

Business Process Management Implementation

Nigel Leeming

(*Director and Chief Technologist, Ivencia Ltd, United Kingdom*)

(Email: nigelleeming@care2.com)

Abstract

Business Process Management is a complex subject on two levels. First is the business thinking that will bring BPM into an enterprise and the consideration of how the enterprise will change to realize the benefits: this part of the subject has received some detailed coverage[1]. Second, careful consideration must be made to how it will be implemented enterprise wide, for this is where its major benefits lie. The intent of this paper is to cover the second of these points, as a guide to those tasked with implementing BPM.

The most fundamental concept of BPM, which is the automation of process is examined first. However, this is not enough to understand how BPM redefines the enterprise, nor how it might be implemented. To satisfy these more demanding points of interest, we must delve into the terms used to describe BPM and its products, we must examine how people work from a machine-logic point of view and how this logic is reproduced in the many products currently available.

Implementation requires an integrated enterprise architecture to allow communication across all corporate systems. An architecture to meet these needs is detailed in Automation Oriented Architecture.

Finally, an examination is made of the pitfalls a BPM implementer might come across.

Introduction

The term *Business Process Management* (BPM) means many different things. It is a catch all term for grouping methodologies and products which converge upon business processes. Product and service vendors, anxious in their endeavours to differentiate their offerings, give the term ever further depth.

At its most basic, it is a means of automating business processes, such as *process insurance claim*. At its most complex it is a way of defining, running and improving an enterprise.

It assumes some form of handing over the knowledge contained in manual processes and legacy applications to a BPM tool, which is then empowered to perform the processing on behalf of the enterprise, interacting with people and systems as defined in each process.

As the BPM tool is aware of the processes, business rules and information flowing through those processes, it is able to produce audit and management information, which when coupled with the reduced human workload, enables the continuous optimisation or re-engineering of processes.

It will enable governance, help to meet Sarbanes Oxley (US), Basel II (Europe) or FSA (UK) requirements. It will manage workflow, worker collaboration, presenting a single view across many processes and legacy systems.

It can be considered an evolutionary step from automation in the production line to automation in the office, producing a uniformity of process so that one worker's output will match another's.

This is what it can do. BPM vendors claim it can save money as computers are doing the work that previously took manpower. Yet coming to terms with where to start, and deciding exactly how to achieve the more promising claims can be a daunting task. And it is not a task that can be undertaken easily, for BPM products must interface with legacy systems as they cannot replace them in their entirety in a single hit. Enterprise integration is one of the keys to success with BPM, and occasionally the most expensive hurdle.

To achieve the more profound goals of BPM, considerable investment and careful thought must take place to define the process whereby BPM is brought into play, and with it, a shift in the ways of doing business and developing software. Defining, programming and re-engineering business processes are roles more slanted towards business analysts than software developers. Software developer roles move toward enterprise-wide definition and exposure of services, providing fault tolerant interfaces to legacy systems so the analysts can more easily adapt to their role as process programmers.

Although a daunting task, the benefits of a truly process managed enterprise are potentially huge in cost savings and uniformity of output, freeing up workers to deliver an excellent service to customers.

Why BPM?

Once the future was a shining land where automatons would do all our work for us. We would be free to pursue our leisures, and create the utopian dream that was forever promised.

Now we have so much work to do we cannot squeeze enough into our days. We must get more worker-time through efficiency drives, training, outsourcing or automation.

BPM can deliver in all four of these areas. It will make workers more efficient by handling more of the repetitive, calculative and re-keying tasks a worker is likely to confront, leaving

them free to concentrate their efforts on the complex or abstract features of a case requiring human intervention. When a computer takes care of much of the work, its complexities are encoded and hidden from the worker, thus training time is reduced. By automating tasks, BPM tools effectively outsource the work to computer rather than to an offshore company.

The benefits of automation

Automation promises us many great things that people are not very good at. When asked to do the same task repeatedly, the human mind will introduce variety where none is required. We can get sums wrong, we can read and write numbers and names wrong. We can forget, ignore, put off, dream and chat when we have more important callings.

Little change will come from focusing on the variety; it has been necessary in our ability to adapt and evolve; it is part of our nature. We are not good at repeating things. Machines, on the other hand, can do little else.

Once people are freed of their monotonies, they can begin to interact with other people in a meaningful, value adding way, rather than hurrying through their list of things to do under the watchful eye of efficiency.

The possible benefits of automation are as follows:

- Unburden your work force
- Improve compliance
- Improve governance
- Increase auditability
- Increase accuracy
- Increase repeatability
- Increase processing speed
- Reduce the learning curve
- Supports initiatives for customer satisfaction
- Reduce employee turnover
- Reduce training where processes are automated
- Become a more flexible business

The carrot of automation is that it can help with many aspects of a business. The stick is that if you do not automate, business will become ever more complex and more costly.

Why we need automation

When our original IT systems were written, they met most of the needs for which they were created. Then the needs grew and changed. We redeveloped our IT systems, but they fell ever more short of goal, and more worker time, knowledge and effort was required to fill the gap between fast growing business requirements, and slower growing system abilities.

More people doing more work under more pressure results in more difficulty, more mistakes and more layers of supervision.

