also pose considerable problems - brittleness, inflexibility, isolation, non-extensibility, lack of openness etc -, the so called legacy system problem which opens up a new research topic, legacy system migration. This paper provides a brief review of the main issues involved in legacy information systems migration.

1. Introduction

Legacy information systems[1] typically form the backbone of the information flow within an organisation and are the main vehicle for consolidating information about the business. These mission critical legacy ISs are currently posing numerous and important problems to their host organisations. The most serious of these are:

* these systems usually run on obsolete hardware which is slow and expensive to maintain;
* maintenance of software is generally expensive; tracing faults is costly and time consuming due to the lack of documentation and a general lack of understanding of the internal workings of the system;
* integration efforts are greatly hampered by the absence of clean interfaces;
* legacy systems are very difficult, if not impossible, to expand.

In response to these problems, many organisations now wish to move their legacy systems to new environments which allow information systems to be easily maintained and adapted to new business requirements but retain functionality of existing information systems without having to completely redevelop them. This is the essence of *Legacy Information System Migration*.

2. Migration methodologies

Given the scale, complexity and risk of failure of legacy system migration projects a well-defined, detailed methodology that can be easily implemented is essential to their success. Although legacy information system migration is a major research issue, there are few comprehensive migration methodologies available.

A naive approach to migrating a legacy system would be to redevelop the legacy system from scratch, using a modern architecture, tools and databases, running on a new hardware platform. This approach is known as *Big Bang* approach, also referred to as *Cold Turkey* [1]. In reality, the risk of failure is usually too great for this approach to be seriously contemplated.

The Forward Migration Method [1], referred to here as the *Database First* approach, involves the initial migration of legacy data to a modern, probably relational, Database Management System and then incrementally migrating the legacy applications and interfaces. The legacy and target systems operate in parallel throughout the migration. A Forward Gateway[2] enables legacy applications to access the database environment in the target side of the migration process.

Using the Reverse Migration Method [1], referred to here as the *Database Last* approach, legacy applications are gradually migrated to the target platform while the legacy database remains on the original platform. The legacy database migration is the final step of the migration

---

[1]A legacy information system can be defined as "any information system that significantly resists modification and evolution", [2].

[2] A Gateway can be defined as "a software module introduced between operation software components to mediate between them" [Brod95].

process. A Reverse Gateway enables target applications to access the legacy data management environment.

Using the most mature currently available approach, the *Composite Database* approach [1], legacy applications are gradually rebuilt on the target platform using modern tools and technology. The legacy and target systems form a composite information system during migration, Figure

A (forward) gateway allows the legacy system access data in the target database. A (reverse) gateway allows the target system access data in the legacy database. Data may be duplicated across both the legacy and target databases. To maintain data integrity, a Co-ordinator is used which intercepts all update requests, from legacy or target applications, and processes them to identify whether they refer to data replicated in both databases.
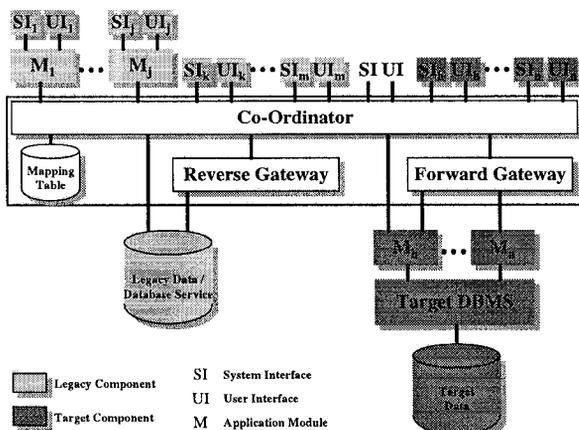
Figure 1 Composite Database Migration Approach

A significant drawback of approaches which use gateways is their inherent complexity, since they require the interoperation of (potentially) heterogeneous environments. Another approach, the Butterfly methodology, presented in [2], questions the need for the legacy and target systems to interoperate during migration.

3. Tools to support legacy system migration

Legacy information system migration is a long process, typically lasting five to ten years [1], and is also a highly risky undertaking. Migration tools can considerably reduce the duration of a migration project, helping the migration engineer in tedious, time-consuming and error-prone tasks. Figure 2 presents a classification of tools that could be needed throughout a migration project.

The majority of tools available are those needed in any software engineering process (e.g. target system development, and testing). Specific tools for legacy migration are still to come (e.g. justification, target system cut-over, and legacy system understanding) .

4. Legacy migration research issues

Legacy system migration could quite legitimately encompass a large number of different areas of software engineering: program understanding, database understanding, system development and testing, etc. Figure 3 shows a classification of the most important issues that must be addressed during a migration project.
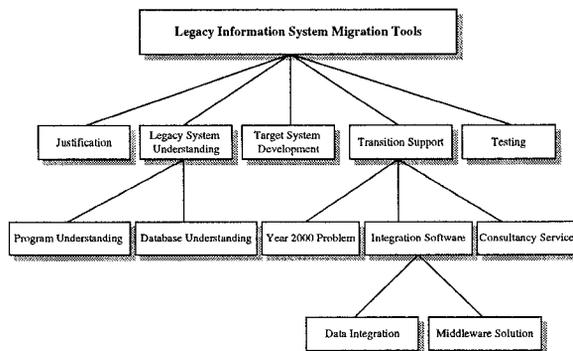
Figure 2 Classification of Legacy Migration Tools

Much research is still needed in the area of legacy migration. Issues like target system testing and cut-over, legacy data cleaning and mapping for example, require further investigation.
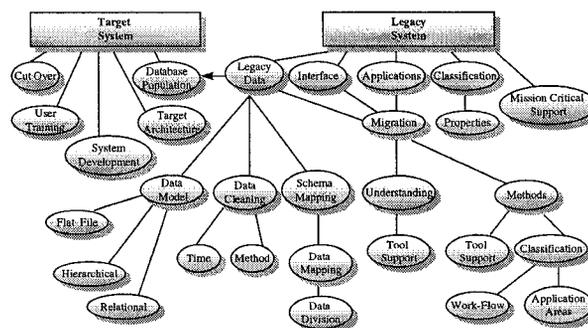
Figure 3 Legacy System Migration Issues

5. References

[1] M. Brodie, and M. Stonebraker, "Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach", Morgan Kaufmann Publishers, 1995

[2] B. Wu, D. Lawless, J. Bisbal, J. Grimson, V. Wade, D. O'Sullivan, and R. Richardson, "Legacy systems migration - a method and its tool-kit framework", in Proc. of the Joint 1997 Asia Pacific Software Engineering Conference, and International Computer Science Conference (APSEC'97/ ICSC'97), December 1997, Hong Kong, China.