



Application Portfolio Management

Real problems need a real solution

By Peter Szirmak

Executive Summary



We did it for decades - Building new systems, introducing new technologies, massaging - or sometimes outright bastardizing' - large IT portfolios (as a result of corporate mergers and acquisitions), buying package software solutions and fitting them to existing systems, committing to a new standard every few years, switching to a new platform, etc.

A young industry trying to find the best and quickest way to gain advantage over the competition - or simply introducing something drastically new every few years just for the sake of keeping up! For the last few decades, in fact for most of the life of Information Technology as we know it, we haven't had more than a few years go by without some latest and greatest storm changing the landscape, again and again.

Back in the early nineties, computing became available to the public at large. A new breed was born: a young generation who could learn how to use the tools quickly (even at home) and could do wizardry with emerging client server then web development tools. Sometimes in a few days, systems could be built; seemingly in front of your eyes. And so we developed systems much faster - often without caring about documentation, architecture or even simple coding standards. Today, many of those applications linger around without anyone daring to touch them.

We spent a large effort to make systems Y2K compliant. In many cases, we actually made the systems date compliant for another eight thousand years. Too bad that most of the programmers who actually understand what these systems do will retire in less than a decade...

Lucky is the CIO who has a single or even just a few platforms to manage. Most are dealing with a myriad of platforms, languages, standards, in-house developed architectures, methods, communication techniques and customized packages (where sometimes the custom code is larger than the package itself). Documentation is outdated or simply non-existent. But it is also scattered around: from modeling tools to word processors to comments buried in the code.

And lucky is the programmer who gets a call at 3 a.m. in the morning and can fix the problem with absolute confidence. Do we really know what a change might cause downstream?

A simple question such as "What do you have and how is it related?" would find its answer only after serious digging around - information about information systems and how they actually work together, let alone by themselves, is not evident. Information is scattered and is fragmented - and much of it lives in people's heads - programmers who dug into the code so many times they actually remember...


At best, companies have a high-level understanding of their *physical environment* (e.g. a given application runs on a given hardware platform) or some very high level documentation of how business systems relate to software assets (e.g. a given business system contains certain applications).

Organizations will also typically have a good *detailed* understanding of their various IT assets (may have implemented a document management software solution or application mining tools for specific platforms) but each area is separately managed by the enabling software.

As a consequence, our industry has experienced a steady increase in the cost of maintaining Information Technology assets with less time and money left for new initiatives. Forrester reports that 73 percent of IT budgets are spent on maintenance activity. And even with an ever increasing effort to understand the current application portfolio, more and more companies are faced with embarrassing computer glitches or outright down times and loss of business. The rule seems to be simple: if it hasn't happened to you, it will.

Enter Application Portfolio Management (APM)

There is a new breed of tools that promise you to automatically discover the application portfolio, its relationships and impact points. These tools also provide summary and statistical information about the portfolio and how it changes. Companies who have pioneered even early generations of these tools experienced 20 to 30 percent savings of their



maintenance costs - with an ROI of less than a year. With such a large promise, it is perhaps not surprising that Forrester predicts a 25-fold increase in the use of these tools, by 2008.

Recently, Forrester identified us as leaders in this emerging field. In fact we put forward the need for a comprehensive application relationship management solution in 2002, even before the term Application Portfolio Management itself took hold in our industry.

This paper describes the need, concepts and goals of Application Portfolio Management. More importantly, it highlights two major aspects of Application Portfolio Management: **coverage and implementation**. These two characteristics of an APM solution in fact form the very backbone of success: without proper attention to these, one may follow a promising dream of control over the application portfolio - without ever really and effectively achieving the goal.

Coverage refers to the ability of an APM tool to understand any platform, any language, any in-house developed architecture or communication method, any implementing dialect - in short any IT component you may have in your application portfolio. Without such coverage an APM tool cannot claim to manage the full portfolio - at best these tools are extended application or infrastructure mining tools. While coverage of some area is better than nothing, the more you leave out, the less effective the solution will be.