The obvious solution is to write newer applications to better meet the new requirements. The hope is to plug the gap; the reality is that more people are needed to write and support more systems. Such thinking leads to an explosion of systems and support staff.

The problem is expounded by acquisitive companies. They have bought up other companies' system explosions and need to reduce the amalgamated burden.

Four factors determine how many workers an organization must employ to perform a task:

1. The number of people required to do the work
2. The number of supervisors/trainers required to support the workers
3. The number of people required to support the systems
4. The amount of work the systems can remove from the people

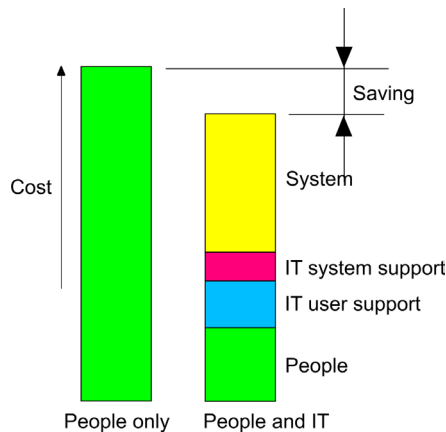


Figure 1. Static model of business costs and savings over manual processes

Beyond the static model, we see the effects of ever increasing requirements, from customers, auditors, customs & excise, shareholders, standards authorities and governance measures.

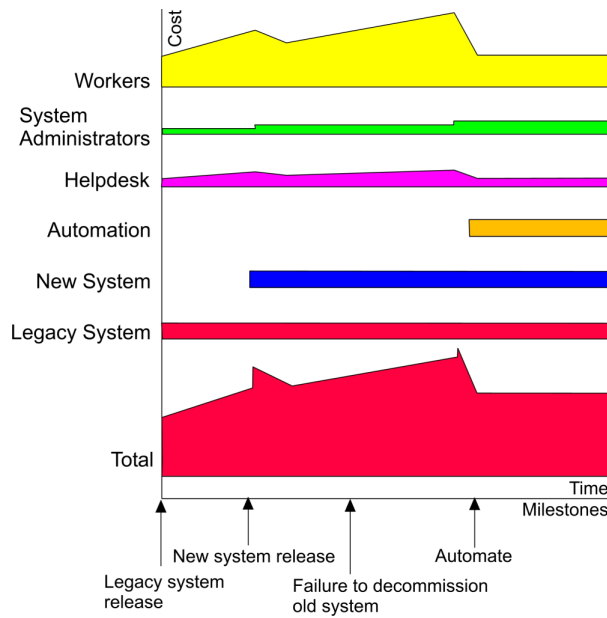


Figure 2. Dynamic model of costs as new systems are introduced

The scenario represented by figure 2 is as follows:

Long ago, a system was developed to support a business process. Over the years, the increasing requirements of the business were filled partially by redeveloping the software and partially by hiring manpower to manually do what the system could not. A new system intended to replace the old system was introduced. It was far more capable than the old system, and all new work was moved to it. Immediately, worker numbers were reduced and helpdesk numbers fell proportionately. The savings were partially offset by the new system administrators and training required on the new system, but once it was up and running, the cost savings would be fully realised when the old system was decommissioned. But decommissioning is often an extremely difficult thing to do. In the end, the old system was left running to handle the cases which remained on it. As with the old system, the new one gradually fell ever shorter of the requirements placed upon it. Workers increased.

The decision was made to automate the process. A single process running in a BPM environment would do most of the work and communicate with the newer legacy system where all data would be maintained. More system administrators were required, but less workers and support staff when the automated system went live.

This is an illustrative example of what might happen. Let's take a look at a real case, reported in Insurance & Technology News[2] about the automation of new policies at US Insurer North Carolina Farm Bureau using Clear Technology's Tranzax.

NCFB New Policies

As a result, "Our policy turnaround has been reduced from three weeks to five days maximum and most of the time it takes only two days," relates Squires. "Our quality has improved dramatically and it's almost error-free. Our once-manual rating methods are more consistent and we were able to re-deploy 30 percent of our staff to other areas of the business." Additionally, the two month training process that was necessary for employees involved in the old underwriting/policy processing routine, has now been reduced to two days.

A benefit summary for process automation

- Automation can help workers do more work more easily.
- Automation tools can fill the gap between the workers' needs and the legacy systems' abilities more easily than further developing those systems.
- Automation provides workflow across systems, removing informal people-driven processes and presenting a single user interface.
- Automating business processes makes things happen in a more controlled and measurable way.

To understand how automation is achieved, we must first consider the terms being used by the vendors of BPM software, and examine how people do their work.

Business Processes Management Terms

This section defines and highlights the relationships between terms.

