



---

# Legacy Transformation White Paper

Business Rule Extraction and Transformation of COBOL  
Applications

---

Dr Cyrus Montakab

[www.softwaremining.com](http://www.softwaremining.com)



# Table of Contents

INTRODUCTION.....	3
BACKGROUND .....	4
THE HEART OF THE PROBLEM .....	4
WHY SOFTWAREMINING?.....	5
WHY MIGRATE? THE RETURN ON INVESTMENT .....	6
THE DIFFICULTY WITH ROI .....	6
REDUCE COSTS.....	7
FLEXIBLE AND FUTURE-PROOF – BUSINESS AGILITY.....	7
IMPROVE INFORMATION ASSETS .....	8
E-BUSINESS ENABLER .....	8
CASE STUDIES.....	9
CASE SCENARIO 1 - REJUVENATION OF A LEGACY APPLICATION .....	9
<i>The Benefits of Legacy Application Rejuvenation To These Clients:</i> .....	9
CASE SCENARIO II - DOCUMENTATION OF EXISTING SOFTWARE .....	10
CASE SCENARIO III - INTEGRATION ISSUES .....	10

# Introduction

***“\$2 trillion worth of assets that are dated, inflexible and often at a high degree of risk for failure” EDS Legacy Modernization Report - Feb 2008***

Many SME companies, together with almost all Fortune 500 organisations, rely on legacy applications for the day-to-day running of their business. Most such organizations appreciate the importance of the web, trading exchanges, integrated supply chains, etc. and believe that certain parts of their existing systems have to be enhanced or reworked to benefit from these recent technology innovations. This work involves re-documentation, extraction of business rules and often an expensive rewrite of the legacy systems.

There are many business reasons why the value of investment in legacy systems needs to be unlocked. B2B is probably going to be the biggest, forcing a lot of companies to re-evaluate their systems. One of the biggest reasons for this new market of software transformation to exist is that organizations are just getting tired of applying continuous fixes to their legacy systems. Consider how many changes and new requirements organizations have been faced with in recent years:

- GUI interfaces
- Y2K
- Euro
- B2B integration
- Hardware obsolescence
- Web-enabling

Most organizations would be inclined to address the issues with small fixes, or adding software-layers that handle a specific issue.

However, at some point it will become apparent that the rate of change will not decrease, so there will be a never-ending flow of fixes and modifications necessary. Every year another 'layer' will be required to integrate or facilitate the 'next big thing' or the next business necessity. In the medium to long term it will be more cost effective to carry out the transformation of legacy systems into newer technology, to provide a better platform for change

The tools and services that *softwaremining* offer will reduce the cost and increase the associated benefits sufficiently to make transformation a viable option. Its offerings help in the Analysis, Documentation, Business Rule Extraction, and Automatic Rewrite of COBOL Legacy Systems into Java and C#.

This Paper describes the software translation and transformation landscape, identifies the particular problems that faces suppliers of such services and position *softwaremining* within the spread of options available to clients. Some technical descriptions of product elements are included and suggestions regarding cost savings and return on investment are also discussed, to assist the reader in understanding the potential benefits to their organisation.

***Organisations need to continuously apply continuous fixes to their legacy systems. Consider how many changes and new requirements organizations have been faced with in recent years:***

- ***GUI interfaces***
- ***Y2K***
- ***Euro***
- ***B2B integration***
- ***Hardware obsolescence***
- ***Web-enabling***

## Background

***“Too valuable to eliminate, too costly, complex, and interdependent to replace, existing IT applications consume a staggering 73% of application budgets.”*** Forrester Research

The Transformation and Consolidation of Legacy Applications can be justified for many varied reasons, most of which are well documented and understood:

- normalisation of multiple applications (acquired due to M&A activities)
- high cost of maintenance of some hardware
- forthcoming obsolescence of hardware - as in HP3000 mainframe
- future enhancement of the applications
- Web-enabling
- Connectivity to B2B exchanges, trading hubs, etc.

However until recently the only available option was incremental fixes, and the provision of additional layers to address such issues. The costs involved in a manual rewrite of the application are prohibitively high. Now the new analysis, componentisation and translation tools provide a more suitable option. These tools can go beyond simple translations – to allow better analysis of the original code – and conversion of them into a variety of ‘business’ suitable forms such as ‘business-rules’ in ontology (OWL) or as java code.

### *The Heart of the Problem*

The challenge for software engineers has, in recent years, been to develop a process of transformation that automatically creates Java applications that replicate the function of the COBOL applications yet are written as Java should be written such that it is understood by Java specialists rather than COBOL specialists. The key to achieving this is centered mostly on business rule extraction.

