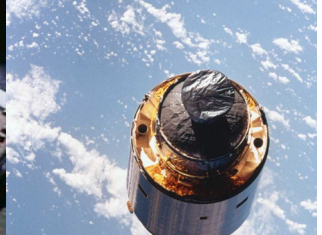


Dunelm

Services Limited



Consultants in Applied Research & Development for
Information & Communications Technology



Session 3: SOA Maturity Models

Service Oriented Architecture: Preparing Your Business & IT Products

2009 © Dunelm Services Limited

PUBLICATION INFORMATION

The publication history for this booklet is:

<i>Version</i>	<i>Date</i>	<i>Summary of Amendments</i>
A	June, 2009	The first release of this booklet.

License

This material is made available under a Creative Commons License. This License, termed ‘UK: England and Wales Attribution version 2.0’, enables anyone to copy, distribute, display and perform the work, and to make derivative works. The only condition is that attribution credit must be made to the original author; details for how this credit must be given are available on the Dunelm web-site at www.dunelm.com/projects and following the links for the Digital 20/20 project. Further details on the Creative Commons License are at: creativecommons.org/licenses/by/2.0/uk/



EXECUTIVE SUMMARY

One of the most significant problems with using architectures and frameworks is trying to work out how they fit into the current systems. In most cases a current system has to be migrated to the new form as opposed to creating a new system from scratch. For many years, the software industry has advocated the use of Maturity Models (the most widely adopted being the Capability Maturity Model Integration for software development). In the case of SOA, a maturity model is needed to enable organizations to identify the extent to which they either use or could use SOA. The same model can then be used to help organisations plan how to improve their use of SOA.

Apart from being able to describe the level of maturity the model also needs to provide metrics that can be used to measure the level of maturity. It is these measures that an organisation uses to determine their current level of maturity. Those same metrics are then used to guide an organisation in how to improve their adoption of SOA to increase business productivity.

TABLE OF CONTENTS

Publication Information	2
Executive Summary	3
1. Introduction	5
2. SOA Maturity Models	8
3. A Simple Maturity Model.....	10
4. Open Group's SIMM	13
5. OSIMM Maturity Levels	14
6. Organisational Dimensions.....	19
7. Domains & Maturity Indicators.....	22
8. Maturity Model Matrix.....	26
9. Assessment Methodology	28
10. Benefits of Increased Maturity	32
11. Product Perspective	36
12. System Perspective	38
13. Two Simple Case Studies	40
14. In Conclusion.....	43
Appendix A – Bibliography	46
Appendix B – Acronyms.....	47
Index	48

1. INTRODUCTION

How does the senior management team of an organization determine how to use SOA? In fact, how do they even become aware of SOA? Even though SOA is business-oriented, it is still a tool of which many senior managers are unaware. Remember, that SOA has its roots in technology and IT consultancy. The answers start with the senior technology managers and their appreciation of the need to show that the IT provision within an organization has direct business benefit. SOA maturity models provide one way in which IT managers can show how Web-based solutions improve business processes.

There are three learning objectives for Session 3 and the material on SOA maturity models, namely:

- The first is to introduce and explain the SOA Maturity Model from the Open Group. This is called the Open Group Service Integration Maturity Model, the OSIMM and was developed with IBM. The OSIMM is one of several maturity models for SOA (this is not an attempt to suggest that it is the best). Instead the aim is to identify the characteristics required for such a model. The OSIMM is schematically represented as a matrix of seven maturity levels with seven organisational dimensions. This creates a matrix of 49 cells with each cell assigned a number of maturity indicators which are used to decide the state of maturity of an organization being assessed using the model;

- The second is to explain how an organization can gauge its level of maturity as defined by the Maturity Model. An important feature of a maturity model is that it must be easy to map current systems onto it and also allow new system designs to be evaluated. It must be possible to map the properties of a system onto a set of points in the maturity model so that the system SOA maturity can be expressed. The next step is that the maturity model must indicate what changes have to be made to improve the SOA capabilities of a system. A substantial questionnaire is used to evaluate the status of the organization and the responses to these questions are used to determine the value of the maturity indicators for each cell in the matrix;
- The third is to explain how an organization can use the maturity model to plan the development of its own systems and/or products and to be more innovative. A maturity model is a tool for use by the senior technical team of an organization. Two simple examples will be used to explain how the assessment methodology in the OSIMM can be used to evaluate SOA maturity. On the matrix visualization, the aim is to have a mapping on the right hand side i.e. the highest level of maturity for all seven dimensions.

The material covered in this session starts with a description of the characteristics of an SOA maturity model. This is followed by a description of a simple SOA maturity model from Sonic and an in-depth analysis of the OSIMM from the Open Group. The uses of SOA maturity models from the perspectives of a product

developer and as part of system deployment are discussed. Two simple case studies are used to explain how the OSIMM could be used.

Maturity models are business tools and should be used as a guide. Like all business tools they are limited and imperfect. Such tools should be used as part of a continual evaluation process. Improvement should also be continual and targeted on one facet of the business at a time. The aim should be to achieve optimal performance for the organization as a whole and this will occur with a process of continual evaluation and improvement. Large-scale changes should be avoided as these are difficult to manage and liable to destabilise the organization as a whole.

2. SOA MATURITY MODELS

A maturity model provides a systematic framework for carrying out benchmarking and performance improvement. The original software Capability Maturity Model® Integration (CMMI) [CMMI, 02] was created by the Software Engineering Institute at Carnegie Mellon University (USA). CMMI is a process improvement approach that provides organizations with the essential elements of effective processes. It can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps integrate traditionally separate organisational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes.

The SOA equivalent model is focused on how SOA can be used to support and improve business processes. The characteristics for a maturity model that can be used to evaluate the SOA capability of an organization are:

- It must define how an organization's SOA capabilities can be identified and quantified. Quantification is essential otherwise improvement can only be defined in terms of 'either doing or not doing' a process, 'using or not using' a technology, etc. This makes small but continual improvements hard to measure and evaluate. Small and continual improvement is what would be expected even in the most mature and advanced of organizations;

- Support a range of descriptions and classification from the most primitive of starting points to the most optimal, aspired operational capability. Organizations will be in different stages of maturity and in an enterprise environment, different business units will have different levels of maturity. It is important that any maturity model must be applicable to parts of an organization or business process because the adoption of SOA may not be appropriate to an organization as a whole;
- It must provide a process by which the required incremental improvements can be defined so that an organization can make the changes. The aim of a maturity model is to help an organization to improve. The incremental changes do not have to be similar in range of scope and implementation effort;
- It must provide a methodology that an organization can use to determine its level of maturity. External help may be required to use the methodology but it is essential the senior management in the organization has ownership and fully participates in the process.