Implementation refers to how an APM solution is best applied against your application portfolio. A few questions are worth asking:

- Is it possible to create a fully automated tool, available out-of-the-box, which will parse in *any* component of your application portfolio? Our industry spent decades to create the current complexity - is it possible (let alone cost effective) to create a tool that will automatically understand the complexity built into your application portfolio and would handle any possible combination, out-of-the-box?
- Is it possible to create a fully automated tool, available out-of-the-box, which can be effectively operated by your in-house staff - learning how to customize the tool and fully understanding the literally thousands of possible parsing errors such a complex tool would inevitably produce when applying against a constantly changing IT environment? Or would you have to build a significant team to support such a tool, made up of parsing experts, grammar technology experts as well as experts of the tool itself? Would the cost of maintaining this team offset the resources saved in maintenance? What is the real, total cost of ownership?

Real life problems require real-life solutions. In our development shop, we believe that there are no magic bullets in life. A comprehensive, effective solution must take into consideration the issues described above. The answer, perhaps, is surprisingly simple - and perhaps lacks the sexy promise of magic. But it actually works.

The APM solution that can answer to these issues must handle, out-of-the-box, most of the standard languages and platforms used in our industry (from COBOL to Java to C and so forth). But it must also be able to handle the actual implementation of these languages - i.e.. the parsers used must be customizable to adjust to the actual usage of these languages. It must be able to incorporate the many unique rules and methods you may have in your particular application portfolio. Most importantly, the solution must offer the ability to generate parsers for any structured source, any architecture, in-house developed or bdd-ball' languages and platforms - in other words the solution must be expandable literally without limits.

The APM solution must also provide a clear, true picture of total cost of ownership - such that the implementation and continuous usage would not result in a possibly larger resource and financial drain than the savings sought.

Our unique APM solution does exactly that.

The goal is to provide an automatically generated and maintained super meta repository' of all relevant relationships of an organization's technology artefacts, made visible via an enterprise portal to support strategic planning, IT development and day-to-day maintenance. The APM solution should parse in any artefact, regardless of platform, language or implementation method.

What is ARM - Application Relationship Mapping??

ARM provides a set of technology to create and maintain a central repository of all your IT assets. It is a *true and complete* APM solution that allows your organization to understand its IT assets, including documentation, application source code, business objects, hardware and all other components. It provides a continuously current view of: what assets exist, what assets are used / unused, how those assets relate to each other, and what relationships exist within those assets.

The importance of implementing such a roadmap cannot be over-emphasized. All aspects of information technology management benefit from such a strategy: from everyday development and maintenance activities to strategic platform migration, application integration, and outsourcing or re-platforming projects. All IT projects hinge on understanding what assets exist and how they need to change to support the new business realities. The ability to analyze information effectively, in a timely and quality manner, directly affects the organization's use of resources and its overall IT expenses..

The ARM roadmap provides organizations with a strategic advantage over competitors..

Integrating Information

To ensure that business objectives can be met on a repeatable and cost-effective basis, IT organizations need a connected view of their assets. In order to reduce costs associated with the planning, designing and executing of IT projects measurably and permanently, organizations need an automatically generated knowledge repository that can answer such simple questions as 'What do I have?;', 'How do my assets relate to each other?;', 'How do they depend on each other?;' or 'What is the impact of changing / removing / adding components?;'.

Making Information Visible and Understandable

Relationship information is inherent in the IT assets, but without a full APM solution, it is fractured and hidden. Application code, hardware, documentation, etc. all contain relationship information within them. ARM allows organizations to pull such information to a higher level of integration easily and automatically. It makes those relationships visible and provides context for them. The ARM strategy also defines how the information will be dynamically maintained, to ensure its future currency and accuracy. The resulting integrated, current, and understandable view of the assets provides maximum value to the organization.

