

Application Modernisation

TECHNOLOGY AUDIT

SoftwareMining

SoftwareMining

Summary *SoftwareMining grew out of the perceived need for tools and services capable of supporting enterprises faced by the demand to migrate their core business applications onto new platforms. SoftwareMining can provide such support either through a professional services offering, through which translation can be undertaken largely off-site, or can provide a powerful tool kit to support internal transformation being attempted by a customer. The product element of SoftwareMining makes use of flexible heuristic approaches, and is based upon Artificial Intelligence (AI) programming language – the ability to configure the product to highly specific circumstances is very high as a result of this.*

SoftwareMining can act as a service provider or deliver product, but in either case supports the transformation of existing enterprise code from COBOL into more immediately usable and flexible equivalents, such as Java. Its AI-based tools deliver extensive reporting capabilities, coupled with rapid componentisation approaches. Although a small company at present, SoftwareMining is establishing a global presence via an expanding network of business partnerships and reseller networks, and Butler Group believes that its technology is well positioned to play a role in the enterprise migration market.

Although enterprise migration projects have been negatively impacted by budget cutbacks, like most IT initiatives, Butler Group believes that the market will become more dynamic in the near future, and that SoftwareMining is well positioned to expand along with it. The current market demand is for non-invasive solutions to issues with core enterprise applications, a market that SoftwareMining does not directly support – however, as market forces shift inevitably towards the recognition that the applications upon which the enterprise has become dependent must be changed eventually, the opportunities open to SoftwareMining will expand rapidly.

► MARKET ANALYSIS

The issue faced by organisations that have been operating core elements of their business based upon solutions that have been in place for an extended period of time is that those solutions were never designed to interact in a flexible manner with the rest of the enterprise. Although still performing vital tasks, some enterprise investments are simply out of step with modern business expectations and strategies – logic dictates that a choice must be made between either replacing these systems outright, or finding some means by which their functionality can be accessed in more dynamic, yet still cost-effective, ways. In many cases, and certainly in the immediate future of the Application Modernisation market, the latter option is receiving serious consideration at the enterprise level.

Non-invasive techniques, such as screen scraping, have been used to capture mainframe-based activity and data in order to present them in more user-friendly formats. The current trend in the market is to consider the use of added layers of functionality positioned between end-users and the core business applications, with Web service protocols, particularly eXtensible Markup Language (XML), being used to enable effective exchanges between front- and back-office applications.

Although these Web service layers do create further layers of complexity and additional requirements for system and process management, they also promise greater access to existing functionality without the risk of compromising the core applications that the business is dependent upon. Because Web service protocols pose little additional expense and are also inherently low in risk when deployed internally, their market value is currently compelling.

In direct contrast, invasive procedures and technologies involve direct alteration to the existing enterprise solutions, and tend to be seen as high risk. The applications in question are often poorly documented and understood, making the prospect of effecting change in a safe manner an uncertain one. Long-term, however, the benefits of successfully modernising an enterprise application, for example by transforming business logic into reusable components, or even migrating the solution to a different platform, can be considerable.

The organisation will be able to reduce maintenance costs, through the elimination of redundant code, and closely control application and process activity based on greater levels of understanding of what the system is actually doing. The core activity of the organisation will be future-proofed against new technology developments and business demands, which is not the case with non-invasive alternatives – but the perceived risk of tampering with business-critical systems often proves to be a powerful inhibitor.

The Application Modernisation market is truly horizontal, as companies in every industry sector and of every size are faced by the demand to improve the accessibility of their core applications and ensure their ongoing value to the business. In Butler Group's view, the choice is not whether or not to undertake Application Modernisation, but when and how the process should be done: and although a short-term solution using non-invasive techniques will generate rapid results, the only sure route to long-term future-proofing lies with direct, invasive, work upon the applications identified for change.

► OVERVIEW

The Application Modernisation market covers a range of products and approaches, with only the common theme of attempting to extract functionality from existing systems in more effective ways; whether emulation-based, transformational, or Web service-oriented, vendors are attempting to provide opportunities to acquire and utilise business processes based on older technologies with greater flexibility.

The dilemma for modern businesses is that although systems that have supported their efforts for many years are still productive and dependable, these self-same systems do not lend themselves well to modernisation, or integration with agile business strategies. Compounding this issue is the fact that technological advances can quickly relegate even the most contemporary systems to a legacy status.