Two different SOA maturity models will have different quantifications of maturity but they must cover the same organisational aspects otherwise there would be no agreed, common understanding of SOA. It doesn't really matter which maturity model is used. It is more important to use a process to help adoption as opposed to attempting to identify and use the 'best' process.

3. A SIMPLE MATURITY MODEL

There are various maturity models available for SOA. Sonic Software and its partners jointly developed the SOA Maturity Model. This is a simple model that is designed to show the increasingly positive impact of SOA adoption from a business benefits perspective. It provides IT decision makers with a simple framework for benchmarking the strategic value of their SOA implementation and a model for visualizing future success. This maturity model provides a framework, with five levels of maturity, as shown in Figure 1, for discussion between IT and business users about the applicability and benefits of SOA in an organization across five levels of adoption maturity. The five levels are:

- Initial services (level 1, the least mature) – the start of the journey is to begin research and development projects on SOA and services. The focus should be on immediate organisational needs and should include support for custom integration of a small number of key services. This work will typically focus on the core Web Services technologies. A key step, upon completion, is to undertake a detailed evaluation of these initial projects to quantify cost, time and business benefits and to establish best practice processes for future projects;
- Architected services (level 2) – the primary business benefits at this level of maturity are IT cost reduction and improved control for multiple integrated applications. The approach

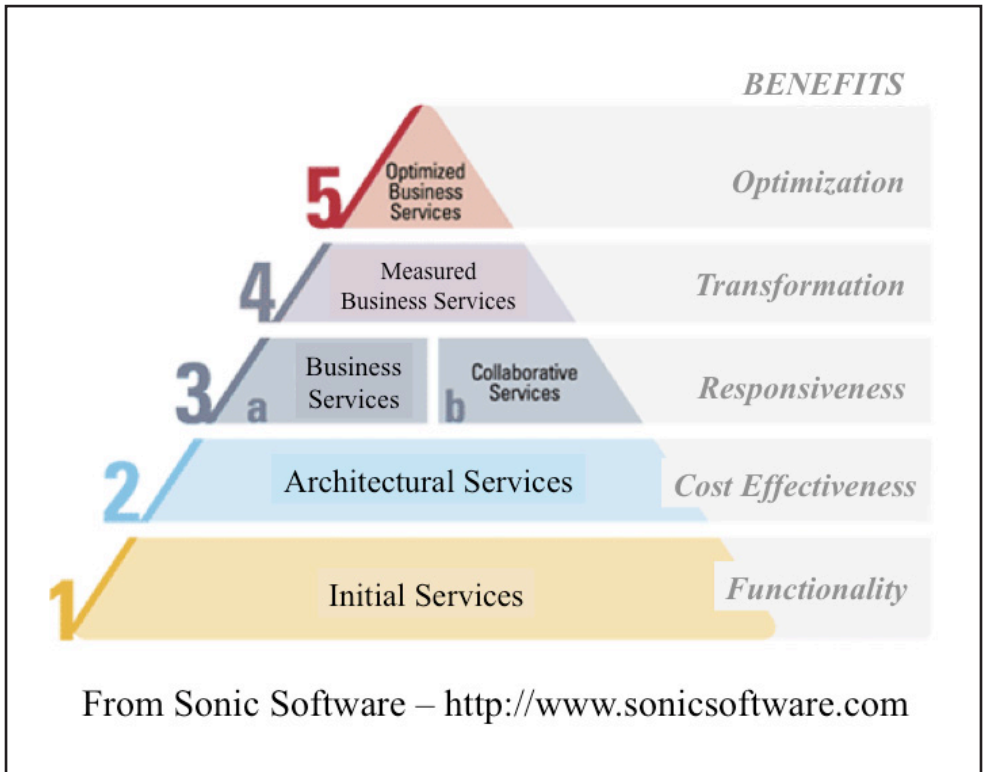


Figure 1 Simple maturity model from Sonic Software.

is to undertake institutionally co-ordinated use of SOA as a technology and to integrate SOA into an organization wide development process. At this point the broader set of Web Services standards will be employed;

- Business services (level 3a) – at this level of maturity, service reusability begins and the organization as a whole becomes more able to change business processes quickly and effectively. SOA now underpins the full range of business processes and there is a clear understanding of how SOA can

be used to further improve the operational performance of the organization;

- Collaborative services (level 3b) – integration with external partners is realised using SOA. Cross-enterprise security is enabled and long running transactions are supported. New collaborations can be easily established provided the new partner also supports the appropriate SOA standards;
- Measured business services (level 4) – clear enterprise-wide business-oriented performance metrics are established and the business now undertakes real-time performance and capability evaluation. All business units become responsible for driving their own continual improvement;
- Optimised business services (level 5, the most mature) – at this level of maturity SOA is delivering enterprise-wide business optimization. The organization has firmly established processes that are continually evaluating and optimising the business as a whole. The benefits of SOA are clearly recognised from the business perspective.

4. OPEN GROUP'S SIMM

The Open Group' Service Integration Maturity Model (OSIMM) contains both a framework and a process [OSIMM, 06]. As a framework, the OSIMM provides seven dimensions across seven maturity levels. OSIMM defines the organization in terms of a set of "dimensions", representing different views (e.g. business, architectural) of that organization. The seven dimensions are the Business, Organization, Methods, Application, Architecture, Information and the Infrastructure. The seven maturity levels are Silo, Integrated, Componentized, Services, Composite Services, Virtualized Services and Dynamically Re-Configurable Services.

Within each dimension the organization is modelled in more detail and each dimension is divided into several "domains", each domain having a set of possible maturity indicators indicating the level of maturity of that domain. The maturity level of each domain can be aggregated into the maturity level of the dimension, and the total set of maturity levels for all the dimensions provides a holistic view of the service integration maturity level of the organization.

