





Table of Contents

Introduction: Business Modeling / Analysis for SOA	3
Business Promises of Service-Oriented Architecture	3
Business Agility & Integration	3
Service-Oriented Enterprise	3
SOA for Public Service	4
Principles and Challenges of SOA	4
SOA in a Nutshell	4
SOA Critical Success Factor	5
SOA Tenets	5
SOA Principles	6
SOA Business Analysis Expectations	6
Additional Expected SOA Capabilities	6
Externalization of Services	6
Outsourcing	6
Merger & Acquisitions	7
Enterprise Mash-up	7
SOA & Traditional Business Analysis	7
Artifacts of the Traditional Business Analysis	7
System Focus	8
Lack of Enterprise Vision	8
Quality and Completeness Challenge	8
Failure to “Encircle” Business & System Services	8
Temporary Solution – Business Analysis by Architects	8
Key Principles of SOA Business Analysis	9
Integration with Business Process Management	9
Visual Modeling	9
Iterative Modeling	10
Reference Models	10
Zachman Framework	10
Model Driven Architecture	10
Contextual Business Analysis	11
Business Context Analysis	11
Capability Modeling	11
Common Information Model	11
Business Services Definition	11
Interactive Collaboration Modeling	12
Future Usage Workshops	12
Additional Modeling Sources	12
Traditional BPM & Requirements Artifacts	12
Bottom-up Modeling	13
Practical Details	13
Selecting Modeling Software	13
Process Modeling	14
Hub-and-Spoke Modeling	14
Further Information and Legal Notes	14



Introduction: Business Modeling / Analysis for SOA

Abstraction Business Analysis for SOA is a response to the urgent need to get SOA from pilot stage and turn it into a mainstream IT practice. As this white paper shows, while SOA is in urgent need for compatible Business Analysis methodology, with traditional Business Analysis not been capable to deliver what SOA needs.

Abstraction Business Analysis for SOA is the methodology specially designed to deliver required information to SOA Architects and Integration and System Engineers, as well as to facilitate the realization of SOA advantages. The methodology is based on two sets of activities:

Contextual Business Analysis defines overall capabilities that the solution should deliver. It defines the overall Business Domain, captures mandated Business Processes, identifies solution-agnostic business rules and builds the Business Capability Model.

Business Services Definition is an iterative and interactive exercise that provides detailed definition of Business Services on a Conceptual Level. It is conducted with strong business involvement, and transfers sufficient knowledge to the business to enable the business to generate advantage out of SOA-enabled flexibility.

Business analysis for SOA is not possible without significant elevation of the role of Visual Modeling. While traditional Business Analysis views Visual Modeling as a graphical addition to specification documents, SOA Business Analysis is conducted within a complex Visual Modeling environment, with interactive workshops, documentation, QA, reporting and many other capabilities provided by Visual Modeling software. Business Abstraction's SOA Business Analysis is enabled by excellent ergonomics, scalability and extendibility of Sparx Systems Enterprise Architect in combination with the Business Abstraction extension to Enterprise Architect.

Business Promises of SOA

SOA is almost unanimously described as the most important development of the incoming decade. It is seen as a critical component for achieving business advantage.

Business Agility & Integration

SOA should enable an enterprise to dramatically reduce time-to-market for new products and services, especially in information-based industries.

It should also reduce the cost of integrating business functions, increase productivity of all operations related to information and make enterprise more transparent and manageable.

Service-Oriented Enterprise

The ultimate promise of SOA is a Service-Oriented Enterprise (SOE), an enterprise that can engage other parties through orchestration of services, as well as to provide externalized services as products.

At this stage Service-Oriented Enterprise remains a hot IT topic, while the leading business schools do not yet display comparable levels of enthusiasm. However, there are some early examples like Amazon and Google. Also, anecdotal evidence suggests that SOA has only just reached the level of maturity where business can become interested. However, as a first Service-Oriented Enterprise to enter a specific segment is likely to capture a significant chunk of the market and deliver a severe blow to the opposition, the fact that no competition has functioning SOE at that stage should not be seen as a sign of no urgency.



Multiple sources, including the Business Abstraction SOA Maturity Model, suggest that it will take several years for the business to develop understanding, culture and practices sufficient to transform the business into a SOE. In light of that, a SOE can be much closer to some companies than to the others.