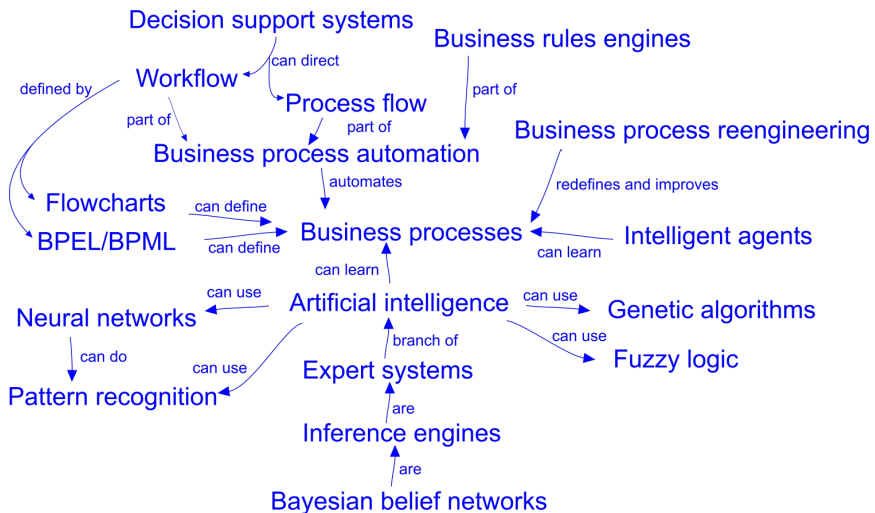


Figure 3. The relationships between some BPM terms

BUSINESS PROCESS MANAGEMENT IMPLEMENTATION

Artificial Intelligence	Machine based human-like reasoning
Bayesian Belief Networks	Reasoning in uncertainty without enough information to make a purely logical decision.
BPEL (Business Process Execution Language)	An XML based language used to define and execute steps in a predefined process
BPML (Business Process Modeling Language)	An extension of Unified Modeling Language (UML) specifically for modeling business processes
Business process	The series of steps and associated rules followed to achieve a desired outcome in a business setting, often represented by a flowchart
Business Process Automation	A subset of BPM concerned with the modeling and automating individual processes.
Business Process Management (BPM)	Monitoring, measuring, controlling and optimizing business activities using automation technologies
Business Process Reengineering	Redefining a process for better compliance, faster speed of execution, easier management or cost savings
Business Rules Engine (or Rules Engine)	A large set of if-then-else statements using weighting mechanisms, iteration and recursion to make business decisions
Decision Support Systems	The use of decision trees to calculate probabilities of success and costs of proposed actions
Expert Systems	The use of rules and data to formulate an opinion
Flowchart	A graphical representation of a process
Fuzzy Logic	An extension of Boolean logic to account for partial truths
Genetic Algorithm	An algorithm that applies breeding and mutation to converge upon a solution
Inference Engine	Uses a large knowledge base to deduce results through inference
Intelligent Agents	Software that carries out tasks without supervision, applying some intelligence and learning.
Neural Network	Real or virtual devices, inspired by neural structures, in which several interconnected elements process information simultaneously, adapting and learning from past patterns
Orchestration	Automated processes are triggered events or started at predefined times. Orchestration is used to describe the tightly or loosely coupled mechanism which coordinates this across the enterprise or in a particular layer
Pattern Recognition	The basis of all recognition (voice, handwriting, face etc). Patterns are identified in streams of data and matched against those on record or learned and recognized by neural nets.
Process Flow	The detailed flow of a business activity, e.g. take sales order
Workflow	The ability to control work at a macro level, and flow it between people or organizations

Related terms

BPM is often so dependent upon other activities (e.g. integration with legacy systems) that many related terms from technological neighbours have worked their way into its lexicon.

Business Activity Monitoring	Monitoring the business activities of individuals and groups
Service Oriented Architecture	An architecture in which applications are exposed as sets of services
Web Service	A software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards
Enterprise Application Integration (EAI)	It is difficult to automate without some form of enterprise wide integration. EAI can provide that integration, often using a web services approach combined with an integration hub or bus

How People Work

The work people do can be thought of as the application of knowledge to a particular business context, following a predefined sequence of tasks contained within a workflow. To be processed in a timely manner, the workflow must be contained within a transaction, and the transaction requires messaging across a network (of people) to function.

In terms of a stack akin to the ISO 7 layer networking model, we have:

Knowledge	The knowledge I need to do my job
Context	The context I am working in, e.g. adding a customer, or processing a customer's insurance claim
Sequence	The processes laid down by my employer for doing my job
Workflow	The passing in to and out of my area of responsibility
Transactions	The wrapping of workflows into transactions that have defined start and commit points, and rollback possibilities.
Messaging	The means of coordinating the transactions
Network	The medium through which the messages move

Table 2. The work stack

We are not software structures

We work concurrently. We collaborate, sharing unstructured and structured information. We do not always follow the predefined logic, often doing things differently or in a different order to those around us.

Software, on the other hand, is a sequence of hard coded steps based on Boolean logic. We have traditionally programmed our processes into our software, each one at a time, which is why our costs are high, and changes difficult.

BPM software attempts to enable changes in process to keep track with changes in the business environment. This is the fundamental difference between process software and all other software. We have shifted a level of abstraction above program, to process. In doing so, writing software becomes a process configuration exercise, more uniform, and easier to rework.

Thus, BPM attempts to remove software production from the critical path of business change, while reducing costs.

BPM Products

To truly replicate the roles people have in business, an automation tool must be a master of the top 5 layers of the work stack in table 2. Whether it also provides all of the nuances of other programming languages depends upon the product, its intended uses and its maturity.