As a process the OSIMM contains a simple methodology for ascertaining the level of maturity of an organization for each of the seven dimensions. This creates a maturity mapping that is used by the senior management team to identify which parts of the organization should undergo improvement.

5. OSIMM MATURITY LEVELS

At the heart of OSIMM is a conceptual model depicting seven levels of business and IT maturity within an Enterprise. Each of the seven levels reflects a possible abstract state of an organization in terms of its maturity in the integration of its services (business and/or IT). OSIMM uses this model to help determine the Service Integration Maturity level of the organization by measurement of certain aspects of parts of the organization that are to undergo transformation. The seven maturity levels, as shown in Figure 2, are:

- Silo level (least mature) – individual parts of the organization are developing their own software independently, with no integration of data, processes, standards or technologies. This severely limits the ability of the organization to implement business processes that require co-operation between the different parts, and the IT systems cannot be integrated without significant manual intervention, such as re-keying and re-interpretation of data;
- Integrated level – technologies have been put in place to communicate between the silos, and to integrate the data and interconnections. The construction of an IT system that integrates across different parts of the organization becomes possible. However integration does not extend to common standards in data or business processes. Therefore to connect two systems, it requires a, possibly complex, conversion of the data, operations and protocols used by these systems.

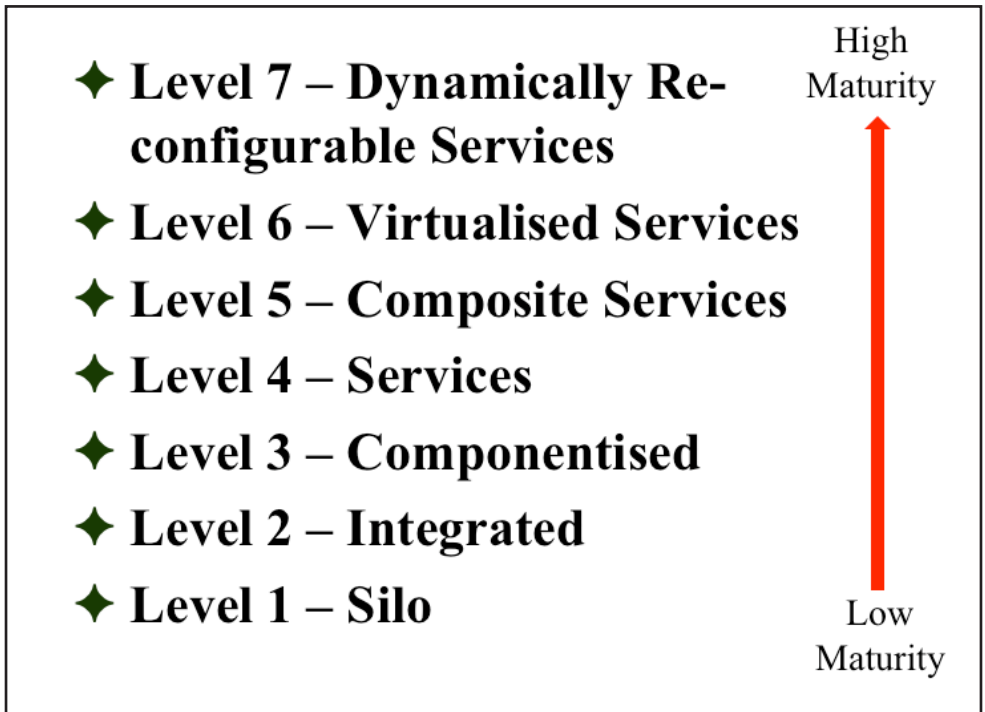


Figure 2 Maturity levels in the OSIMM.

Each such connection may require bespoke code and adapters, leading to a proliferation of software that is difficult to manage and complex to code. Therefore, it is not easy to develop new business processes;

- Componentized – the IT systems in the silos have been analysed and broken down into component parts, with a framework in which they can be developed into new configurations and systems. There may also be some limited analysis of the business functionality into components. Although components interact through defined interfaces, the

way that these components interact together is not loosely coupled, which limits the interoperability between systems in different parts of the organization (or even different organizations within the business “ecosystem”) and causes difficulties in development of business processes that can be constructed across the parts of the organization;

- Service – composite applications can now be built from loosely-coupled business services. The way that services may be invoked is based upon open standards and independent of the underlying application technology, and running on an IT infrastructure that supports the services with suitable protocols, security mechanisms, data transformation and service management capabilities. The services may therefore interoperate across all of the parts of the organization and even across different organizations within the ecosystem, and may be managed by assigning responsibilities for Service Level Agreements (SLAs) to relevant parts of the organization. However the flow of control within a composite application is still defined by bespoke programming, rather than by a declarative flow language. The business functionality has been analysed in detail and is broken down into business services residing within a business architecture that ensures that business services will interoperate at the business level. In addition, it is possible to define the services via a specification language that unambiguously defines the operations performed by the service, permitting the construction of a catalogue of services. The combination of IT and business service architectures permits the construction

of systems based upon these services, operating right across the organizations in the ecosystem. However at this stage the composition of services is still performed by developers writing bespoke code, thus limiting the agility of the development of new business processes;

- **Composite Services** – it is now possible to construct a business process for a set of interacting services, not just by bespoke development, but by the use of a composition language, such as BPEL, to define the flow of information and control through the individual services. This permits the assembly of business services into composite business processes, which may be short or long running, without significant construction of code. Therefore the design and development of business services is agile and may be performed by developers under the close guidance of business analysts;
- **Virtualized Services** – the business and IT services are now provided through a façade, a level of indirection. The service consumer does not invoke the service directly, but through the invocation of a “virtual service”. The infrastructure performs the work of converting the virtual invocation into a physical call of the real service, and may as part of this conversion change the address, the network, the protocol, the data and the synchronization pattern that is contained in the call. Such conversions may be a complex service in their own right, such as the transformation of data from one data model to another. The virtual service thereby becomes more loosely

coupled from the infrastructure on which it is running, permitting more opportunities for the composition of business services. This is in contrast to the lower levels of service maturity where the service is more closely coupled to the infrastructure. Although virtualization has been used in non-SOA systems, this level extends the concept (and advantages) of virtualization to business services;

- Dynamically Re-configurable Services (most mature) – prior to this level, the business process assembly, although agile, is performed at design time by developers (under the guidance of business analysis and product managers) using suitable tooling. Now this assembly may be performed at “runtime”, either assisted by the business analysts via suitable tooling, or by the system itself. This requires the ability to access a repository of services and to query this repository via the characteristics of the required services. In its simplest form, these characteristics may have been defined in advance, restricting the system to selecting and locating specific instances of services.