SOA for Public Service

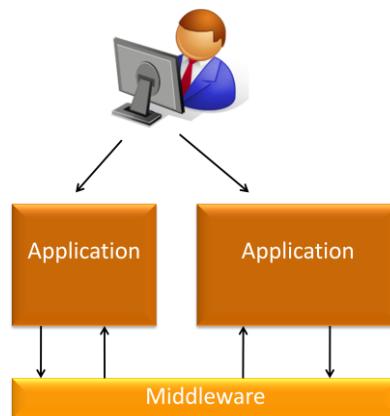
Although the concept of “competitive advantage” is not applicable to public service, the interest in SOA is rising, sometimes ahead of private enterprise. While for an enterprise the rate of change is a business decision, for a government agency the rate of change is dictated by a political process that is even less responsive to IT woes. Requested changes in the way public services are delivered often require that a significant transformation of Business Processes is delivered within a very short timeframe.

Principles and Challenges of SOA

The following describes key principles of Service-Oriented Architecture.

SOA in a Nutshell

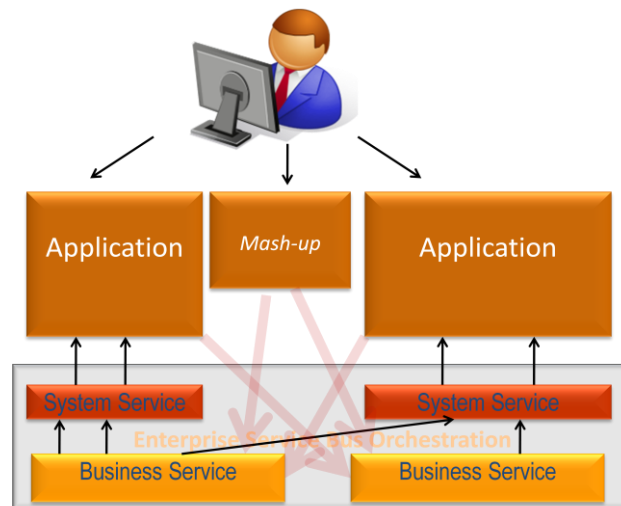
Traditional application architecture sees people – staff members, customers and other parties – working with individual applications. Applications can communicate directly, using middleware, or through access to common data resources.



Traditional Enterprise Integration

SOA uses Enterprise Service Bus (ESB) software to build two levels of service:

- System Services, or fine-grain services, which provide access to individual applications;
- Business Services, or coarse-grain services, which provide system-agnostic access to application capabilities, and should be described in business-centered terms.



SOA Integration

Business Services are implemented by orchestrating System Services. Every application uses Business Services only. High-level processes are implemented through orchestrating Business Services.

SOA Critical Success Factor

SOA is the 3rd generation of encapsulation-based paradigms, following the Object-Oriented (1st generation) and Component-Based (2nd generation) approaches. Both earlier encapsulation-based approaches proved to be highly beneficial when object/component definitions were artfully crafted, and hugely disappointing when the approach was implemented without any attention to paradigm shift or establishing a culture of continuous improvement of design skills.

It is logical to suggest that business-friendly and well-designed Business Services will deliver the agility that business needs. It is also logical to suggest that badly designed Business Services will deliver nothing but an extra layer of complexity, frustration and potential failures.

Definition of Business Services also determines the rest of the SOA implementation effort. System Services are defined by the systems, and implementing Business Services from System Service is reasonably straightforward.

Overall, proper definition of Business Services should represent the business and be compliant with SOA principles.

SOA Tenets

The following Tenets were initially provided by Dan Box of Microsoft. They are universally recognized fundamentals of SOA:

1. Boundaries are explicit;
2. Services are autonomous;
3. Services share schema and contract, not class;
4. Compatibilities are based upon policies.



While Tenets 3 & 4 are more technical, Tenets 1 & 2 are applicable to Business Analysis. From experience, traditional Business Analysts struggle with explicit borders, and tend to define services that are far from autonomous.

SOA Principles

While SOA Tenets are well-defined, there is no single authoritative set of “SOA Principles” as many authors put forward their points of view. However, several principles are often repeated:

- Transparency: Usage of another Business Service should be explicit;
- Context-independent Service Contracts: the same service contract should work within different existing and future processes;
- Vendor-independent: adherence to standards and business needs rather than suggestions of a specific vendor;
- Stateless & Asynchronous: the service should maintain the state of entities rather than the state of a communication;
- Abstract/Conceptual: the definition should be abstract enough to survive replacement of a software package.

SOA Business Analysis Expectations

As an activity positioned between business and IT, SOA Business Analysis is expected to:

- Capture context and requirements in a form that will enable further Architecture and Design activities.
- Introduce the business to the concepts of Service Orientation to enable further usage of Service capabilities.