A BPM product is a programming tool created to model and run business processes. The programming interface may be flowcharts, but the principles of software engineering,

such as program construction and error handling, are as important as with code based programs. This can be overlooked in the analyst as programmer approach to process automation.

Modern programming languages provide much more than the basic constructs. A reasonable list might be sequence, conditional branching, structured loops, concurrent threads, inter-thread communication and synchronization, instance initialization, manipulation of variables and data types, throwing and catching exceptions, waiting on a lock and resuming afterwards, testing a predicate on several fields, logical and math operations, subprogram calls, and assigning and freeing up storage, software and hardware resources.

The BPM marketplace

The number of BPM products is mushrooming, and consolidating only slightly at the same time. Tibco has bought Staffware, and Oracle has bought Collaxa, a BPEL engine. In the BPM space, the BPMG (The Business Process Management group, www.bpmg.org) currently lists over 400 vendors, including document management specialists.

Two industry bodies support BPM. They are the BPMG, mentioned above, and the BPMI (The Business Process Management Initiative - www.bpmi.org). The BPMI is responsible for many BPM standards including Business Process Modelling Language (BPML). Some vendors use BPML in their products, but it is far from universal. Vendors BEA, IBM and Microsoft have proposed BPEL (ex BPEL4WS), a similar but more restricted notation used by some integration and automation products.

The so-called *pure play* vendors are under pressure from products reaching in from established operators in the related fields of document management, enterprise application integration (EAI), and from ERP leaders.

Many vendors claim to be the leader in the field, but this is confused by a number of factors, including exactly how the market is defined, and without anyone publishing the indicators by which they claim to lead.

Gartner, Forrester and other analysts continue to evaluate the BPM market, and their analyses show up a number of interesting facts about the BPM market:

- The market is not consolidating
- No-one has emerged as a dominant force in the past two or three years
- A great many challengers exist
- The leading analysts do not wholeheartedly agree about what makes a BPM leader

So what is a BPM product?

BPM offerings are part programming language, part operating environment and part person replacement. Despite the array of advanced technologies available, those that employ

artificial intelligence are highly specialized niche products. The leaders have products based on workflow, rules engines and process flow.

Some automation products use Java or a bespoke language to create their process flows. Others use graphical techniques or flowcharts so that analysts can model rather than having coders code.

There are two major features to a BPM product: the design time, and the run time environments.

The Design time environment

In the design time environment, a designer must be able to specify:

- Business Processes
- Business Rules
- Workflow
- Presentation (to user)
- Data
- Points of integration
- Hardware and software configuration
- Roles and related activities
- Management Information (MI) & activity monitoring

The Runtime environment

At runtime, the execution engine schedules and orchestrates workflow and processes. A mechanism for viewing the activity of the engine must be available for monitoring and support.

Workflow management, depending upon its sophistication, allows managers to view the current and historic load on any worker, and reassign work from one worker to another.

Dependencies

A BPM solution is usually dependent on external systems for integration with legacy systems and access security.

Comparing and Selecting a BPM product

To compare BPM products is an extremely difficult task due to unfamiliarity with the medium, the vendors, and the wealth of products all appearing to do the same thing.

Here is a list of questions that might help:

- How are the business processes configured and by whom?
- How does it integrate with the systems it is automating?
- What does the user see (if anything)?
- What is available to the manager or supervisor?
- What is available to support personnel?
- Who else like us uses your product?
- How are automation products delivered, reconfigured and updated?
- Can it really automate what our people do?
- Are the technologies and ways of creating BPM solutions compatible with our way of working?

There is, in addition, one consideration that must come before the automation project, which is:

- How do we realize the benefits of automation?

A \$1M automation solution can easily save \$10M per annum on paper, but how do you realize that benefit? It is rarely as simple as removing the people whose work has been replaced by machine.

The list above is a simplified set of questions. For a more testing set, consider those in *The Emergence of Business Process Management*[3].

Automation Oriented Architecture

Two goals of BPM are to do the work that people normally do, and to enable the rapid change of business processes in an agile environment. The enterprise architecture must facilitate these goals.

In this section, we look at an enterprise architecture designed to support BPM, under the name *Automation Oriented Architecture*. It is a layered construct with three areas of concern, where each area is the focus of a different field of expertise.