This latter point reminds us that Application Modernisation is not purely a mainframe issue at all; the drivers toward modernisation begin to bite wherever business processes are undermined by poor interfacing between systems and users, as well as applications themselves.

SoftwareMining takes the view that in order to resolve issues with existing technology investments, it is necessary first to capture detailed business knowledge, and then use this information to generate business strategies. Clearly, mapping existing processes simplifies any subsequent componentisation and re-deployment into new services and formats. Simplicity is a key element of the SoftwareMining message, and a priority is to avoid adding unnecessary layers of complexity to enterprise systems that are already unwieldy.

SoftwareMining typically undertakes a service contract for the transformation of existing enterprise code. Initial analysis will be carried out jointly by SoftwareMining personnel and customer representatives, for example, specifying variables for conversion according to need. This situation may change as the company's services become more product-oriented, but it is very likely that service will remain a strong element for the company.

SoftwareMining provides tools to automate analysis, documentation, business rule extraction, and the rewriting or conversion of legacy code from COBOL into Java/Java Server Pages (JSPs). The demand for this type of automation is evident from the fact that manual execution of these processes is notoriously error-prone and time consuming, and is a significant demand on enterprise resource.

Butler Group believes that a wide range of functionality and disciplines will be required in order to extract truly valuable results from enterprise software projects, ranging between the discovery of application and program resource through to the management of new processes based upon components of business logic; every product and service attempting to support such projects should therefore be flexible enough to accommodate several, if not all, of the elements that now come under detailed consideration.

► DISCOVERY OF APPLICATIONS

In many cases, the legacy environment will be both complex and poorly documented. If applications are to effectively interact against this backdrop, modernisation tools must first act to inventory legacy systems, determine relationships in order to preserve the business process flow, and establish what data elements are present.

Ideally, this process should be clearly documented, or alternatively the modernisation tool should capture or create documentation from the legacy source itself.

SoftwareMining notes that documentation on existing enterprise systems becomes less reliable with age, and may be incomplete, or even missing altogether. The company provides tools capable of re-documenting systems in the form of Universal Modeling Language (UML) Activity Diagrams, which can be used to represent business process flows in each program. If desired, associated comments from the program can also be included in such charts, ensuring that the context of what is being presented is maintained.

Document generation is carried out using the following four categories of documentation:

1. Data, i.e., COBOL File Descriptions (FDs).
2. Dependency upon data items (FD Descriptions) found in each program.
3. Flow logic in each program, charted from Function names, and associated comment areas. Representation is carried out using UML Activity Diagrams.
4. Inter-program flow logic, again represented via UML Activity Diagrams.

► APPLICATION ANALYSIS

Having determined what elements of the legacy system are present through discovery processes, the next logical step is to analyse what they are doing. Analysis should identify key elements, such as Job Control Libraries (JCLs), process dependencies, and, most important of all, determine how much 'dead' code is present in order that this can be removed, reducing maintenance costs.

Impact analysis can be carried out using SoftwareMining tools, enabling an analyst to assess the effects of proposed changes to data structures – this enables sections of code to be clearly identified that will be affected, prior to changes being made. Additionally, the identification of code that is no longer in use can be automated in the same manner.

SoftwareMining adopts the position that code should be transformed into more flexible and modern forms following analysis. In order to automate this transformation to as great an extent as possible (simplifying the process and reducing demands on skill levels), the company has based its tools upon Artificial Intelligence (AI) principles. The tools themselves are programmed in List Processing (LISP) language and are driven by heuristics – the result is that the system as a whole is extremely flexible and easily customised into an exact match for specific customer needs.

► PRESENTATION LOGIC

The problem with legacy code is typically not that it is substandard, or inadequate for enterprise needs, but rather that its capabilities must be accessed via difficult-to-use interfaces. Application Modernisation tools may address this issue by dedicating functionality to the presentational layers of legacy solutions, in either an invasive or non-invasive manner.

SoftwareMining makes use of clear and accessible interfaces, enabling simple use of its complex functionality, but does not otherwise focus upon the management of presentation logic. This is appropriate, given that the aim of its products and services is to translate existing code, rather than merely open up new avenues of access to the mainframe.

► BUSINESS LOGIC

Separating the presentational layers from the business logic layer can be an equally important element. Business logic, when clearly defined, can be componentised for extensive re-use across processes, simplifying the development of new services and enterprise initiatives. The nature of legacy code, often obscure to contemporary developers, demands strong support in the enablement of business logic extraction and conversion.