6. ORGANISATIONAL DIMENSIONS

An organization's SOA or desired SOA scope can be assessed using a set of dimensions that address organisational issues ranging from the business to the technology perspectives. These organisational dimensions, as shown in Figure 3, are:

- **Business** – the Business dimension is focused around the business architecture, the organization's current practices and policies around the business architecture, how business processes are designed, structured, implemented and executed, how costs of IT capabilities are allocated throughout the enterprise, and how well the IT capabilities support flexibility of the business, business agility and business SLA management. As the business dimension includes IT strategy it would thus include a high-level quantifiable monetary value justification for moving from one maturity level to a higher level of maturity;
- **Organization** – this dimension is focused on the structuring and design of organizations and resulting measures of organisational effectiveness in the context of an SOA, for example, SOA governance. This includes the types and extent of skills, training and education that are available within the organization, the existence of a formal governance process to keep IT activities and capabilities aligned with the needs of the whole business, how IT management is organized and how costs are allocated;

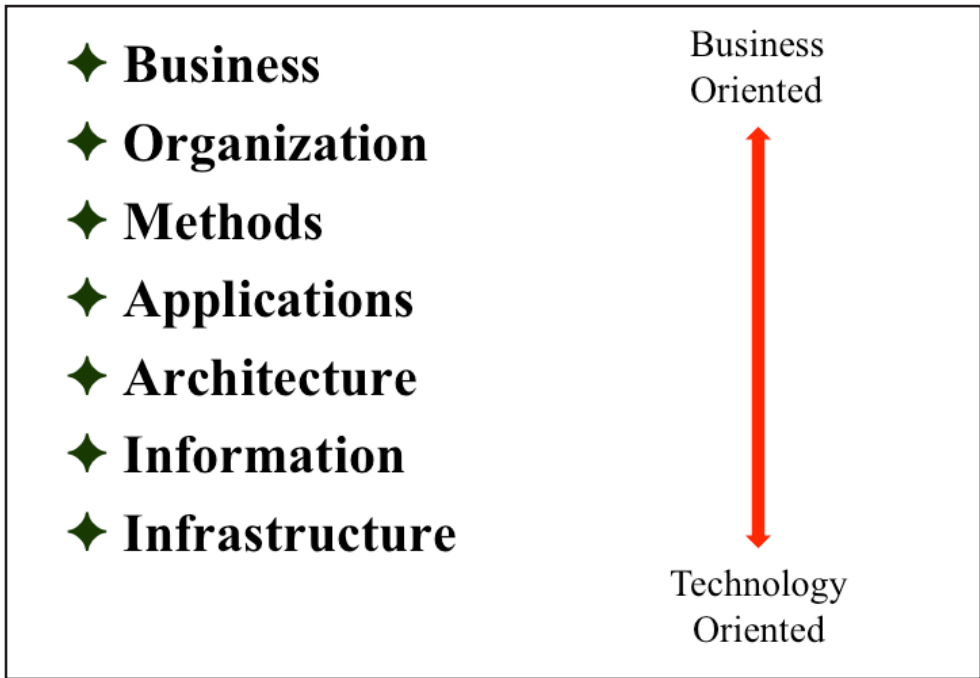


Figure 3 Organisational dimensions in the OSIMM.

- **Methods** – this dimension is focused on the methods and processes employed by the organization for its IT and business transformation, and the organization’s maturity around the Software Development Life Cycle such as the use of requirements management, estimation techniques, project management, quality assurance processes, design methodologies and techniques and tools for designing solutions;
- **Applications** – the application dimension is focused on application style, structuring of the application and functional decomposition, reusability, flexibility, reliability

and extensibility of the applications, understanding and uniform use of best practices and patterns, whether multiple applications have been created to serve different lines of business with essentially the same functionality, and the availability of enterprise schema and object models;

- **Architecture** – the Architecture dimension is focused on the topology, data characteristics, business information model, integration techniques, enterprise architecture decisions, standards and policies, web services adoption level, experience in SOA implementation, SOA compliance criteria, and typical artefacts produced;
- **Information** – this dimension is focused on the information modelling aspects, access to enterprise data, abstraction of the data access from the functional aspects, data transformation, service and process definition, handling of identifiers, security credentials, knowledge management, and content management;
- **Infrastructure** – the Infrastructure dimension is focused on the organization's infrastructure capability, service management, IT operations, IT management and IT administration, how SLAs are met, how monitoring is performed and what types of integration platforms are provided.

7. DOMAINS & MATURITY INDICATORS

A domain is a sub-division of a dimension for which there is a maturity indicator i.e. each dimension has several domains and each domain has a maturity indicator. The maturity indicators are assessed by a set of questions for each dimension. Some of the standard set of questions is described below, but it is anticipated that more questions will be added as part of the customisation of the assessment framework, at the start of an OSIMM assessment.

For the Business dimension the questions focus on how the business process management activities contribute to the operation objectives of the organization. Typical questions include:

1. What are the major business drivers for this initiative?
2. Is your current Business Process Architecture formally defined and documented?
3. Is your Business Process Architecture complete and current?
4. How is ROI measured in Business Process Management?
5. What is the current cost model?

For the Organization dimension the questions focus on ascertaining the organisational strength in IT and the capability

to address the business perspective of SOA. Typical questions include:

1. How does IT governance relate to your SOA?
2. How is the IT governance related or aligned with the corporate governance?
3. What are the governance functions and responsibilities?
4. How would you describe your IT cost model?
5. How big is your development team and how is it organized?

For the Methods dimension the questions focus on detailed IT development processes and capabilities of the organization.

Typical questions include:

1. What are the current requirements elicitation and requirement management practices?
2. What design methodologies and best practices are you currently adopting?
3. What is the current practice for service development and management?
4. What is your current project management framework?
5. What is your current quality assurance process?