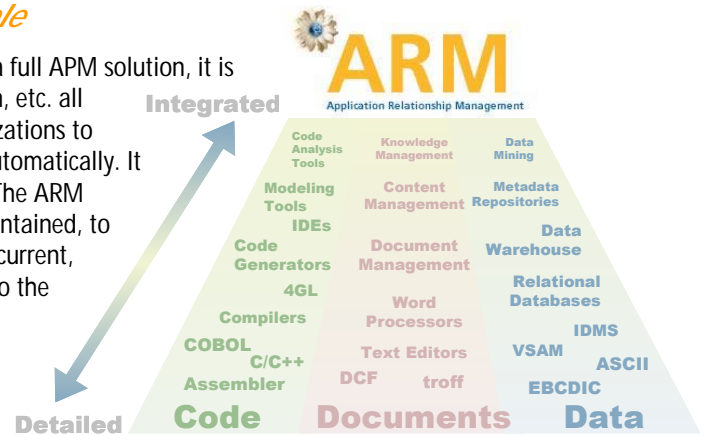
ARM can be looked at as the new enabler of the evolution from a detailed to a higher level of understanding, bringing progressively integrated views of code, documents and data under one 'umbrella'.

Leveraging Other Initiatives

ARM leverages investments in other initiatives, such as Knowledge Management or Application Mining, by connecting all these and other areas of the IT organization.

This figure depicts how ARM relates to those other IT initiatives.

The ultimate success of individual initiatives, such as Document Management and Application Mining initiatives, will depend on how well these areas connect to each other. Having a solid document management process by itself will not answer the questions: 'How do my documents relate to what I have?;' - or 'Which documents contain relevant information?;'.



Similarly, leading application-mining tools such as CAST and Revolve will provide the organization with detailed information about their various application platforms. However, they will *not* answer the question: How does my client server application relate to my mainframe applications or to the documents and other objects that I have?'

Managing Business Initiatives Effectively

ARM provides a real context for managing initiatives from a variety of disciplines within the organization effectively and efficiently under the umbrella' of ARM. This ensures that all your initiatives are consistent with and tightly integrated too the overall strategy of the organization (no silos)..

Transitioning to Maturity

ARMs technology set moves an organization from the current norm of effectiveness to the future standards of performance required to compete and excel. The industry environment described below shows some of the immediate challenges that are slowing this necessary transition and reducing organizational effectiveness

These challenges must be overcome in order for an organization to survive and progress. Only a true APM strategy, such as ARM can position the organization for success..

Moving from the Current to a Desired State

Current State of the IT Industry



The inability to provide the right knowledge to the right people at the right time is expected to cost Fortune 500 companies over \$57 billion over the next two years'(International Data Corporation).

This widely quoted statement summarizes well-known realities in our industry. Slow, costly execution of projects results in delays for bringing new business initiatives to market. The Standish Group Chaos study for the year 2000 found that 23% of projects outright failed and 49% were challenged. Of these challenged projects, the average project was 189% over budget, 222% behind schedule and delivered only 61% of originally specified features.

Silos of Information

For IT initiatives, the amount of information to be examined and analyzed is growing at an exponential rate. At the same time, information is decentralized, stored in different locations, and often in different formats.

To address these challenges, corporations have typically installed software to assist in organizing information on: source code, documentation, execution statistics, business models, etc. However, for each silo of information, the software tool chosen is usually platform-specific, and/or implemented as a separate instance for that area of operation (e.g. department-specific).

The set of information management tools used within an organization might be:

- Separate application mining tools each providing information about source code on specific platforms
- Separate modeling tools to organize process and data models
- Document management software that targets documentation only

- Ad-hoc tools (e.g. departmental macros, etc.)

None of these tools was designed to communicate with other areas in the organization where related information exists. Discovering and documenting such relationships is left for manual research and understanding.

The time and effort spent in information gathering is directly related to the accessibility of information. A typical analysis will need to gather information kept in people's heads, in hardcopy only, on the mainframe or on the PC, in various formats. Some information is in a completely raw format (e.g. source code) or is kept in the proprietary format of the tool organizing the given information (e.g. application mining tools).

Lack of Qualified Resources

The proliferation of knowledge management and other tools, all intended to organize and maintain these various silos of information, necessitate that an IT professional must become a Jack of all trades; understanding multiple environments and tools. The tools themselves are often complex, each requiring significant experience and understanding.

In almost every IT organization, initiatives exist to attract, train and retain qualified resources. However, having deep understanding and knowledge of how IT assets relate to each other is usually the privilege of a few. These resources have obtained their knowledge through many years of working with these assets. This deep individual knowledge is typically not captured anywhere else in the organization. As a result, IT organizations rely unduly on the knowledge of a few key individuals. These resources become the bottleneck in initiatives. As a result, all projects can expect a slower than optimum delivery and may experience costly delays.