SoftwareMining's heuristic-based approach restructures code into more legible formats, with a level of success of 90-95%. Although the translated code must therefore still be validated by an experienced user (perhaps a senior developer, administrator, or third-party expert), this will still be a much shorter process than manually changing the same code. As an example, consider that with 100,000 lines of COBOL code a manual rewrite would take around two years; with SoftwareMining, the same task can be completed in four or five weeks.

Once extracted, business logic can be converted into the following:

- Java, C#, or Borland Delphi/Kylix.
- Semantic data, with constraints turned into UML.
- Web Ontology Language (OWL).

The most recent addition to the SoftwareMining tool set is its business rule extraction module, which is also based on heuristics and AI. Generated code utilises Object Orientation (OO), with a well-structured thin framework. The coding of the tool has been kept as close as possible to standard Java language, and uses the same interfaces – this minimises the need for developers to acquire familiarity in specific aspects of Java options, such as diverse libraries.

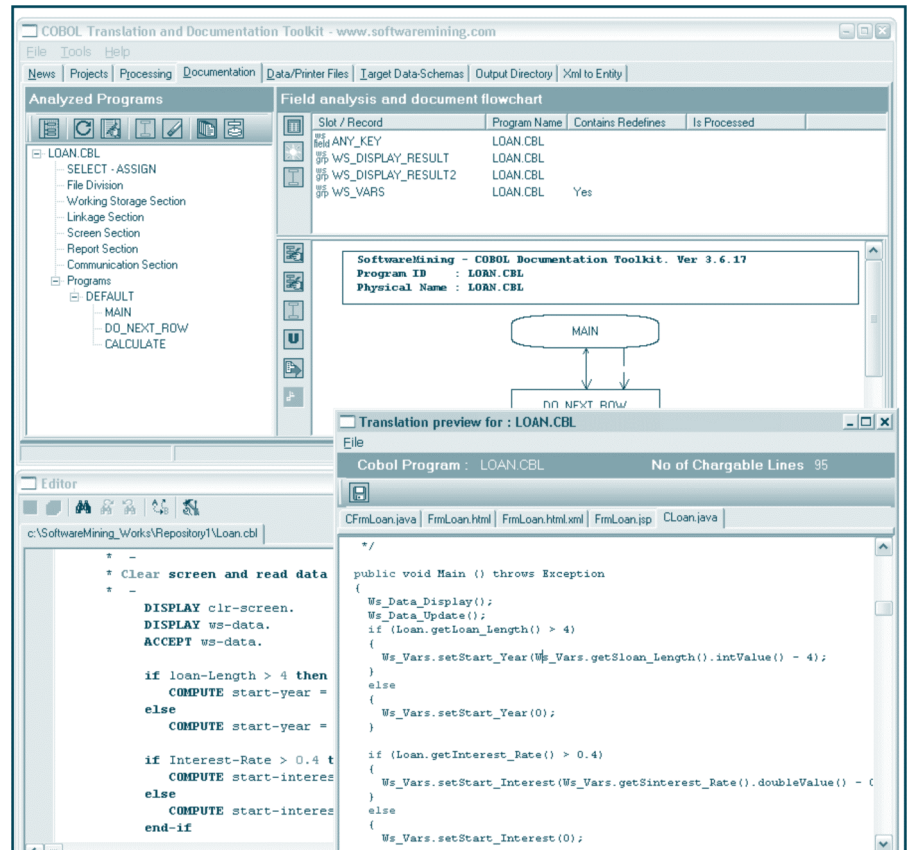
As a result of this adherence to standardised Java, SoftwareMining expects that applications that its services and tools have transformed will be easier to deploy and maintain across diverse architectures.

► BUSINESS LOGIC USAGE

The aim of capturing and improving access to the underlying business logic is to ensure that the business rules that depend on this can be more flexibly supported and utilised. Business rules must also be identified and documented in order to simplify any subsequent recombination into new services. One logical option is for this information to be stored within a business rule repository. Application Modernisation tools and services should also be assessed in terms of how they achieve conversions from legacy code, such as which languages can be operated with, and the question of how this conversion is carried out in real time should be addressed.

Business rule extraction is enabled by a series of pre-defined filters, based on operational and data constraints. Operational filters centre around categories of statements, such as screen manipulation, or persistence-related operating system commands. Data filters also serve to distinguish between syntactic and semantic constraints. OWL is used in the exporting of business rules, which enables these to be passed through inference engines. Options such as quality assurance, and even communication with the Semantic Web, become available as a result.