For the Applications dimension the questions focus on the set of applications that are currently deployed. Typical questions include:

1. What is your current application development style?
2. How common is reuse in your organization?
3. What types of reuse do you engage in and how is reusability measured?
4. What types of languages does your organization use?
5. How reliable are your business-critical applications?

For the Architecture dimension the questions focus on the underlying architecture used to support the deployed IT services and applications. Typical questions include:

1. How would you characterize your architectural topologies?
2. What is the standard communication style in your architecture?
3. How is integration achieved in your architecture?
4. How mature are your Web Services implementations?
5. Does your organization have, or are you developing, a Business Information Model to standardize data and message formats and concepts across the enterprise?

For the Information dimension the questions focus on use, access and flow of data within the business. Typical questions include:

1. Do you have independent data models for different applications?
2. Do you have a common data model across all applications?

3. Do you have different data models but mapping rules to convert between the different models?
4. Do you have difficulty in moving data between applications? For all applications? For only some?
5. If you have a common data model or mappings between multiple data models, how is this defined? By programming objects in APIs? By XSD schemas? By written documents? By other computer-based modelling tools? By other non-computer-based modelling tools?

For the Infrastructure dimension the questions focus on how the IT infrastructure is managed. Typical questions include:

1. How are your IT SLAs transformed from the business SLAs?
2. Have you defined SLAs around quality of service? How is this monitored and measured?
3. Have you defined any SLAs around security and privacy? How is this measured and monitored?
4. What platforms are currently in use for integration?
5. What is your current change management process?

It is important to ask as many different questions as possible and to make sure that they are answered honestly. The aim is to get a clear view of the real current state, not one that is imagined. Some of the questions for the different dimensions should cover the same topics so that consistency of responses can be checked to confirm accuracy and reality.

8. MATURITY MODEL MATRIX

The OSIMM matrix, as shown in Figure 4, has the seven levels of maturity on the x-axis (each level is a column in the matrix) and the seven dimensions on the y-axis (each dimension is a row in the matrix). Each cell in the matrix corresponds to a summary of the maturity level for all the domains “underneath” the cell. There are 49 cells each with a label that summarizes the results of the maturity indicators assigned to that maturity/dimension intersection.

For example, consider the cell ‘Information x Silo’, with the label “Application Specific”. The Information dimension has a number of domains, including skills, management and others. For each domain there is a maturity indicator for the Silo level that, if present in the organization, would suggest that the organization is at the Silo level. These include “Skills are Application specific” and “Project Management is specific to each team”. If a significant number of the indicators for the Silo level are present then this suggests that the Information dimension for the organization is basically at the Silo level, as summarized by the label “Application Specific”. Each Dimension may be assessed in a similar way, leading to a level assessment for each dimension of Business, Organization etc. The overall picture, in terms of the assessed maturity level for each dimension may itself be created to provide a view of the overall maturity level of the organization. This creates an overlay on the matrix where each dimension has an allocated maturity level.

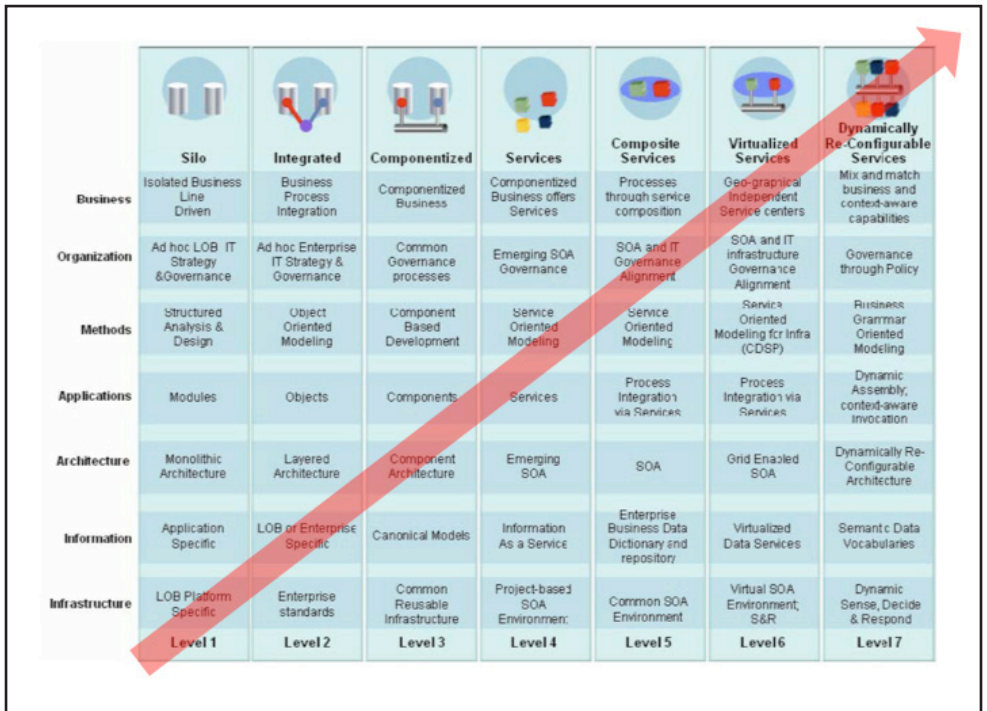


Figure 4 Maturity matrix for the OSIMM.

In general, the aim should be to move all dimensions to a maturity level of seven (dynamically re-configurable services). If the business-oriented dimensions are at level 7, this is a reflection of how well SOA is used to empower the business process operations as opposed to being more technology focused. An organization that is poorly prepared for SOA will find itself mapped onto the left hand side of the matrix i.e. the organization cannot show how the IT infrastructure impacts the business processes etc. Improving maturity is the process by which the mapping of the organization moves from left to right on the matrix.

9. ASSESSMENT METHODOLOGY

The OSIMM is used to support an SOA assessment of an organization. The scope of such an assessment using OSIMM could be a single project or a line of business, the entire enterprise, or a service ecosystem. Analysis consists of the following three activities:

- Assessment of the current state of the business, organization and IT with regard to their maturity level in the context of a SOA;
- Goal state identification and definition, building a vision of what the client's business, processes, staff and IT plant would look like if they were transformed to a highly-capable SOA;
- Production of the recommendation report that identifies the current maturity levels of the various domains, describes the ideal goal state, and defines a roadmap showing how the client can advance to that goal state.