Having knowledge centralized in the head of a few individuals also results in slow and costly training of new resources. Building organization-specific IT knowledge takes many months, if not years. New resources become truly effective much slower than desired.

Mis-Utilized Tools

Corporations spend significant amounts on purchasing and maintaining tools to provide organized information on specific areas of IT. At the same time, tool vendors struggle to keep their tools utilized by their clients but, more often than not after a relatively short period of time, many of these powerful tools become 'shelf-ware'.

The underlying reality is simple: the tools themselves are designed for specialists, but are purchased for general use. The information available via these tools is extremely important for specialists but is much more than most IT professionals need on a regular basis. Lacking any other way of getting at the information, corporations purchase a much larger number of licenses than they truly need, in the hope that occasional users will use the tool for getting the information they need. In reality, occasional users become non-users and return to old-fashioned, mostly manual research (once the effects of the original training fade, the tools are perceived as complicated and unfriendly). This results in a large number of licenses becoming 'shelf-ware'. Typically, after some time, tool usage is reviewed, the overall return on investment is judged to be low and all licenses are terminated, resulting in the return to manual analysis for all. Soon after, a new tool is purchased, starting the cycle again...

We must move beyond these current challenges to a coherent and integrated environment where we can use integrated tools to best advantage - only in such an environment can we excel.

But integration is a large and complex task, well beyond the available delivery capability of a typical IT organization. As mentioned elsewhere, more and more time is needed just to keep the lights on. Organizing and developing the necessary technology to make such organization a repeatable and sustainable exercise cannot be achieved within the current spending realities and constraints, in most cases. This is in fact a vicious cycle: no time and effort is left to organize and understand the IT environment better; hence it becomes less and less possible to carry out tasks effectively and the environment becomes even more complex. One clearly needs to break out of this cycle; the sooner the better.

Desired State

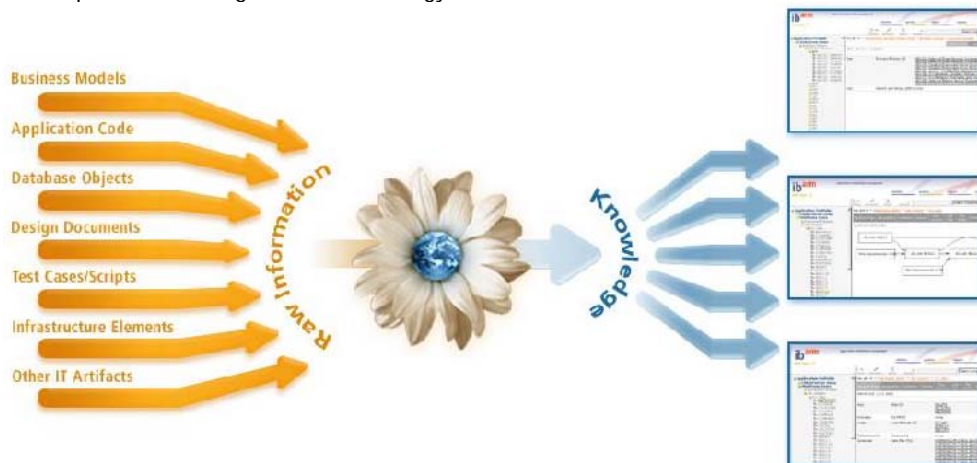
The Gartner Group predicts that IT expenses (as a percent of overall corporate expenses) will grow from 30% today to 50% by the year 2010. Such expenses will continue to be challenged, resulting in progressively severe budgetary pressures for IT. This is not a viable business environment and has to be changed through a new vision.

Delivering more with less requires an overall strategy for a significant and ongoing productivity increase. Current initiatives such as: document management, application mining, etc. are not sufficient in themselves to achieve this turnaround. The business information needs to be fully integrated and all of its related initiatives need to be connected in order to produce the desired productivity increase. If there is no overall strategy, then these initiatives may only add to the creation of information silos and other problems discussed above.

ARM provides the overall strategy that connects those initiatives to the complete set of business information driving the organization.