***Whilst manual re-documentation provides additional benefit of extraction of business rules from the existing application, the process is inevitably error-prone ...***

Consider the documentation issue. The automatic generation of documents from legacy programs can only provide an overall view of what the application does, using the program names, routine names and the associated comments as an indication of the functionality. This automatic regeneration can be significantly faster and cheaper than a manual effort but the manual process, in the hands of an experienced analyst, may provide additional benefits in the extraction of rules from the existing application. The problem, inevitably, with the manual approach is that the process is not only error-prone but with a complex application in particular, is guaranteed to introduce errors and mistakes. This is virtually unavoidable.

So, is the extraction of business rules beyond the capabilities of mining tools?

- Is it possible for tools to mine the legacy applications, and be able to separate the software-activities (e.g. screen or persistence handling) from the business knowledge?
- What can the system do with the business rules once extracted? Could they be put to use beyond the regeneration of a functional specification?
- Is it possible to create facilities to modify business-rules and *then* automatically generate a new set of code in Java/J2EE, or C#/ .NET?

## *Why softwaremining?*

After more than 20 man-years of development, a solution to these challenges is available in the form of a unique toolkit and specialist migration service. The *softwaremining* technologies go beyond providing the documentation, componentisation and automatic regeneration that is available from other suppliers.

*softwaremining* provides the means of extraction of business knowledge from the application program code and presents it in such a manner that it can be used in a machine-readable format.

The focus has been to allow businesses to take advantage of the new technologies to benefit the business, instead of using the new technology to address issues of old technology.

This differentiation goes to the heart of the 'transformation' strategy. The 'transformation' solutions mean enabling IT departments to concentrate on, for example, XML and B2B delivery of products and services and to increase competitiveness, rather than focusing on solving the internal technology problem of how to get two legacy systems integrated via the use of XML.

With the new advances in Knowledge Representation and *Ontology* fields with origins in AI - (see [w3.org/ontology](http://w3.org/ontology)) we now have a means for the representation of the extracted business-rules ready for integration in the 'Semantic Web' and ready for processing via off-the-shelf inference-engines.

***... the full benefits of realisation of a B2B exchange will take more than a simple 'XML' interface to the legacy system***

The importance of the web and B2B exchanges to support trading in the global village across the whole supply chain is now widely understood. Most organizations recognize that at least certain parts of their legacy systems have to be integrated. The mistake that many of these organizations make, often because of over-zealous but convincing sales talk from suppliers, is that the full benefits of realisation of a B2B exchange will take more than a simple 'XML' interface to the legacy system.

Consider the options: The migration of the new systems via manual re-write is too risky and too costly (because it is an extraordinarily labour-intensive exercise). Other options – such as screen scrapers, are unable to answer any of the aforementioned issues.

This is therefore where *softwaremining* comes in. Its tools offer sufficient business advantages, as well as cost savings, that finally justify an Automatic Transformation.

## Why Migrate? The Return on Investment

A 50% or more reduction in IT expenditure may seem impossible for any company with the massive legacy "drag" that Forrester talks about but some *softwaremining* clients are seeking to do just that. One typical example of this is the financial services organization that has a specific aim:

***"...to reduce (IT costs) by 70% across the division and to reduce back office costs significantly."***

Furthermore, this same client has a broader vision that goes beyond hard cost reduction benefits, saying:

*"Further benefits anticipated are improved data flows within the division, exploiting our information assets, streamlining existing business processes, better client integration using Internet technology and establishing a global, low-cost environment."*

How can large commercial organizations possibly expect to be able to do this? The answer lies in the transformation of legacy COBOL code to a distributed computing environment. As companies start to understand what is now possible *softwaremining* is hearing ever more goals and aspirations similar to those in the quotes above. This document therefore summarises these and other potential benefits available from *softwaremining's* migration technologies & services.

### *The Difficulty With ROI*

According to Computerworld (October 2001) 83% of companies do not track or measure ROI on technology projects. Even worse, those that do generally track timelines and costs but not value. Part of the reason for this is that it is often extremely difficult to measure ROI. Furthermore, tracking ROI can itself become an additional cost to the business. Another reason is that those responsible for the projects fear that in actuality the project will not give a good enough ROI and 'success' is therefore measured as nothing more than having completed the project within agreed budgets and timescales.

In the early days of ERP implementations many organizations were stung by the implementation costs. Although a client might spend a million dollars on software for a major implementation they would often spend ten times that much on implementation (source: Axon Solutions Ltd, 2002) when all factors such as cost of employees, external integration services, etc. were included.