An OSIMM analysis is conducted using the following steps:

- Identify the current pain points in the business and define the scope of activity – the pain points define where the organization considers that its processes need to be improved, and can be used to focus the subsequent OSIMM analysis. At this stage an initial list of pain points must be determined, and the scope and structure of the OSIMM engagement is agreed. The dimensions and domains in the OSIMM may be used to

assist the definition of the scope;

- Use these pain points to focus the assessment framework and maturity indicators – on the basis of the agreed scope, an assessment matrix must be created, based upon the full OSIMM matrix, but tailored to focus on the key pain points. The selection is made from the existing set of dimensions, domains, maturity indicators and questions, but new indicators or questions can be developed as appropriate;
- Determine the current maturity level by comparing the current state of the organization against the maturity indicators – using the assessment matrix developed in the previous step, interview the key staff from the organization to assess the current state of the organization and hence its current maturity level. The interviews must be based upon standard questions provided within OSIMM, and can include additional questions considered relevant by the OSIMM practitioner. On the basis of the answers to the maturity indicators and other questions, the current maturity level must be determined for each domain, and aggregated through the dimensions to the overall state of the organization. The use of an automated tool is recommended to facilitate this process;
- By considering the maturity indicators and their effects determine the reasons for the pain points – an assessment can be made of the reasons for the pain points in terms of the characteristics of the current business and IT processes at their current maturity level;

- By considering the characteristics of the maturity indicators in higher levels, identify how the pain point reasons may be alleviated, and which target level of maturity is suitable – the future desired state must also be determined from the interviews with the key staff, with particular focus on those individuals who may be seen as having a good understanding and vision of the future requirements. Consideration can be given to the reasons for the pain points determined in the previous step, and how these may potentially be resolved by the characteristics of the future business and IT processes at their future maturity level;
- Compare the current and target level maturity indicators on the roadmap to determine what development is required – the previous steps have identified the current and future maturity levels across all of the domains and dimensions in the assessment matrix created in the first step. The OSIMM practitioner must now determine the gaps between the current and future maturity levels, and create the roadmap that takes the organization from current to target. The maturity indicators for each domain must show the current and desired state, and the steps in the road map must be constructed in order to take the domains from current to desired, and to alleviate the pain points. This should also take account of the constraints and prerequisites that will exist between the different IT and business entities that need to be put in place. It should be noted that the output of the OSIMM road map is intended to provide a relatively high level statement of the activities that need to be undertaken, and that further more

detailed analysis can be undertaken, outside of the OSIMM analysis, of the precise nature of the activities.

It should always be noted that any methodology should be ‘tweaked’ to fit the problem in hand. Never let a methodology prevent you from doing the detailed work required.

10. BENEFITS OF INCREASED MATURITY

The benefits that accrue to an organisation from moving to a higher level of maturity are:

- From Silo to Integrated – organisations transforming from a Silo to an Integrated maturity level will significantly reduce operational and maintenance costs. These cost reductions are realized by reducing redundant and laborious data entry processes, reducing batch cycles to transform and transfer the data from one system to another. From this transition the data is available on a real-time basis, with reliable delivery of data and automated data format conversion for the integrating systems. The transformation from structured programs to object oriented would also leverage reusability of the code and help in reusability and reduction of the software maintenance complexities since the software is more modular. The modular code increases readability of the code thus reducing maintenance time;
- From Integrated to Componentized – organisations transforming from an Integrated to a Componentized maturity level would benefit in preparing themselves to expose the business functionality at more granular level, such exposure is required at more advanced maturity levels. The reusability also matures to be at a business function level as compared to application level and enhancements. New functionality is achieved through refactoring of the existing applications into smaller reusable components. The

disaggregation of the business in itself helps in reducing the complexities and facilitates the analysis of the impact of the componentized organization on new business models and business transformations. This componentization also helps the organization in reducing the time to market and increases IT response to business changes;

- From Componentized to Services – the transformation from a Componentized to a Service maturity level makes the organization be viewed more as a service provider to other organizations within the enterprise or external to the enterprise participating in the value chain. Business services now become reusable. This maturity level reduces the need for (and hence the cost of) redeveloping the same functionality for multiple systems by the provision of reusable business services called through a standardized interface, irrespective of the technology platform on which the application is running. These business services can also offer access to data in a controlled and timely manner that reduces inconsistencies in the data within systems that access and update the data. The investment of effort in service identification, specification, developing, testing and deploying a service is paid back when new systems require the same service from the providing organization, since the cost of infrastructure and maintenance of common functionality is reduced;
- From Services to Composite Services – organizations transforming from a Service to a Composite Service maturity

level have structured their business and IT support so that new business processes may be more rapidly constructed out of business services, and providing new business functionality to different parts of the organizations may be achieved more easily. This also reduces the time to market for a new business model due to a change in business strategy and or business transformation. At this level of transformation it is primarily a recomposition of the business services provided by different organizations within an enterprise of the value chain of the enterprise;

- From Services to Virtualized Services – organizations transforming from a Composite Service to a Virtualized Service maturity level benefit from a significant degree of flexibility in the design of integrated systems, in that different types of service (in terms of protocol, data models etc.) that would otherwise not be interoperable can be more easily integrated. In addition, systems may be reconfigured to achieve higher reliability, without the consumers having to modify their code. Virtualized services will enable organizations to better align business requirements with IT capabilities by building robust services that are highly flexible, manageable and scalable consistently;
- From Virtualized Services to Dynamically Re-configurable Services – organizations achieving this level of maturity would have completely decomposed services with service configuration information stored in a database for the service to be dynamically configured based on the dynamic nature

of service requests. This provides a superior flexibility for business transformation and provides a complete business and IT alignment. This provides autonomic features for the infrastructure to sense and respond to service requests within the organization and enterprise with high resiliency. Organizations at this level of maturity would have services that provide an agile capability to meet SLA's by allocating capacity dynamically with increased flexibility, which makes the organization highly competitive. This capability would also enable the Organization to optimize services for high availability and scalability without impacting service levels and reduces the complexity of deploying services.

