Automation Oriented Architecture

Part 6 of an 8 part paper on Business Process Management (BPM)

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 IT architecture must facilitate these goals.

In this section, we look at such an architecture, under the name of *Automation Oriented Architecture*. It is a layered construct with three groups, where each group is the focus of a separate area of business.

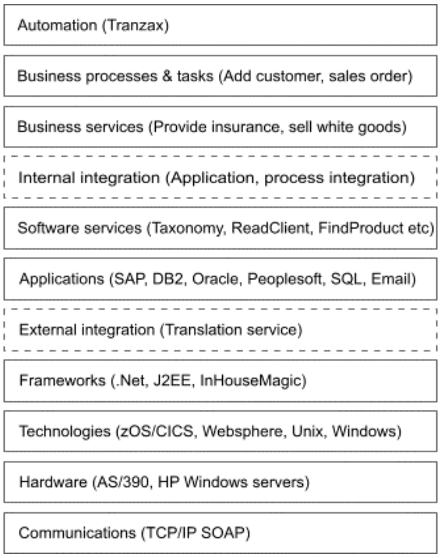
Automation (Tranzax)
Business processes & tasks (Add customer, sales order)
Business services (Provide insurance, sell white goods)
Internal integration (Application, process integration)
Software services (Taxonomy, ReadClient, FindProduct etc)
Applications (SAP, DB2, Oracle, Peoplesoft, SQL, Email)
External integration (Translation service)
Frameworks (.Net, J2EE, InHouseMagic)
Technologies (zOS/CICS, Websphere, Unix, Windows)
Hardware (AS/390, HP Windows servers)
Communications (TCP/IP SOAP)

Figure 1. The 11 layers of AOA

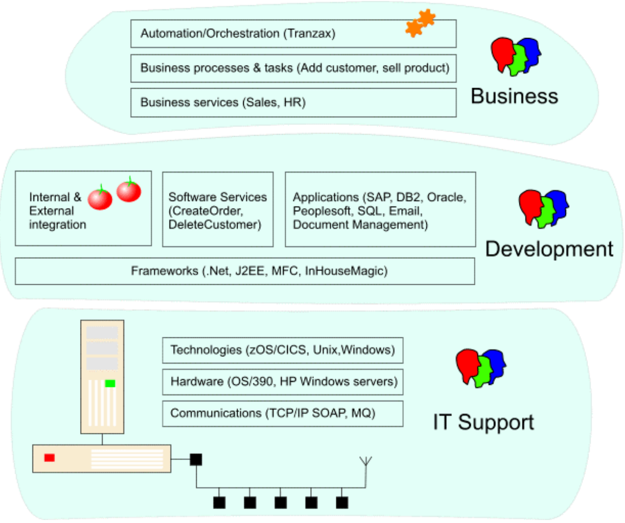
Automation/Orchestration (Tranzax)
Business processes & tasks (Add customer, sell product)
Business services (Sales, HR)
Business
Internal & External integration
Software Services (CreateOrder, DeleteCustomer)
Applications (SAP, DB2, Oracle, Peoplesoft, SQL, Email, Document Management)
Frameworks (.Net, J2EE, MFC, InHouseMagic)
Development
Technologies (zOS/CICS, Unix,Windows)
Hardware (OS/390, HP Windows servers)
Communications (TCP/IP SOAP, MQ)
IT Support

Figure 2. AOA – The big picture

The 11 layers of AOA

Business focused AOA – Automation, business processes and business services

In the business layers, the business analyst is concerned with the services the business provides, how those services are achieved by process and workflows, and 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 reengineering effort followed by automation will result in statements like *"if we knew that beforehand, we wouldn't have reengineered 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.

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.

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 backoffice systems.

Integration

Everyone seems to want more interconnected computing. From the single user on the internet, through businesses wanting less systems, less duplication of data and function, and easier access to dispersed information, and to governments and law enforcement agencies wanting worldwide instant information collation and retrieval.

The most common solution to the problem of integration is to put in an integration hub. It is a beguiling concept, one that anyone with experience of a bicycle wheel can get right away.

"Let's put in a hub to get our systems talking to each other," is ringing through the offices of the corporate world in the hope that this magical piece of lego will suddenly transform the way things work. In reality, an integration hub is not an easy project to tackle for two reasons. First, it's hard to understand and do. Second, tackling such a project requires experience, and the organizations wanting to do it don't have integrated systems, therefore they do not have the experience.

An integration hub must tackle the following:

- 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 allows communication with the outside world.

Internal

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

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

The main applications used by an enterprise are becoming ever less varied. The great powerhouses of the past are being amalgamated through judicious imitation or outright purchase. Some industries have escaped these troubles, and stuck with their 30 year old mainframe systems. Regardless, they all use 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 in the case of old mainframe applications, screen scraping.

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 used 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, runtime 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, calculations
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

AOA and Governance

In 2000, according to McKinsey[1], 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[2] definitions of governance highlights three areas of focus as follow:

Corporate Governance balances the power of the CEO with the board's role as custodians of the enterprise.

Enterprise Governance responsibly controls the journey in the strategic direction and the day to day activities.

IT Governance ensures IT sustains and extends the organizations strategies and objectives.