Unlike application software itself, the *softwaremining* products may not necessarily be used over a long period of time. They are used to transform a legacy system into something more flexible and cheaper to maintain, ready for B2B and Web integration. So the value of the products is the difference in the value of the legacy system ( $V_L$ ) and the transformed system ( $V_T$ ), factoring in any costs of transformation ( $C_T$ ) and can be expressed as:

*Value of softwaremining Transformation Service* =  $V_T - V_L - C_T$

Cost of transformation ( $C_T$ ), whilst not zero, is nevertheless very small in comparison with total IT budgets of most major corporations and therefore to calculate the value of the transformed system one needs principally to consider the benefits of a transformed system over the original legacy system. It is the sum of the hard and soft, tangible and intangible benefits that make up a total value assessment of the technology. For any individual organization hard facts such as total cost of ownership would be relatively easy to assess and intangible benefits such as publicity to be gained within the industry from taking this technology route much harder to calculate and factor in.

*softwaremining* therefore offers a consultancy service to help IT Directors build a robust business case for transformation activities and the following is a list of key factors to be considered that, in total, make up the vast majority of all benefits to be gained. Note that some of these benefit statements rely on further activity (e.g. positioning for Web integration is only relevant once the Web integration is actually carried out) so some of the benefits of following *softwaremining's* transformation route are entirely dependant on the ability and desire of the organization to follow up on the opportunities that transformation creates.

## *Reduce Costs*

The primary objectives for transformation/migration projects are always the same - to improve operational efficiency and reduce the cost of doing business. The key ways in which the exploitation of *softwaremining's* services enables these to be achieved are:

- Reduction of IT hardware and software costs through the standardisation of IT platforms (replacing mainframes with cheaper servers)
- Corresponding reduction in head count
- Rationalisation of existing applications (reduced duplications)
- Improved operational efficiency and effectiveness via the sharing of IT platforms
- Integration of IT systems with clients & suppliers - leading to automation of accounts payable, streaming of supply chain management, reduced inventory costs, etc.
- Removal of expensive, 'specialist' systems with high support costs
- The production of 'cleaner' applications (removal of dead code) in a scalable environment - one result being increased throughput and thus greater operational efficiency.

## *Flexible and future-proof – Business Agility*

*"California's Controller said it would take six months or more to reprogram their COBOL-based payroll system to issue minimum-wage checks to state workers. Replacing a massive system like California's is expensive, but keeping a dying system on life support can be catastrophic." Tech Republic August 2008*

Today's business landscape requires collaboration with all manner of complimentary organizations and, as the rate of change only seems to increase, puts ever greater demands on companies to be extremely agile and hence able to respond to dynamic and unpredictable markets. It is in this evolving and changing world that *softwaremining* can deliver fundamental benefits to owners of legacy systems. Through *softwaremining* they can:

- React faster to new business opportunities - achieved via reduced complexity.
- Deliver an IT architecture to enable, rather than choke, business evolution
- Build IT systems that can change and be adapted to meet these business requirements as they occur

Once a company has implemented the first stage of B2B integration then new opportunities and business processes may become visible - such as the integration of an insurance claims system into the Web, which then in turn leads to opportunities to connect to B2B exchanges

for example. It is always core IT systems that need to be adapted to take advantage of these opportunities and systems can only evolve and adapt in this way if they are based on architectures and technologies that are flexible.

This is where legacy applications restrict or prevent the agility so desperately needed in today's business world. In a competitive environment those businesses with cleaner code, better documented systems, more flexible technology/architecture/framework can adapt and re-adapt much faster than those who need to apply changes to multiple layers of legacy code.

Invariably such changes are easier to develop and maintain within a uniform system comprising of industry standard third party components such as Oracle database, BEA or IBM Application servers, Java language, and Sun Servers. The alternative, integration and further enhancement of multi-platform applications, although perhaps offering reduced initial investment, will invariably result in further complications in the future development of the overall IT system. The end result of this is reduced flexibility, adaptability and productivity.

### *Improve information assets*

Accurate information is reliable information and hence much more useful. It is information on which business decisions can be made with confidence. High quality information that communicates something important rather than something irrelevant is key, as it offers knowledge and, as the saying goes, knowledge is power. *softwaremining* helps clients in three key ways regarding quality of information:

1. Improves the quality of the information at every point in the process
2. Provides the means for capturing clean, accurate and consistent data to improve decision-making.
3. Automates the end-to-end flow of data, transforming that data into information and hence useful knowledge.

Furthermore, if quality of information is increased and hence only useful information is being processed then there is a further increase in throughput and hence operating efficiency.