11. PRODUCT PERSPECTIVE

When a product is developed for deployment in SOA-savvy environments there are several perspectives to be addressed during the product development:

- The service interactions for the organization using the product – this is the main reason why the product was purchased in the first place;
- The service interactions for the organization hosting the product – if deployment is based upon service hosting then the appropriate management and user administration features are required. This is particularly important if the host is supporting more than one client organization of the product;
- The service interactions that need to be maintained with the product development organization itself – remote user support and maintenance is increasingly important in quality of service provision and cost effective support. This has online access implications for the host or deploying organization;

It is useful to be able to express the capabilities of a product in terms of its SOA maturity implications. For example, a product must have the capabilities that at least sustain the current level of SOA maturity. This invariably requires the product to be easily managed thorough service interfaces and to provide key performance metrics as part of the management capability. Furthermore, it requires the product configuration to be easily

changed to reflect the specific needs of the organization as a whole and for each user in particular. As SOA becomes more broadly adopted, the associated sales and marketing literature will need to explain how the product sustains or improves the business operational capabilities in the terms of SOA. Again, using some commonly recognised SOA maturity model will provide excellent guidance on how to best express this information.

A product should be designed to be deployable in several ways. Apart from the common desktop application and server-based deployment approaches, third party hosting as software-as-a-service, or SAAS, must be addressed. Application service provision was an unsuccessful earlier form of SAAS but Web Services provide a better technology base which may allow SAAS to succeed. Also, improvements in networking technology and the Internet and Web in general make it easier for users to access such services.

12. SYSTEM PERSPECTIVE

For organizations that use SOA based systems, applications and tools there must be unambiguous benefits due to SOA. These benefits include:

- Provision of a planned and auditable strategy through which the business processes for the organization can be continually improved;
- A clear understanding of the Return On Investment (ROI) from each of the systems and their service components, and the associated business processes, within the organization. This includes both direct and indirect determination of ROI (remember that ROI is not the same as being a profit centre). Continual monitoring and evaluation is required because the ROI will vary as the organization's activities change. Also, as the core technology and the associated infrastructure ages re-investment will be required. As the ROI changes, the way in which a business process is performed will have to be altered e.g. when to detect that outsourcing becomes a preferable alternative;
- Support for the increasing corporate governance and compliance demands. This requires the various systems to be able to report key management information so that this can be collected, collated and used when and where required in the organization and enterprise. This will include distributed systems being able to retrieve their metrics in real-time and to

present the information in a variety of reporting formats. It is essential that common semantic definitions of the same data be used across the enterprise so that consistent interpretation and use of the information is possible.

13. TWO SIMPLE CASE STUDIES

We'll look at two simple case studies to describe how the OSIMM can be used to support an organizations exploitation of SOA. Each case study is an amalgamation of the experiences of several real companies.

The first case study is for a medium sized organization that has IT integration issues caused by a series of corporate acquisitions. Some of the acquisitions have business processes based upon SOA while others do not. The latter is typical when a small start-up is acquired and while they must have key products it, they are unlikely to have SOA-based corporate processes. Irrespective of the level of SOA maturity for each of the acquisitions there will be many issues to be addressed as part of providing a set of integrated business processes.

All of the organizations need to undergo an SOA maturity evaluation. The central IT team need to work with the IT teams in the acquired companies to create the agreed SOA maturity assessment questionnaire. Once this questionnaire has been completed the responses are analysed to look at points of commonality and difference. An integration strategy can then be developed depending on the primary business objectives determined at board level. This strategy will identify which business processes need to undergo enterprise-wide integration and which will either remain unchanged or become unnecessary. Once the strategy has been agreed, the detailed implementation plan is developed and executed. A further SOA maturity

assessment evaluation should be undertaken once the operational activities have been given sufficient time to adjust to the changes.

The second case study addresses a small product development company that supplies facilities management software. The company faces increasing competition from SAP who are selling a similar, but functionally less rich, product that is integrated with their back-end systems product range. The company is faced with developing new revenue models for its products, protecting its current client base consisting of small-to-medium sized enterprises that are being targeted by SAP and, finally, to establish new markets. SOA is being considered as the basis for renovating the product line. This is because:

- New revenue models based on pay-per-use through hosted provision is possible. Also, a services-based approach means that per function revenue charging is possible through access control to the service. Support and maintenance costs can also be reduced by using a Web-based approach;
- New large and medium size enterprise markets can be addressed by creating products that can support enterprise integration. Discussions with SAP to provide direct integration to their back-end system is undertaken thereby providing SAP with a better solution for facilities management and using SAP to act as a new channel to market.

The OSIMM is used to produce a simple analysis of the SOA capabilities of the companies that currently use the product range. A simple customer questionnaire on the usage of the products is created and within this are placed key questions that can be used to determine the SOA maturity of the customers with respect to the product. The customers should also be questioned about their plans for adoption of SOA-based products. Once the questionnaire has been completed by a typical representation of the client base (this will almost certainly require one-to-one work with each client) the responses are analysed to determine key indicators for the product and the need for SOA-savvy capabilities. This information is then used to inform the product development plan.

14. IN CONCLUSION

For many years, the software industry has advocated the use of maturity models. In the case of SOA, a maturity model is needed to enable organizations to identify the extent to which they either use or could use SOA. The same model can then be used to help organisations plan how to improve their use of SOA. A maturity model provides a systematic framework for carrying out benchmarking and performance improvement. The SOA equivalent model is focused on how SOA can be used to improve business processes using service-based implementation.

There are various maturity models available for SOA. Sonic Software and its partners jointly developed a simple maturity model that is designed to show the increasingly positive impact of SOA adoption from a business benefits perspective. It provides IT decision makers with a simple framework for benchmarking the strategic value of their SOA implementation, and a model for visualizing future success. This maturity model provides a framework, with five levels of maturity, for discussion between IT and business users about the applicability and benefits of SOA in an organization across five levels of adoption maturity.

A more sophisticated alternative is the Open Group' Service Integration Maturity Model (OSIMM) which contains both a framework and a process. As a framework, the OSIMM provides seven dimensions across seven maturity levels. OSIMM defines the organization in terms of a set of

“dimensions”, representing different views (e.g. business, architectural) of that organization. The seven dimensions are the: Business, Organization, Methods, Application, Architecture, Information and the Infrastructure. The seven maturity levels are: Silo, Integrated, Componentized, Services, Composite Services, Virtualized Services and Dynamically Re-Configurable Services.