To enable effective Service Architecture & Design, Business Analysis should:

- Cover the scope of an enterprise’s collaborating parties;
- Separate processes from capabilities that enable those processes;
- Support the level of abstraction required by SOA;
- Employ specification techniques that will focus on conceptual entities and their transformations rather than systems and processes;
- Avoid over-specification to maximize re-use opportunities: for example, an arbitrary definition of a process can make re-use of existing services that produce the same outcome impossible.

Additional Expected SOA Capabilities

Externalization of Services

Service-Oriented Architecture is seen as an opportunity to eventually externalize services initially designed for internal consumption.

Outsourcing

Outsourcing of non-core business processes has become increasingly common. However Enterprise Architecture support for outsourcing remains a critical issue. IT solutions for outsourcing should combine:



- Effective integration of outsourced and in-house processes;
- Adequate protection of information from non-sanctioned access and update, extended to all information not involved in outsourced activities;
- Additional capability to monitor productivity and quality, as well as provide assistance if required, to compensate for elimination of direct contact and common corporate culture;
- Ability to change supplier of services, or engage multiple suppliers to facilitate costs and quality improvement.

Service-Oriented Architecture seems like a perfect solution to effective outsourcing of business processes. However to realize this promise Business Services should be designed along the most logical lines of separation, with attention to all above factors. Needless to say, that can be achieved only with significant involvement of business expertise of higher ranks than usually available to IT projects.

Merger & Acquisitions

One of the drivers for SOA is the continuous Merger & Acquisition process. When two enterprises are merged, the board expects significant economy of scale from merging similar functions, including IT.

Although there is not enough history of SOA to rely on historical data, one can expect that more flexible, more service-oriented IT will stay, while spaghetti-integrated IT will be dissected and absorbed.

Enterprise Mash-up

The concept of “Enterprise Mash-up” is actively discussed. As an enterprise develops a set of reliable, safe and well-defined services, these services can be quickly “mashed up” in an application.

The promise of Mash-up is fundamental to the future of enterprise IT. In the early 90s it was considered perfectly appropriate to develop an enterprise application in FoxPro, then in Visual Basic or Access. Later those applications were impossible to maintain, were not scalable, and were impossible to integrate. As a result, the whole idea of developing small applications for niche business needs was purged from enterprise IT.

Ability to Mash-up enterprise Web services into small applications as required is seen as re-birth of rapid support from IT, on much more robust platforms. To support Mash-up capability, the services should be safe to use, easy to understand, and designed for re-use.

SOA & Traditional Business Analysis

Artifacts of the Traditional Business Analysis

Traditional Business Analysis produces two types of documents:

- Business Requirements Specifications, which can be seen as description of Business Processes in relation to a specific software system. Business Analyst will collect business expectations in relation to the usage of the system.
- Functional Requirements Specifications, which describe system behavior in exhaustive detail.



System Focus

Traditional Business Analysts discuss requirements in terms of behavior of a particular software system. As a result, the requirements they collect already presume implementation through a specific system.

That contradicts the key principle of SOA that the solution should be orchestrated from services provided by applications.

Lack of Enterprise Vision

Traditional Business Analysis works within the scope of a specific project and specific system. SOA Business Analysis should provide a solution for an enterprise. Narrow scope would not allow for service definitions that will be re-usable across the enterprise.

It would be a mistake to see this as an issue of scope alone. Mechanistic extension of the scope will not alleviate the problem, but rather exacerbate it. The techniques of traditional Business Analysis cannot handle the extended scope of SOA.

Quality and Completeness Challenge

It is a known factor that the outcome of traditional Business Analysis lacks completeness and precision. The impact of this problem is relatively minor as it is mitigated by further refinement through Interface Design and Coding stages – UI Designers and Developers resolve issues and acquire additional information through meetings with and feedback from Business Analysts and Subject Matter Experts. Indirect proof of that phenomenon is the significant level of problems faced by most of the offshored development projects.

While quiet correction of requirements through implementation can work for traditional development, it poses a critical risk for SOA. Services that are not designed correctly will not be able to deliver required advantage. However modification of a service that is already in production is highly problematic. Producing almost identical service contracts for any new project will make a mockery out of Service Orientation.

Failure to “Encircle” Business & System Services

While traditional Business Analysis focuses on an individual system, integration with other systems is often left out of the scope of the analysis. System-to-system integration is seen as an engineering task to be decided as implementation of requested system behavior.