### *E-Business enabler*

The collaboration demanded from today's businesses' operating within the global village does not just come with flexible and adaptable systems. Although that is important, it is the power of the Internet that has produced a step change in the way organizations trade with each other, exchange information and interact for mutual benefit. Whilst the Internet as a whole was initially over-hyped it is now undervalued with much nervousness around IT investment in Internet technologies.

The application of *softwaremining's* tools to legacy code unlocks the value in the applications, enabling them to exist in an Internet-ready environment without huge investment in specialist e-commerce applications. In addition, they can then be connected to existing e-business architectures, ERP applications and B2B exchanges, etc. This can mean all the opportunities that e-business brings - closer client relationships, the locking in of suppliers, new channels to market, new market sectors, etc. - all because *softwaremining* makes the difference in connecting legacy applications to an e-business world.

## Case Studies

### *Case Scenario 1 - Rejuvenation of a Legacy Application*

A software company has a portfolio of products in the retail market sector. The primary application was written in the 80's in COBOL language and is responsible for a significant portion of their revenue. The Sales and IT directors have been looking at revitalising the product range with a view to increasing the company revenues from sales of their products. The increase in sales revenue could be generated through:

- **New Clients** - When faced with a range of competitor's options, such clients would invariably opt for new technologies. The reason for this is not the technology itself but the proven advantages of the technology. A centrally controlled browser-based or Client Server application is more saleable than a DOS based one. " ... Would any IT department standardize on a DOS based word processing application when there are other choices?"
- **Existing Client Base** - Selling upgrades to existing clients is the cornerstone of many software companies. A 'new technology' upgrade is as much a compelling reason as additional features in an established product. When, for example, Microsoft introduced Windows or Windows NT (New Technology) there was little change in terms of an operating system, control over hardware, monitor, file access, bios. The introduced changes were non-functional such as ease of use, longer filenames, stability, etc.

#### The Benefits of Legacy Application Rejuvenation To These Clients:

- **Reduction in deployment costs** - Typically this is associated with new installations or upgrades on individual PCs using a browser based deployment. Where deployment has been primarily on mainframes, then the cost of ownership can be substantially reduced by moving to smaller platforms such as Sun workstations.
- **Other Benefits of Internets, Intranets and Extranets** - Most companies are already aware of these benefits. However, building such capabilities into the existing product range is more of a challenge. In 1998, seeing the future of the Internet, Bill Gates ordered the product development team at Microsoft to build Internet features into the product ranges. For most other companies the strategy is a valid one - although more difficult to administer/manage (specially non-IT companies).
- **General Product Development Issues** - As with any software application, the product can benefit from additional features, enhancement and functionality. This becomes even more evident if various applications are successfully integrated together - e.g. an Inventory Management (IM) application linked to a Sales Order Processing (SOP) system. On their own these applications may contain most of the required features. But after the integration new requirement would soon creep in. "Having the inventory data at hand, it would be good if the SOP system could notify the IM system of the future / regular orders thus enabling ... ". With the long-term view that the application would eventually need to be upgraded into a new language/architecture, any further investment in development within current architecture/language may not be recoverable.

## *Case Scenario II - Documentation of Existing Software*

Recent Mergers and Acquisitions of two large financial Organizations have led to a large degree of duplicity of their IT systems. The duplicity spans across all of the IT systems - front office, back-office, payroll and etc. The initial rationalisation of these systems requires detailed documentation - listing detail functionality of each system - i.e. a description of what each system does, what the business rules are and how the system interacts with other systems.

The next requirement is to remove the duplicity and provide integration between remaining systems. The requirements outlined in this scenario could be present in any type of company in any business sector.

The CORECT tool provides detailed documentation from the structure of the code thus enabling the comparison of the different systems. Whilst the comparison of functionality is a manual job; *softwaremining* helps to provide a better integration platform for the remaining systems via upgrading them to the more current languages, industry standard 3rd party components (Oracle Database, IBM or BEA Application Servers, Sun hardware, etc).

## *Case Scenario III - Integration Issues*

A large insurance company has a host of IT systems for Treaty Management, Claims Administration, Policy Administration, Account Management, Underwriting, Financial Management and Payment. This functionality is provided via a series of custom made applications written on different platforms.

The main focus in integration is communication across different parts of a multi-platform system. However, provision of external channels to suppliers and clients (EDI and Extranet) is also needed. All the new channels are expected to re-use the existing internal components. Therefore all the components are to standardise on use of XML as a means of communication, i.e. all system interfaces - internal, external and EDI components to communicate with each other via a XML.

The CORECT tool generates code that can meet the XML level integration required here. The generated code utilises a flexible architecture with such requirements in mind. The framework layer may be adopted to work with XML messaging without any need to change a single line of business code.