Applications and business rules alike can be automatically rewritten using SoftwareMining tools, into either Java J2EE or C#. Integration can be eased in framework-based architectures via the use of XML and Web services, with the end result being flexible code that can be executed upon a variety of client/server or distributed systems.



Cobol Toolkit Screenshot

► COMPONENT MANAGEMENT

Breaking legacy processes into reusable and highly flexible components is the goal of most Application Modernisation vendors, as this step alone greatly improves application development efforts, and also reduces operational costs. The level of componentisation varies considerably from vendor to vendor, according to focus and core strengths. Component management is a key area to vendors and end-users positioning themselves to take advantage of the Web services model.

An interesting element of SoftwareMining's approach is that although it is inherently componentisation-oriented, the heuristic element of its functionality enables results to be delivered in a very flexible time scale. Instead of being committed to a total analysis of the code base prior to any benefits of translation being delivered, code can instead be opened up to integration and/or interfaces as the project progresses. This form of componentisation is not only more rapid, but also cost-effective; it also delivers the opportunity for reducing the time before transformed software can begin generating Return On Investment (ROI).

No changes are made to existing processes or code until the transformation is complete and has been validated. Once this step has occurred, data migration from the original system can be carried out, in what is intended to serve as a seamless transfer between contemporary and legacy platforms. Processes should therefore suffer no ill effects, or be compromised in any way, as a result of a correctly managed transformation.

► DEPLOYMENT

SoftwareMining states that until recently, in most cases, its customers would opt for a translation service, in which consultants from the company itself carry out analysis of existing code and systems. This avoids the problem that detailed knowledge of transformation issues is relatively scarce, and may not be present at all in some companies that are facing the demand to transform their existing code. SoftwareMining can carry out analysis and transformation on- or off-site, depending upon the customer's preferences and requirements.

SoftwareMining has been developing the software elements of its business with a view towards positioning them as marketable products, which may alter how it works with customers to some extent; nevertheless, the fact that its service can be obtained cost-effectively is likely to ensure that use of product and service remains as a commonly adopted option. The company's COBOL toolkit, CORECT, is at release version 3.6, and runs on Microsoft Windows platforms. CORECT supports most dialects of COBOL, including embedded SQL, and CICS.

In the initial stages of any transformation project, the customer will need to provide detailed technical information concerning the existing applications that have been identified for migration, and if the customer intends to use SoftwareMining tools rather than service, then some training in the use of the products should be considered. Testing and deployment phases will also require an Integrated Development Environment (IDE), such as JBuilder or Visual Age, along with database licenses and an application server. The alternative is to accept SoftwareMining's service offering, in which instance these responsibilities will largely be handled off-site.

The deployment time will be entirely dependent upon the scale of the transformation being proposed, and it is therefore very difficult to provide an average. In general terms though, it is reasonable to expect that the automated features of SoftwareMining will deliver results in a vastly shorter time frame than manual alternatives. Application 'tuning' is estimated as taking two to four weeks per 100,000 lines of translated code; again, this is significantly faster than manual processes.

The considerable differences between projects that SoftwareMining can support also makes it difficult to estimate average costs for an implementation. Two basic service types are available, a translation only service, where costs run as low as less than 0.30 UK pound (30 pence) per line (although it is worth recalling that enterprise code can run to millions of lines, this is still an attractive cost), or a full turnkey service, involving translation and subsequent testing. This latter service will usually average less than one UK pounds per line of code.

In order to deliver a full business solution, SoftwareMining must integrate with the enterprise application server. Given the emphasis upon Java, it is natural to find that Java-based application servers, including the market leading BEA WebLogic and IBM WebSphere, along with the popular Apache Tomcat, have been prioritised for such integration. Initially, the translated application will run side-by-side with the existing source during a testing phase. Once testing generates satisfactory results, a data population phase can be undertaken, and finally the new application will go live and replace the original.

► MARKET STRATEGY

The goal for SoftwareMining is to transform existing enterprise code from mainframe or legacy status, into modern and more flexibly deployed formats such as C# or Java. This leaves behind the COBOL and any dead code, and offers the enterprise greater flexibility in its selection of new technologies. Translated applications will 'behave' in the same way as the originals, and there are no obligations to change business processes as a result of changing technology. The difference is that the translated applications have been rendered easier to interact with, and cheaper to maintain as well.