This practice contradicts the SOA demand that allocation of responsibilities to systems is decided ahead of discussing detailed representation.

Equally traditional Business Analysis usually fails to define explicit borders between services. This happens most often when distinctively different services use the same system.

Temporary Solution – Business Analysis by Architects

The reasons listed above are not unknown to SOA project leaders. That does not mean that all SOA projects without proper Business Analysis will fail.

Anecdotal evidence suggests that in many cases Architects act as Business Analysts, engaging in communication with the business first-hand, and designing from unrecorded knowledge. While this approach may work on few relatively small projects, it cannot sustain enterprise-wide adoption of SOA:



- There is already an acute shortage of SOA Architects. It is highly likely that a future project following the same pattern will not be able to recruit a sufficient number of Architects.
- At the moment vendors and consulting companies send their most distinguished experts to the frontline to gain early credentials and establish strong presence in the SOA marketplace. We can see the *crème de la crème* of the world IT expertise getting involved in discussions with Subject Matter Experts. However as SOA projects become routine and numerous, uniquely qualified experts will be replaced by regular technical troops, making the absence of proper Business Analysis more obvious.
- The current arrangement does not deliver sufficient skills to the business. Even if Architects manage to design effective Business Services, they do not allow sufficient penetration of Service-Oriented culture into business practices to ensure that the business can take advantage of their newly-acquired capability.

Key Principles of SOA Business Analysis

The following principles were identified as the solution to the deficiencies of the traditional Business Analysis:

- Employing conceptual rather than system-specific definitions;
- Integration with Business Process Management;
- Introduction of service-orientation to business processes;
- Applying Visual Modeling for facilitation of conceptual thinking and productivity;
- Using Iterative Modeling techniques to ensure quality of solution.

Integration with Business Process Management

Traditionally mapped Business Processes cross borders of business and even system services. This violates the most basic principle on SOA that service implementation should not be dictated. At the same time, “detailed” Business Processes are practically impossible to work with – they are hugely complex and interwoven.

SOA Business Analysis should work with Business Processes by defining the higher-level processes in terms of using Business Services. Then lower-level processes will describe implementation of individual service contracts.

Visual Modeling

SOA Business analysis employs modeling in UML. The choice of UML over other notations is based on a unique combination of features:

- **Popular and Standard:** UML is a recognized and practically uncontested standard for visual modeling
- **Multiple Overlapping Views:** UML offers multiple options of representing information. For example, the same behavior can be documented as processes (Activity Diagrams) or in the form of State Transitions with focus on individual entities.
- **Extensibility:** UML allows definition of Stereotypes that provide specific interpretations of generic concepts like Class or Use Case.



Iterative Modeling

While it sounds suspiciously similar, one should not see Iterative Modeling as part of Iterative Development. While Iterative Modeling can be efficiently combined with Iterative Development, its main purpose is to replace Iterative/Agile Development with an alternative that offers lower costs and easier maintenance.

Horizontal Iterations

Horizontal iterations are introduced to improve productivity and quality of modeling. Different types of models look from different perspectives. Instead of requesting the modeler to provide a specific type of artifact for a signoff, for example a Use Case Model, the methodology allows Modelers to work on several models at a time. Cross-reference rules and the overlapping nature of UML allow modelers to use one model to improve and complete another.

Horizontal iterations improve productivity by eliminating “Analysis Paralysis” and allowing modelers to choose the part of the model to work on according to the information they or available Subject Matter Experts have.

Horizontal iterations also improve quality of modeling by providing the capability to use one model to verify another.

Vertical Iterations

Vertical iterations are introduced to modeling practices to replace rather unrealistic iterative development of services. Vertical iteration is a sequence of modeling activities that:

1. Start from higher-level definition of the problem;
2. Work down to Solution Model;
3. Use Solution Modeling to produce examples and detailed descriptions of impact on the problem level.

As a result, the main pattern of Iterative Development is preserved. Captured requirements are verified by presentation of outcomes. However, modeling replaces actual development, thus reducing the cost of iterations and enabling non steps like package acquisitions that cannot be iterated through.

Reference Models

The Abstraction Business Analysis is conformant with two leading reference models: Zachman Framework for Enterprise Architecture and OMG’s Model-Driven Architecture

Zachman Framework

Zachman Framework is the leading approach to Enterprise Architecture. Abstraction Business Analysis is mapped to Contextual, Conceptual and Logical levels of the Zachman Framework, and references to the Zachman Matrix.