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,

[1] Three surveys on corporate governance, Paul Coombes & Mark Watson. McKinsey Quarterly 2000.
[2] Spotlight on governance, Stacey Hamaker. Information Systems Journal, Vol 1, 2003.

Patterns and Automation Oriented Architecture

The 11 AOA layers are an abstraction across a business, through the layers of technology and into other systems.

This does not mean they represent other abstractions, and here we will investigate how some common patterns are reflected in the AOA layers.

Client Server, Multi-tier

The 11 layers are not a stack, but rather a choice of ways through the automation maze. As a stack, it would look something like this:

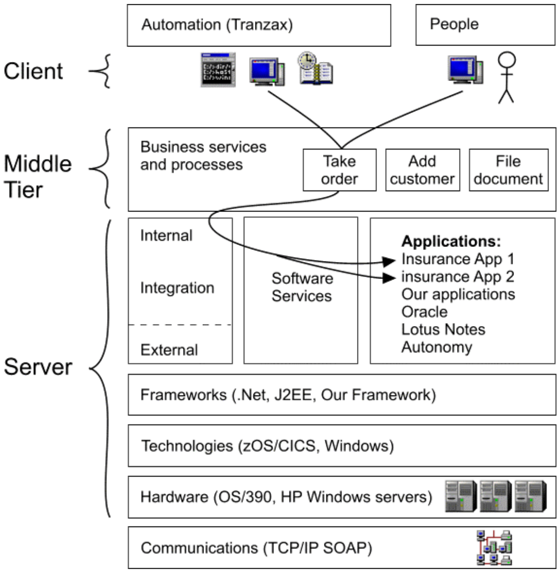

Figure 3. AOA as a stack

Business processes & tasks make up the services. A service is a higher level abstraction for one or more processes.

MVC

In this space, MVC becomes a good discussion point. 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.

There are many **views** of an automated process. Is it running? Another one is 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 split 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 fudge such as screen scraping.

Chain of Responsibility

All workflow systems are related to this. Typically in a chain of responsibility, a process will be passed from one object to another until it reaches one with the knowledge and/or authority level to do something with it. This is exactly what workflow does, except the objects receiving the processes are people.

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 (SOA)

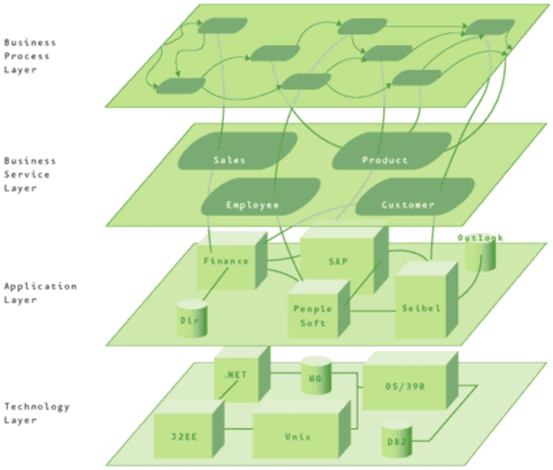

Figure 4. SOA in layers

AOA is obviously derived from SOA, where the automation layer sits atop the layers, integration is inserted into the layers, and technology is expanded out to address the management and governance problems inherent in the different AOA layers.

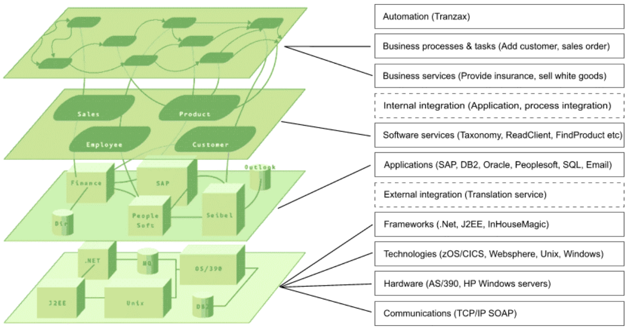
Sales
Product
Employee
Customer
Finance
SAP
Outlook
Dir
People Soft
Seibel
.NET
MQ
OS/390
J2EE
Unix
DB2
Automation (Tranzax)
Business processes & tasks (Add customer, sales order)
Business services (Provide insurance, sell white goods)
Internal integration (Application, process integration)
Software services (Taxonomy, ReadClient, FindProduct etc)
Applications (SAP, DB2, Oracle, Peoplesoft, SQL, Email)
External integration (Translation service)
Frameworks (.Net, J2EE, InHouseMagic)
Technologies (zOS/CICS, Websphere, Unix, Windows)
Hardware (AS/390, HP Windows servers)
Communications (TCP/IP SOAP)

Figure 5. 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 in 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 correctly labels them as software services to avoid this confusion.

Comparison with Event Driven Architecture (EDA)

An event driven architecture would support 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 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, event driven approaches reduce traffic, while improving responsiveness and overall throughput.

Other Papers in this set

Paper 1: What is business process management?
Paper 2: Why automate business processes?
Paper 3: Business process management terms
Paper 4: How people work
Paper 5: Business process management products
Paper 6: Automation Oriented Architecture
Paper 7: Case studies and common pitfalls
Paper 8. The future of BPM