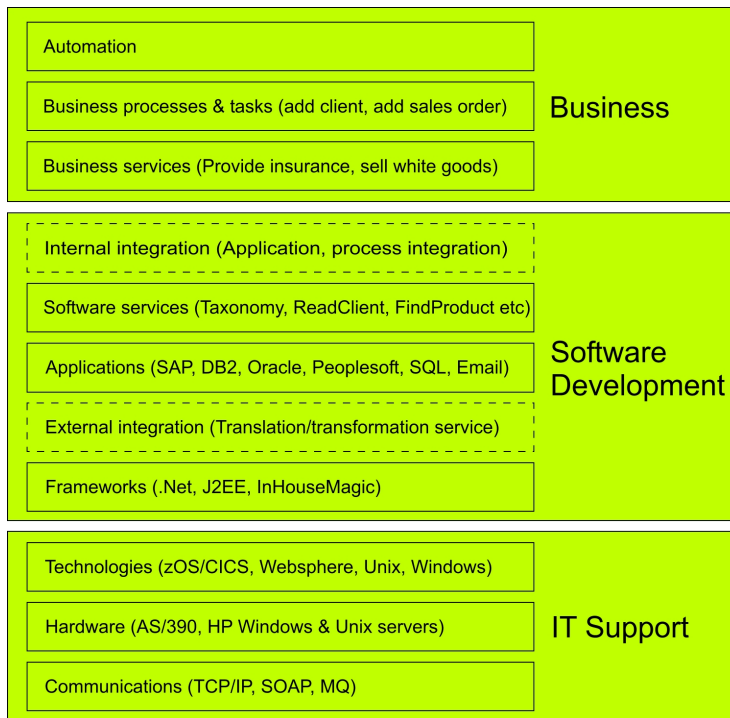


Figure 4. AOA - The big picture

The 11 layers of AOA

Business focused AOA - Automation, business processes and business services

In the business layers, the workers, supervisors and business analysts are concerned with the services the business provides, and how those services are delivered by processes and workflows. The analyst further defines this into how the flow might be automated and orchestrated.

Automation

Here sits the automaton (human, software or both) running through a defined business process.

When developing automation functions, it is important from a business point of view to focus on the benefits derived (e.g. faster processing, earlier income, less manual error) and balance the benefits against the cost of development and implementation.

There is a great temptation when creating automation solutions to re-engineer the business processes at the same time. Doing so makes it far more difficult, and should be

approached as a light exercise focusing on re-engineering to enable or simplify automation. A full re-engineering effort followed by automation will result in statements like "*if we knew that beforehand, we wouldn't have re-engineered it that way.*"

Business Process

Something defined on paper, in software, or mutually agreed as a way of doing things. In this context, processing comprises workflow, work procedures, business rules, data definitions and presentation of information to workers and their supervisors.

Business Service

Services are the core activities of the business. They are the corporate raison d'etre, and held as tasks and processes on paper, in software, and in the hive mind. The business leaders and BPM champions will focus on the business services to identify what will be automated and when, and what the likely benefits will be.

Development focused AOA - Integration, software services, applications and frameworks

The role of development in automation solutions is to provide the environment in which the analyst can automate. Developers are not concerned with flows, user interfaces or business logic, but with providing interfaces between the automation software and the legacy or back office systems.

Integration

An enterprise automating manual and paper based tasks will be relatively rare. In most cases, some form of integration to legacy systems is required.

If integration is being built as part of the BPM project, it will prove challenging as both are complex undertakings. Integration is difficult without expert supervision, for it must address many complexities, such as:

- A method for defining and running cross systems processes (Business Process Engine, using e.g. BPEL)
- A single generalized view of data held by the enterprise (Canonical or common data model), and the ability to transform the data from one system into something acceptable to another
- Code mapping between systems (New York on one system is NY on another, and 126 on another)
- Key mapping between systems (Person ABCD-1234-AB12AB12AB12 in system A is Bob in system B and bob@whoknowswhat.com in another)
- Enterprise level locking
- Agnostic routing (i.e. I have a new order, rather than System X: I have a new order)

- Synchronous or asynchronous message routing/handling
- The ability to translate or delegate translation of protocols and languages
- Audit trails
- Mechanisms for unblocking pipes and rerouting stuck messages
- The ability to persist state across systems
- Data synchronization services
- Exception handling
- Process restart (with multiple restart points)
- An address book of all interconnected systems
- Authentication

Whether a hub or some other mechanism is used, integration will supply services internal to the organization, and also allow communication with the outside world.

Internal integration allows an automator to work on a task regardless of where or how the data representing it is held. It covers some or all of the integration items listed above.

External integration boils down to a four function subset of internal integration, providing addressing, authentication, translation and transformation. Incoming message structures rarely match the internal business representation. Transformation and translation services convert an external format into an internal format, and vice-versa.

Software services

These are the services of *Service Oriented Architecture*. Each application is exposed to other applications as sets of services. A service typically allows any application to read and write blocks of data and exposes applications' internal functions. Within the service, business rules may be applied to the data, which may, in turn, be supplied by another service. Most services are of the form *create, read, update, delete, search* or *read and lock* any item of business relevance. For example, *create sales order, delete customer* or *read report X12* are services.

In a pure service oriented architecture, everything is a service. A service may be required to discover which services are to be used within a given process. Imagine the following scenario:

A new request for an insurance quote comes in. The vehicle is a Volvo FH16. A human operator might pull out a Volvo brochure, or look in some lists or on the internet to find out that this particular model is a 6x2 axled artic, and needs to be processed by the team at Leeds on the commercial vehicles system.

An automated system also needs to look up this information and would achieve it best by calling a taxonomy service.

A taxonomy is a system of classification, and would classify the processing mechanism and computer system for each type of vehicle, and also the classification for each particular model.

Applications

Applications are anything installed on desktops and servers such as analysis tools, financial systems, ERP systems, search engines, web servers, databases, email servers etc.

To make use of these applications, they must be exposed to outside use. The most common form of exposure is through software services or web services. Other forms of exposure are published APIs or screen scraping for mainframe applications.

Frameworks

The frameworks provide shared services to applications. Reuse should occur at this low level, rather than expecting it to occur at the application level.