Competitor tools also possess the ability to rapidly transform code, but SoftwareMining differentiates itself by rapidly producing what it refers to as 'maintainable', full Java code, rather than Java byte-code. The code produced by SoftwareMining is open, not closed, and the company believes that Java byte-code, because of its closed nature, is not significantly different to the existing COBOL code that it is intended to replace.

SoftwareMining makes extensive use of Web services standards and languages, such as OWL and XML. However, the majority of Web services associated with the Application Modernisation market are aimed towards creating new channels of connectivity through to mainframe applications and programs, while SoftwareMining takes the more ambitious step of transforming those same applications entirely. Timing is the great differentiator for SoftwareMining. Results of componentisation can be rolled out at a brisk pace using the heuristic approach, in direct contrast to the lengthy 'traditional' results of componentisation.

The demand to make more effective use of existing enterprise investments, particularly those applications that are still responsible for generating critical elements of the corporate ability to deliver internal and external service, impacts upon horizontal and vertical sectors alike. A particular vertical opportunity for the product aspect of SoftwareMining lies with consultancies, who are ideally positioned to make use of customised versions of its COBOL toolkit. To support this opportunity, SoftwareMining is negotiating the possibilities of opening up parts of its source code to select associate consultancies, which would greatly improve the ability of such users to deliver service to their own customers.

SoftwareMining reaches the market via a network of business partners, Value Added Retailers (VARs), and associate consultancies, as mentioned above. The company is searching for appropriate partners to continue the expansion of this network, and distributor channels are being sought for the global market. Butler Group believes that this is a sound strategy at this stage of SoftwareMining's own development. It has its own in-house service professionals, and is expanding upon the capabilities of this arm in line with customer requirements.

Butler Group believes that the SoftwareMining approach is a very good fit for this niche enterprise market. The flexibility, cost-effectiveness, and rapid results that can be delivered add up to a strong business case for the use of the services or product on offer. The most serious issue that the company must deal with is its size – this is still a small, Europe-centric, company, and many potential clients are accustomed to having enterprise software problems resolved by large global providers. Disregarding this company because of its size would, however, be a mistake, as the results it can deliver make it worthy of the most serious enterprise evaluation.

► KEY FINDINGS AND LOOK AHEAD

KEY FINDINGS

- | | |
|---|---|
| <p>✓ Rapid, heuristic-based approach to componentisation.</p> <p>i Based upon standardised Java and C# to simplify usage and integration.</p> | <p>✓ Clear documentation can be quickly generated about the existing environment.</p> <p>✗ Small company in an increasingly competitive market.</p> |
|---|---|

Key: ✓ Product Strength ✗ Product Weakness i Point of Information

LOOK AHEAD

Release version 4.0 of the CORECT tool will be released in mid-2003, and SoftwareMining releases updates each quarter. The developmental roadmap includes support for the transformation of Fortran and Basic languages. In addition to developmental efforts, SoftwareMining is expanding its network of business partnerships and reseller channels.

► COMPANY PROFILE

SoftwareMining is headquartered in the United Kingdom, and plans to open offices in the United States and Latin America. The company is privately held, and was founded in 1996 by Dr Cyrus Montakab. The company's main aim has been to develop its product element to the point where this could be marketed, and this goal was attained in 2002.

SoftwareMining now has clients across Europe utilising its products and services, and has processed over one million lines of enterprise code in the last 12 months. SoftwareMining's customers include Deutsche Bank, SwissRe Insurance, and Sema Group. Sales, marketing, and business strategy are managed in the UK, whilst development is undertaken in Russia. Employee growth in all divisions is reported to be set for significant rises in all divisions in the coming months, and expectations are high that the company's presence and sales in the US will also ramp upwards sharply.

► CONTACT DETAILS

SoftwareMining

Allerton House
50 Lower Green Road
Esher, KT10 8HD
UK

Tel: +44 (0)1202 653516

Fax: +44 (0)1202 653034

E-mail: sales@softwaremining.com

www.softwaremining.com

Important Notice:

This report contains data and information up-to-date and correct to the best of our knowledge at the time of preparation. The data and information comes from a variety of sources outside our direct control, therefore Butler Direct Limited cannot give any guarantees relating to the content of this report. Ultimate responsibility for all interpretations of, and use of, data, information and commentary in this report remains with you. Butler Direct Limited will not be liable for any interpretations or decisions made by you.