Within each dimension the organization is modelled in more detail and each dimension is divided into several “domains”, each domain having a set of possible maturity indicators indicating the level of maturity of that domain. The maturity level of each domain can be aggregated into the maturity level of the dimension, and the total set of maturity levels for all the dimensions provides a holistic view of the service integration maturity level of the organization. As a process the OSIMM contains a simple methodology for ascertaining the level of maturity of an organization for each of the seven dimensions. This creates a maturity mapping that is used by the senior management team to identify which parts of the organization should undergo improvement.

It is useful to be able to express the capabilities of a product in terms of its SOA maturity implications. For example, a product must have the capabilities that at least sustain the current level of SOA maturity. This invariably requires the product to be easily managed thorough service interfaces and to provide key performance metrics as part of the management capability. Furthermore, it requires the product configuration to be easily

changed to reflect the specific needs of the organization as a whole and for each user in particular. As SOA becomes more broadly adopted, the associated sales and marketing literature will need to explain how the product sustains or improves the business operational capabilities in the terms of SOA. Again, using some commonly recognised SOA maturity model will provide excellent guidance on how to best express this information.

For organizations that use SOA based systems, applications and tools there must be unambiguous benefits due to SOA. The key benefit is that SOA can provide a clear understanding of the ROI from each of the systems, and their associated business processes, within the organization. Continual monitoring and evaluation is required because the ROI will vary as the organization's activities change.

Finally, if the evaluation of a maturity model highlights flaws that would cause problems for its adoption then change the model to suit your local needs. These models are business tools. Use them appropriately and tailor them to fit your organization's requirements.

APPENDIX A – BIBLIOGRAPHY

- [CMMI, 02] *Capability Maturity Model Integration (CMMI) for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Continuous Representation (CMMI-SE/SW/IPPD/SS, V1.1, Continuous*, CMMI Product Team, Software Engineering Institute, Technical Report CMU/SEI-2002-TR-011, 2002.
<http://www.sei.cmu.edu/publications/documents/02.reports/02tr011.html>
- [OSIMM, 06] *The Open Group Service Integration Maturity Model (OSIMM)*, Draft, The Open Group, 2006.

APPENDIX B – ACRONYMS

API	Application Programming Interface
BPEL	Business Process Execution Language
CMMI	Capability Maturity Model Integration
IT	Information Technology
OSIMM	Open Group's Service Integration Maturity Model
ROI	Return On Investment
SAAS	Software As A Service
SLA	Service Level Agreement
SOA	Service Oriented Architecture
XML	Extensible Markup Language
XSD	XML Schema Definition

INDEX

A

API 47

Application Programming Interface.
See API

Applications 20, 23

Architected services 10

Architecture 1, 13, 21, 22, 24, 44,
47

Assessment Methodology 28–31

B

Benefits of Increased Maturity
32–35

BPEL 17, 47

Business 1, 11, 13, 19, 22, 24, 26,
33, 44, 47

Business Process Architecture 22

Business Process Execution Lan-
guage. *See* BPEL

Business Process Management 22

Business services 11, 33

C

Capability Maturity Model 3, 46,
47

Capability Maturity Model Integra-
tion. *See* CMMI

Case Studies 4, 40–42

CMMI 8, 46, 47

Collaborative services 12

Componentized 13, 15, 32, 33, 44

Composite Service 33

D

Domains 22–25

Dynamically Re-configurable Ser-
vices 18, 34

E

Extensible Markup Language.
See XML

I

Information 1, 2, 13, 21, 24, 26, 44,
47

Information Technology. *See* IT

Infrastructure 13, 21, 25, 44

Initial services 10

Integrated level 14

IT 1, 5, 10, 14, 15, 16, 17, 19, 20,
21, 22, 23, 24, 25, 27, 28, 29,
30, 33, 34, 35, 40, 43, 47

M

Maturity Indicators 22–25
 Maturity Model Matrix 26–27
 Measured business services 12
 Methods 13, 20, 23, 44

O

Open Group’s Service Integration
 Maturity Model. *See* OSIMM
 Open Group’s SIMM 13
 Optimised business services 12
 Organisational Dimensions 19–21
 Organization 13, 19, 22, 26, 35, 44
 OSIMM 4, 5, 6, 7, 13, 14, 15, 20,
 22, 26, 27, 28, 29, 30, 31, 40,
 42, 43, 44, 46, 47
 OSIMM Dimensions
 Applications 20, 23, 48
 Architecture 1, 13, 21, 22, 24, 44, 47, 48,
 49
 Business 1, 11, 13, 19, 22, 24, 26, 33, 44,
 47
 Information 1, 2, 4, 13, 21, 24–25,
 26–27, 44–45, 47, 48–49
 Infrastructure 13, 21, 25, 44–45, 48–49
 Methods 13, 20, 23, 44, 49
 Organization 13, 19, 22, 26, 35, 44
 OSIMM Maturity Levels 4, 14–18

P

Product Perspective 36–37

R

Return On Investment. *See* ROI
 ROI 22, 38, 45, 47

S

SAAS 37, 47
 Service – composite 16
 Service Level Agreement. *See* SLA
 Service Oriented Architecture.
See SOA
 Silo level 14, 26
 Simple Maturity Model 10–12
 SLA 19, 35, 47
 SOA 1, 3, 4, 5, 6, 8, 9, 10, 11, 12,
 18, 19, 21, 23, 27, 28, 36, 37,
 38, 40, 41, 42, 43, 44, 45, 47
 SOA Maturity Models 1, 4, 8–9
 Software As A Service. *See* SAAS
 System Perspective 38–39
 V
 Virtualized Services 13, 17, 34, 44

W

Web Services 10, 11, 24, 37

X

XML 47

XML Schema Definition. *See* XSD

XSD 25, 47

This work was funded by Digital 20/20. Further information is available at <http://www.digital2020.org.uk>.

This booklet was written and produced by Colin and Christine Smythe of Dunelm Services Ltd (<http://www.dunelm.com>), at 34 Acorn Hill, Stannington, Sheffield, S6 6AW, Tel: 0114-2334009, E-mail: colin@dunelm.com.