There are many frameworks available these days, providing abstractions and services to what lies beneath. Some come with purchased applications, other are .Net, J2EE, MFC, and vendor specific offerings. Many internal software development teams have created their own frameworks.

IT Support focused AOA - Technologies, Hardware and Communications

Once automation solutions have been created, they need looking after. Each technology and piece of hardware utilised must be carefully monitored to maintain the flow. Broken flow will create backlogs and blockages, the freeing of which IT support must expedite.

Technologies

Operating systems, controlled environments such as Javabeans, CICS, COM+, messaging (eg Websphere MQ).

Hardware

Servers, networks devices, wires, radio links, laptops, phones, PDAs, routers, switches, firewalls, load balancers etc.

Communications

Many modern systems talk TCP/IP. Some use http, others such as the UK national lottery use X.25. In the murky world of aged legacy systems RS-232 may be the only option.

AOA and business

AOA Management, Monitoring, and Management Information

Each layer in the AOA architecture - and in the business enterprise - needs different management and monitoring. This section lists how each layer is managed, monitored, and what management information (MI) it will provide.

Layer	Management	Monitor	MI
Automation	Who gets access?	Are services running? What is blocked?	What ran when, what was the outcome?
Business processes	Do it this way.	What's going on? Who is doing it? How long is it taking?	How many did we do this month?
Business services	This is what we do	Which are active, passive? What is taking the most effort?	What provides our income, profit?
Internal integration	What talks to what	What is talking?	What talked?
Software services	What can call what	What gets called, what are response speeds?	Service hits, exceptions
Applications	What we use to do business	Ensure the applications are available	Throughputs, licences, applications
External integration	Who we talk to. How we talk.	Who is talking to who? Is bandwidth adequate?	Who talked to who, in what volumes?
Frameworks	What we use to develop applications	Reuse, compliance	Reuse audits of centralized services
Technologies	What OS, and messaging	Keep them up & patched	Up times
Hardware	What boxes we use	Keep them up	Up times
Comms	Firewalls, spam filters, protocols	Traffic peak times, bandwidth usage	Traffic analysis

Table 3. Management, monitoring and MI in the AOA layers

AOA and Governance

In 2000, according to McKinsey[4], investors would spend 18-28% more on a company with governance measures. The reason for this is the high level of expensive corporate failures. Governance measures mean less investment risk.

Hamaker's[5] definition of governance highlights three areas of focus as follow:

- 1 *Corporate Governance* balances the power of the CEO with the board's role as custodians of the enterprise.
- 2 *Enterprise Governance* responsibly controls the journey in the strategic direction and the day to day activities.
- 3 *IT Governance* ensures IT sustains and extends the organizations strategies and objectives.

Within AOA, each layer addresses a different aspect of governance, as detailed in table 4.

Layer	Area	Measure
Automation	Enterprise	What and how to automate
Business processes	Enterprise	Definition of how to do work
Business services	Enterprise	Areas of work
Internal integration	IT	Single representation of business in software, regardless of systems used
Software services	IT	What services support what business processes
Applications	IT	How we maintain and support what we use to do business
External integration	IT	How and when we talk to our collaborators
Frameworks	IT	How we promote reuse
Technologies	IT	What we use to make our software environments
Hardware	IT	How we maintain our hardware
Comms	IT	How we let communication in and out

Table 4. Governance and the AOA layers

Patterns and Automation Oriented Architecture

The 11 AOA layers are an abstraction across a business, through the layers of technology and into other systems. Some of the aspects of these layers are reflected in our software design patterns[6]. A design pattern is a simple, elegant and recorded solution to a specific problem in software design.

Client Server, Multi-tier

The 11 layers are not a stack, but rather 11 points of focus for an automated enterprise. As a stack, it would look something like this:

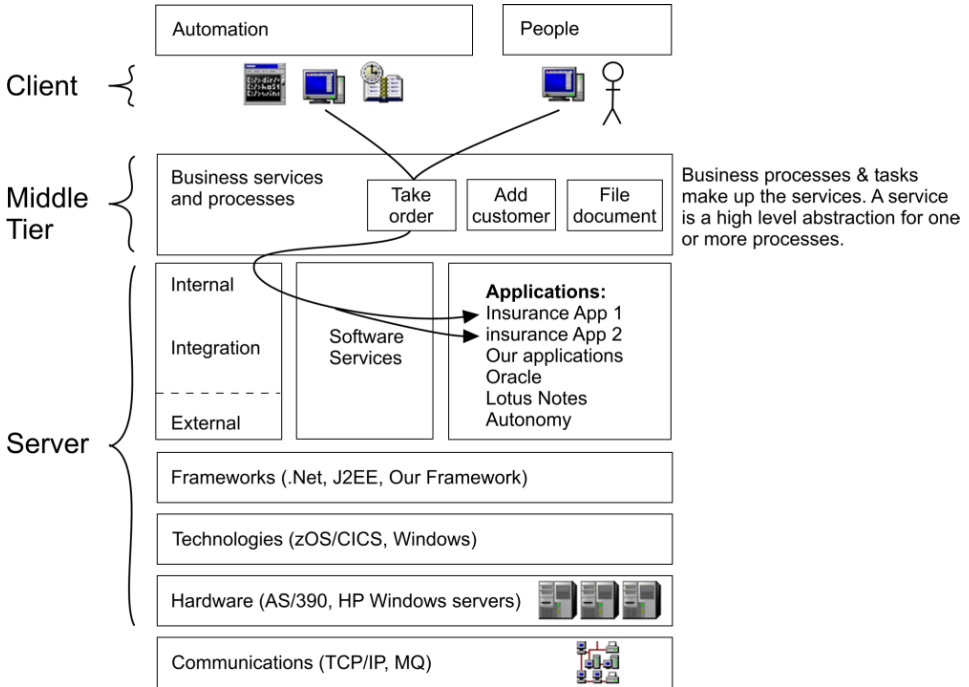


Figure 5. AOA as a stack

Model-View-Controller (MVC)