Model Driven Architecture

Model-Driven Architecture is the set of standards introduced by the Object Management Group (OMG) in 2001. MDA recommends developing solutions by gradually working through three sets of models.

The Computation-Independent Model (CIM) should be abstracted from the notion of computation. The objective of CIM is to model Business Context and Requirements without applying any restrictions on possible IT solution.



The Platform-Independent Model (PIM) is a generic model providing structural and behavioral specifications in generic UML. PIM should be used for general XSD & DDL definitions, software classes and other more detailed specifications.

Platform-Specific Model (PSM) is a set of detailed specifications that can be used for XSD, DDL or code structure generation.

Abstraction Business Analysis is compliant with MDA and can be implemented within MDA structure.

Contextual Business Analysis

The objective of Contextual Business Analysis is to collect requirements and specifications that are applied to the enterprise as a whole. This includes general industry definitions, existing business context, as well as rules, requirements and regulations applied by the regulatory authorities.

Business Context Analysis

Business Context Modeling defines external parties and analyses communication and collaboration between parties. The complexity of Business Context Analysis significantly depends on the complexity of the enterprise.

Capability Modeling

Capability Modeling captures the capabilities of an enterprise without getting into the details of the Business Processes. A number of leaders in enterprise software concluded that Business Process or “how” modeling is too costly and too restrictive for today’s IT. The alternative is Capability or “what” modeling that provides sufficient information without getting into details too early.

Common Information Model

Contextual Business Analysis extends past the Zachman definition of “Contextual” to include lower-level artifacts that do not depend on definitions of Business Services.

The most important addition is the Common Information Model that provides precise structural definitions of all terms used through the Analysis. The Common Information Model is also invaluable as a reference point to all other specifications. This ensures Quality of Modeling, enables Iterative Modeling and is instrumental for navigation and automatic evaluation of the model.

Business Services Definition

Business Services Definition is the key activity of the Abstraction Business Analysis that serves two complementing objectives:

- Produce effective system-agnostic definitions of Business Services;
- Establish productive collaboration with the business to ensure business adoption of SOA.

Business Service & Business Service Contract Definition

Both Business Services and individual Business Service Contracts should be defined in a system-independent manner. Most traditional approaches fail to achieve required precision without



referring to specific implementation details. However Abstraction Business Analysis achieves desired precision of Service definition through:

- Referring to the Service's implementation of Common Information Model elements;
- Referring to conceptually defined State Transitions;
- Defining the interface in terms of Common Information Model;
- Applying Business Use Case Modeling techniques to specify inter-dependencies.

Interactive Collaboration Modeling

Interactive Collaboration Modeling is the key activity in defining Business Services. Through the course of Interactive Collaboration modeling:

- SOA Analysts/Architects achieve progress in understanding of the business functioning sufficient to produce Business Services understandable by the business
- Business stakeholders and SOA Champions achieve sufficient understanding of Service-Oriented to be able to accept Business Services after contributing to their definition

This modeling activity relies on Visual Modeling to visualize proposed solutions. Visual Modeling and advanced interactive techniques enables quick turnaround of ideas and solutions and delivers optimal result within feasible timeframe.

Future Usage Workshops

Service Contract definitions should not only deliver to existing business capabilities, but also to account for the "Mash-up Factor". A Business Service should enable quick implementation of additional functionality by providing a safe, dependable and easy to understand interface. Separate activity may be required to "future-proof" Business Services definitions.

Consider an example where an enterprise is providing services to the general public. It is logical to identify Customer Management as one of the Business Services, and adding a new Customer as a Service Contract. Another Business Service will be a Customer Service that will provide actual services to individual customers. It is possible that the responsibility to ensure that every physical person is represented by only one Customer Record is quite possible.

A "Future Use Workshop" will identify that any new application will have to duplicate this responsibility. "Add New Customer" is an unsafe Contract that may compromise future information integrity, and produce a more robust definition that would incorporate checking for duplicates.

Additional Modeling Sources

Traditional BPM & Requirements Artifacts

Existing Requirements Artifacts collected into Word documents, Requirements Management Systems like DOORS or in Excel spreadsheets represent significant value that should be re-used rather than discarded. Abstraction Business Analysis includes the workflow handling the incorporation of "Raw Requirements", as opposed to Abstraction-structured requirements in UML, and producing traceable SOA-compliant derivation.

The outcome of a pre-SOA Business Process Mapping effort can be used to an even greater degree. Abstraction Business Analysis can include:

- Using Business Processes as traceable source for core Abstraction Business Analysis activities;



- Using the Business Services definitions for documenting “after” processes.