Knowledge Centric IT Strategy

The figure below depicts a knowledge-centric IT strategy, as defined via ARM.



Relevant tools may continue to organize and manage specifics of a given IT asset. However, ARM implements a repository of relationships between and within those assets, thus providing a much-needed higher-level understanding of how those connections and dependencies inter-relate.

A single interface (preferably via the corporate intranet and often made available through a corporate portal) establishes the gateway to all such IT information. Regardless of the actual location or platform of the information, IT professionals will have a single, consistent source for all their information needs.

Relationships within and between IT assets will no longer be the hard-earned knowledge of a few but will be automatically generated into the ARM repository, for all to see. Business rules buried in code or documents will become visible to all.

Existing investments into tools will provide a much greater return. Specialists may continue to use the given tool but information required by occasional users is automatically pulled a level up into the ARM repository. By doing so, fewer licenses will be required but the given tool becomes an integral part of the organization's information infrastructure and

thus will continue to be used. By implementing a properly utilized tool environment, both IT organizations and tool vendors will ultimately benefit.

Benefits of a full APM solution



Implementing ARM will create significant improvements in operational efficiencies, resulting in (conservatively estimated) cost savings of 15-20% of IT budgets related to projects.

The key to ARM is changing the current, static knowledge capture and maintenance to an automated, dynamically maintained process, thus providing the following key benefits:

- Protect IT investments by ensuring their cost-effective usage and evolution
- Standardize IT asset information, regardless of the original platform or format
- Enable knowledge proliferation to occur across the IT organization

ARM will enable tangible cost savings in practically all areas of the IT organization, from strategic planning through development, integration and enhancement projects, to everyday on-call maintenance. Three specific areas of the Systems Development Lifecycle will see the most significant benefits: Estimation, Analysis and QA/Testing.

With ARM, estimation efforts will be more accurate, and many times faster than currently possible. Such efforts currently necessitate extensive manual efforts by a few key resources. With ARM, complete and accurate information is available to as many resources as required, in immediately actionable format.

Reducing dependency on a few bottleneck resources and at the same time facilitating the effective analysis of applications, documentation, etc. naturally results in cost savings at the front end of projects. Easily accessible, complete and correct information will also enable effective test plans to be created, reducing overall costs and ensuring the success of QA and Testing phases. The overall quality of project output will increase accordingly.

Organizations will also benefit from reduced training costs, both formal and informal. Utilizing the corporate intranet to access information reduces the need to train in specialized, more complicated and diverse tools. At the same time, by automatically pulling information inherent in existing assets to a higher, common level, the need for unstructured, informal training (e.g. new resources trained by existing personnel) will be reduced.