Model-View-Controller is one of the best known design patterns. The model is the object or data being worked upon, the view is the on-screen presentation of that data or object, and the controller defines the way the user interface reacts to user input to change the data.

In this space, MVC becomes an excellent discussion leader. Most important is: where does your model live?

Across an integrated space, the *model* may exist as a canonical data definition within the integration layer. It may be an ontological or taxonomical element contained elsewhere. A single model needs abstracted or transformed to be able to be persisted in a back end system and operated upon by an automator.

There are many *views* of an automated process. Is it running? What is currently running? Or, which step is the process in? Or, show me the audit trail. The most obvious view of a process is the design time view, often resembling a flowchart.

And what of the *controller*? Is your automation layer the overall controller, or are you passing responsibility off to another service? How is control shared between the automation software and the integration software?

Façade

If you are integrating a service or process contained in alternate versions on more than one back end system, and you want to build a single process across them, you will need a façade.

Bridge

You will almost always need a bridge to link to the back end system. It may be an established technology such as web services, HAS (ex-SNA), or a service provided by a screen scraping tool.

Chain of Responsibility

In a chain of responsibility made up of workers, a piece of work will pass from worker to supervisor to manager or to specialists until it reaches a person with the knowledge and authority to do it. Often these chains are defined within each process as a workflow, but there may be some benefit to decoupling the workflow from the process.

Publish and Subscribe

Within the integration layer, a subscription service may accept publications from applications, and send out an event to a list of subscribers. All event driven architectures are based on publish and subscribe.

Comparison with Service Oriented Architecture

Service Oriented Architecture (SOA) is a loosely coupled, distributed, coarse-grained collection of easily callable services designed to integrate both new and legacy systems, as well as being more flexible in adapting to future changes. No standard definition exists for it, but it can be thought of as a four layer mechanism comprising the business process layer, the business services layer, the application layer and the technology layer.

AOA is derived from SOA, where the automation layer sits atop the SOA layers, integration is inserted into the layers, and technology is expanded out to address the management and governance issues where automation can contribute.

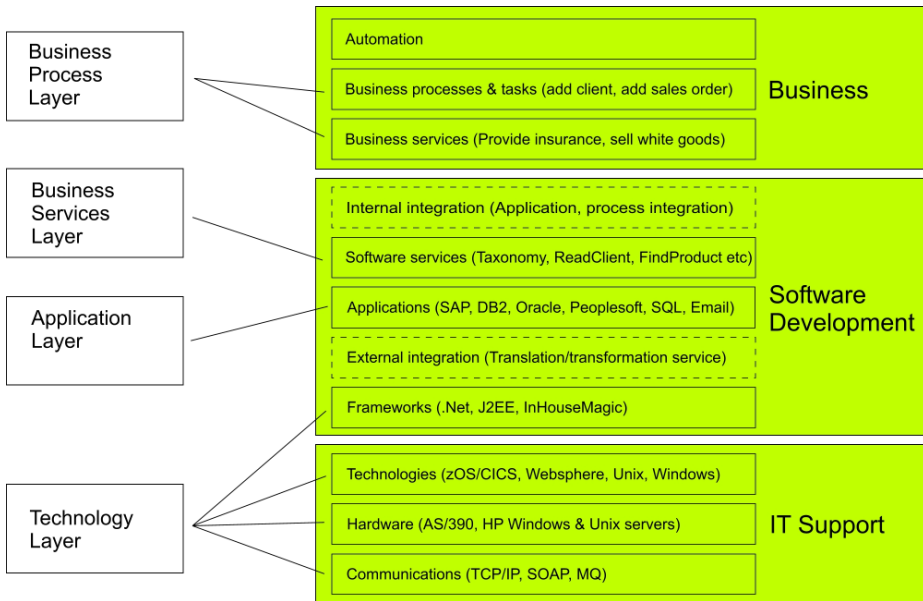


Figure 6. SOA and AOA compared as layered constructs

One important point concerns the use of the word service. We are working in a service based economy according to politicians and financiers. These services are the business services of the third layer and are delivered using the processes defined in the second layer. SOA uses the term business services to mean the software services that are granular access to the applications. This is confusing, and AOA differentiates them as software services.

Comparison with Event Driven Architecture (EDA)

An event driven architecture supports applications raising and responding to events.

As an example, a service oriented architecture would allow an application to call a service to see if the fire alarm was ringing. In event driven architecture, the alarm would raise an event to the integration layer as a publisher. The integration layer would then route that event to a number of subscribers.