Bottom-up Modeling

Sometimes top-down Business Analysis can be helped by reverse-engineering lower-level artifacts, and then abstracting them to proper level. The low-level artifacts can include:

- XSD Schemas, especially industry standards;
- Logical and Physical Data Models.

In many cases, lower-level definitions like Logical Data Models are too detailed and designed with particular implementation in mind. Also, SOA requires close collaboration with the business that cannot be achieved on the basis of XSD Schemas or DDL, even if presented graphically.

Abstraction Business Analysis introduces guidelines and support for abstracting logical specifications into Conceptual and Contextual definitions. This way SOA implementation can benefit from international standards and existing Artifacts without compromising Business-IT alignment critical for SOA.

Practical Details

Selecting Modeling Software

SOA is a hot-ticket item, and any vendor of a visual modeling software suite will claim support for SOA in the very near future. However, not all suites will be able to deliver effective, consistent and robust SOA practices. Through our work we identified the following criteria:

- Ergonomics: Modelers will have to use the software during the workshops, in front of a group of people. Cumbersome and non-intuitive interface will make interactive modeling slow and unappealing to participants.
- Scalability: The model will have to be used by many people in enterprise IT, often at the same time. Modeling software of choice should support working with massive models by significant groups of Modelers.
- Automation: Enterprise SOA will require working with thousands of modeling elements. As repeatedly stated above, many tasks like validation, transformation, or meta-model modification, require running some code against the model. Ability to routinely develop scripts and other additional functionality makes working with massive and complex models possible.
- Reasonable price. Throughout this white paper we have stressed the need for mass adoption of modeling within an Enterprise IT. However it is unrealistic to expect an Enterprise IT to invest \$10,000+ per seat for IT staff actively participating in Solution Implementations. The price of the tool of choice should make the adoption strategy feasible.

Fortunately, Business Abstraction’s long partnership with Sparx Systems provided us with the solution that works. Unlike many of the competing tools, Enterprise Architect:

- Is highly affordable. Its license cost does not create a barrier for mass introduction.
- Is highly customizable. We were able to offer both generic and custom-specific extensions within consulting frameworks. Its COM and proprietary MDG interfaces are refreshingly effective especially when compared with tools that offer VBA customization.



- Is a pleasure to work with in front of the customer. Interactive Visual Modeling is a highly demanding undertaking so the tool of choice should deliver maximum editing and navigation with minimal difficulties and disruptions.

Process Modeling

SOA Business Analysis is a complex process, and as anything else in IT it should be documented if a client intends to ever take the process over from consultants. There is an established practice of documenting process in a text document. However, a document of that kind is hard to produce, and even harder to modify. As a result, the value of out-of-date documents quickly deteriorates, while required overhead increases.

Business Abstraction provides process modeled in UML using Sparx Systems Enterprise Architect. Every staff member capable of modeling will be able to read the process, and the leaders of the activity will find it very easy to modify.

Hub-and-Spoke Modeling

The lack of modeling skills is universally accepted as a key challenge for implementation of model-driven techniques. Modeling skills such as being capable of building effective, robust, readable and navigatable specifications using modeling notation, are quite different from the ability to do basic operations with modeling software.

The essence of modeling is radically different from specification writing. While the latter significantly relies on human effort to understand the documents and use other sources to clarify them, the former should be precise enough to enable “narrow-focused” work.

Most training offered by leading providers concentrates on the formal side of using the software, or on most basic implementation of a technique. This training is highly unlikely to deliver skills sufficient for practical modeling.

Most advanced, interactive courses like Business Abstraction training go to a greater length to develop individual modeling skills. However, even in that case, a lot depends on individual capabilities, talent and motivation. The required pace of introduction of model-driven techniques does not leave enough space for a generous learning curve. There are simply not enough people in the IT industry capable of being quickly trained into proficient modelers to fulfill the demand.

To enable effective introduction of Model-based processes Business Abstraction introduced the “Hub-and-Spoke” approach to resourcing SOA Business Analysis. The approach suggest an experienced Modeler (hub) who does not conduct communication with stakeholders, business etc, acting as a “scribe” for several more traditional Business Analysts, System Analysts or even Architects (spokes).

Hub-and-Spoke Modeling delivers positive ROI by increasing productivity of subject-matter experts, reducing review and approval cycles, and enabling benefits of good model-driven practices.

Further Information and Legal Notes

Further information can be obtained from Business Abstraction:

UML and MDA and trademarks of Object Management Group