Implementing ARM

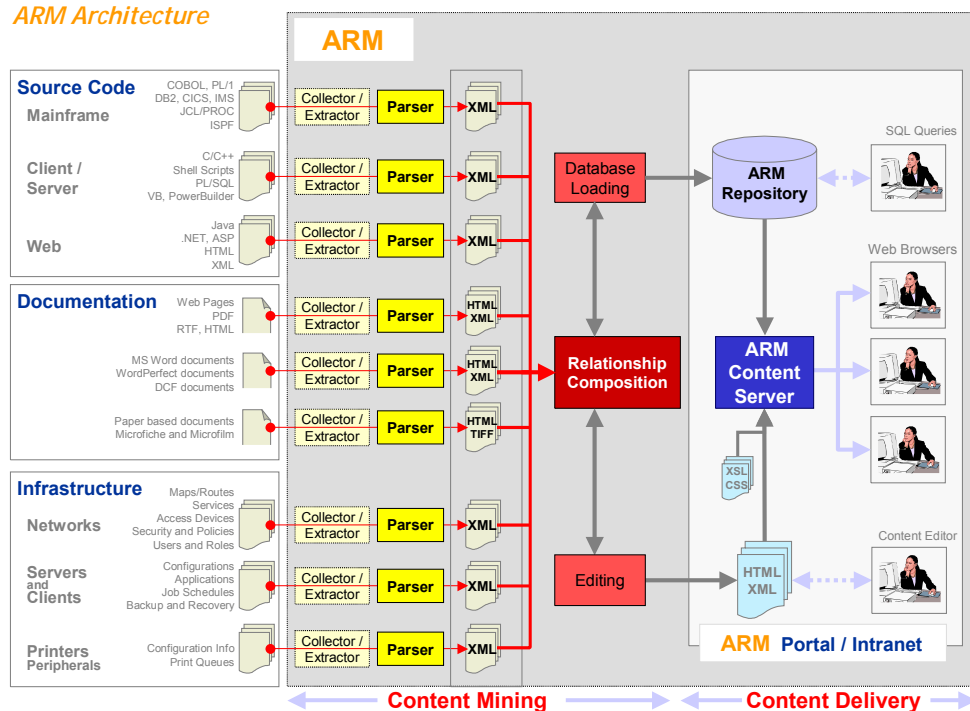
Enterprise Software Solution

The ARM Enterprise Software Solution follows the approach described above and defines, implements and maintains a practical APM solution, with two key distinct characteristics:

- 1) It covers the full application portfolio - not just the standard platforms and languages and not just the dialects and coding techniques commonly used in the industry.
- 2) It is deliverable without significant internal resource allocation and expertise to create and maintain the required, sophisticated technology connections. In fact, it provides the results, without having to spend effort to architect, develop, deliver and maintain the necessary customized APM software capable of such a complex task.

The Technology Deployed


ARM Architecture



ARM Architecture, V2.0, October 2003

Coverage

As mentioned elsewhere, it is practically and literally impossible to create a tool that would, out-of-the-box, cover the myriad of languages, platforms, coding techniques and in-house architecture implementations a typical organization would have. But without such coverage, the APM solution at best would be a band-aid; implementing yet another tool to manage a portion of the portfolio - in reality the most manageable part of the application portfolio that is standard enough to understand by an out-of-the-box tool.



While such a partial APM solution will still bring demonstrable benefits, it stops way short of what true Application Portfolio Management is set out to accomplish: *enabling the understanding and management of the application portfolio in its entirety.*

A sometimes overlooked but key shortcoming of partial solutions is the logical and hard-to-counter argument of application experts for not using the solution: they still need to 'dig into the code' for areas the tool does not cover. More often than not, this leads to the APM solution becoming 'shelf-ware', as IT professionals choose the 'old way' of application analysis (manual digging into the code) firstly out of necessity. Increasingly and over time, the tool will be less and less used, eventually getting back to the very state it was supposed to eliminate. A significant cultural change of how analysis is done will not be helped if the tool trying to solve the issues truly does not answer all the questions the analyst may have.

ARM solves this problem, by providing 100% coverage of the application portfolio, regardless of platform, language, dialect or implementation method. It provides such coverage by, firstly, having a significant set of parsers to cover a large percentage of the portfolio. However, even this set is customizable and tunable to the actual source being parsed. Using the technology, we have processed hundreds of millions of lines of source code. ARM takes advantage of that experience and allows the technology to be quickly adjusted and customized to the actual application portfolio to be parsed in.

Secondly, the technology is designed to be expandable to connect new parsers, generated (as part of implementation) to deal with the customized or less than typical source code the organization has. Almost any ARM implementation will involve, as a matter of fact, the generation of customer-specific parsers. This exercise does not necessitate any customer involvement other than attending a short, facilitated session, to allow ARM implementation specialists to gain in-house knowledge of any grammatical specifics of the source code or coding technique being utilized.

Only with such expandability can an APM tool truly cover the full application portfolio, regardless of what it is. ARM literally covers everything and anything in the IT portfolio: every IT artefact can be parsed in and connected to the rest of the portfolio. Such a solution can also be targeted to answer very specialized and unique, real life problems. For example, in a recent implementation, on top of the typical COBOL, PowerBuilder and other source code, components such as CoolGen models, RACF logs, MQ transactions and CA-7 schedules became part of the repository - to allow the full, end-to-end understanding of the given, complex portfolio.

Clear, Measurable Benefits

ARM provides clear benefits to our clients.

- * Turn-key solution, providing the results, not just the means
- * No hidden costs, clear ROI
- * Minimal client resource requirements
- * Fast implementation