The service oriented and event driven approaches are compatible, and automation oriented architecture is a fusion and expansion of the two, allowing for events and services to coexist. In AOA, the event mechanism is contained within the integration layer in a *publish and subscribe* mechanism.

In some cases, automation by polling queues and calling lists can be extremely inefficient, producing high network traffic of little value. In such cases, an event driven approach reduces traffic, while improving responsiveness and overall throughput.

Common Pitfalls

Terry Harvey [7] noted a number of milestones on the road to the successful adoption of BPM. Some additional points of interest are noted where hindsight may benefit other travellers.

Creating a cross-functional team

BPM requires dedicated resource from legacy teams, integration and the process automation team. To try and deliver by allocating resource as it is required will stretch project schedules.

Integration

Integrating with legacy systems is hard. Getting integration resources when required, in organizations with mainframe development needs, is never easy.

Integration usually comes via a service based offering. If there isn't already a service based approach, it will take more effort.

Testing integrated systems requires a well defined process to effectively test all parts together. Bugs in loosely connected systems are particularly difficult to track down, resulting in long testing cycles.

The tools

The tools are maturing, but don't expect to see the polish you see in most version 10+ packaged applications.

Tools on the integration side are often promise-ware and rarely work in your environment. The reality is that you often have to roll your own.

Re-engineering

Don't do it. If you do it, you will just delay your automation project, and once you find out how it will get automated, you will probably decide you would have automated it differently if you had known. The other reason for not re-engineering, is that once automated, a lot of the problems may disappear in layers of technology.

The best re-engineering is what is obvious to make the digitization process easier.

Measuring the benefits

Many legacy systems prove rather difficult to extract sufficient information from to give a good picture of processing costs. This means that benefits will be hard to estimate.

Delivering the benefits

The hardest thing in having measured the benefits, is how to deliver them. If you automate what 50 people do, can you lose those people?

Desire to do everything at once

As with all software projects, it is important to build small and deliver often. Business processes do not stand still, so a lengthy project will require many process reviews, building inefficiency into delivery.

Big data

BPM tools can (and often do) record everything. It is possible to audit the exact trail through a process of any individual case. To do so means an enormous amount of data, perhaps tens of gigabytes per automated process per year. The erasure of this data must be balanced against audit requirements.

Just another software tool

BPM tools can deliver software just as easily (or with similar hardship) to traditional development tools. The difference in BPM is that it is not just about software development, it is about a process managed enterprise. Sales haste can promote the former at the expense of the latter.

A nice front end

A great selling point of BPM tools is that a process integrated with many backend systems appears to the user as a seamless whole. There is no more clicking from one screen to another, one application to another to process an item of work.

The work in achieving this seamless whole is no less than with traditional coding tools.

Verbosity, distributed code and debugging

A fully automated business process will require many disconnected pieces of code to function. In the BPM runtime, the encoded flowcharts will communicate with compiled code. In the integration space XSL transforms will act upon XML data under the influence of BPEL scripts. In the legacy systems, layers of web and mainframe technologies will provide mainframe services. On one system, an event being raised on a mainframe had to pass through 27 connection points (i.e. pieces of code) to arrive in the BPM execution engine. When we went into pre-production, one bug took three days to narrow down to a security setting on the server. That time was spent wading through excruciating amounts of transforms and BPEL. BPEL can be extraordinarily verbose.

Methodologies

Software development methodologies go some way towards helping to deliver BPM projects, but a slightly different approach is required. In fact, four separate methodologies are required.

- 1 *Methodology 1.* Identification methodology for use by sponsors to uncover the most lucrative areas of automation.
- 2 *Methodology 2.* Process automation methodology for analysts creating processes in flowchart-like programming environments. This must include many traditional programming complexities such as patterns for integration and error handling.
- 3 *Methodology 3.* Traditional software development for the services consumed by the automaton.
- 4 *Methodology 4.* Creating an integration layer requires a methodology specific to integration.

Will things ever be the same again?

BPM requires serious enterprise level commitment. In return, a process managed enterprise is a controlled, auditable, regulation compliant and efficient machine.

BPM is not about the implementation of a new piece of software, but a realignment and fusion of business and IT, requiring a change in the way a business is organized, and in the way it produces and supports its software.

In BPM: The Third Wave[1], the authors claim BPM is the foundation of business thinking for the next fifty years. It is looking ever more likely. It is difficult to see how any service based business using manpower over computer power will be able to compete in an increasingly global, information hungry, always on, market.

References

- [1] Business Process Management: The Third Wave - Howard Smith and Peteringar - Meghan-Kiffer Press, 2003.
- [2] Insurance & Technology News, 2004. <http://www.insurancetech.com/story/news/?articleID=20800096>
- [3] The Emergence of Business Process Management - CSC, 2002.
- [4] Three surveys on corporate governance - Paul Coombes & Mark Watson -McKinsey Quarterly, 2000.
- [5] Spotlight on governance - Stacey Hamaker - Information Systems Journal, Vol 1, 2003.
- [6] Design Patterns - Gamma, Helm, Johnson, Vlissides - Addison-Wesley, 1995.
- [7] Fewer People, Improved Productivity, Thanks to Business Process Management -Terry Harvey, Colonial Penn, 2004.