Technical Standard

OSIMM – The Open Group Service Integration Maturity Model

THE Open GROUP
Making standards work®

It is fair use of this standard to use the terms, names, labels, concepts, etc. contained or described within it. The intent of publication of the standard is to encourage its widespread use.

This standard has not been verified for avoidance of possible third-party proprietary rights. In using this standard, usual procedures to ensure the respect of possible third-party intellectual property rights should be followed.

Document Type

Document Title

ISBN:

Document Number:

Published by The Open Group, <Month> <Year>.

Comments relating to the material contained in this document may be submitted to:

The Open Group
Thames Tower
37-45 Station Road
Reading
Berkshire, RG1 1LX
United Kingdom

or by electronic mail to:

ogedit@opengroup.org

Contents

Preface

The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX certification.

Further information on The Open Group is available at www.opengroup.org.

The Open Group has over 15 years' experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification.

More information is available at www.opengroup.org/certification.

The Open Group publishes a wide range of technical documentation, the main part of which is focused on development of Technical and Product Standards and Guides, but which also includes white papers, technical studies, branding and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/bookstore.

As with all *live* documents, Technical Standards and Specifications require revision to align with new developments and associated international standards. To distinguish between revised specifications which are fully backwards-compatible and those which are not:

- A new *Version* indicates there is no change to the definitive information contained in the previous publication of that title, but additions/extensions are included. As such, it *replaces* the previous publication.

- A new *Issue* indicates there is substantive change to the definitive information contained in the previous publication of that title, and there may also be additions/extensions. As such, both previous and new documents are maintained as current publications.

Readers should note that updates – in the form of Corrigenda – may apply to any publication. This information is published at www.opengroup.org/corrigenda.

This Document

This document is the Technical Standard for the Open Group Service Integration Maturity Model (OSIMM). It has been developed and approved by The Open Group.

Trademarks

Boundaryless Information Flow™ and TOGAF™ are trademarks and Making Standards Work®, The Open Group®, and UNIX® are registered trademarks of The Open Group in the United States and other countries.

The Open Group acknowledges that there may be other brand, company, and product names used in this document that may be covered by trademark protection and advises the reader to verify them independently.

Acknowledgements

The Open Group gratefully acknowledges the contribution of the following people in the development of this document:

- Ali Arsanjani, IBM
- Tony Carato, IBM
- Jorge Diaz, IBM
- Joe Hardman, IBM
- Kerrie Holley, IBM
- Dave Ings, IBM
- Geoffrey Meissner, IBM
- Tilak Mitra, IBM
- David Mott, IBM
- Andras Szakal, IBM
- Srinivasan Vembakkam, IBM
- ..
- Other working group members
- ..
- Members of The Open Group’s SOA Work Group

Referenced Documents

The following documents are referenced in this Document:

- The Service Integration Maturity Model and Process, Technique paper 2.1 , IBM Corporation
-

1 Introduction

1.1 Objective

This document is The ***O***pen Group's ***S***ervice ***I***ntegration ***M***aturity ***M***odel (OSIMM)) version 0.1. It specifies:

- A model against which the degree of service integration maturity of an organization can be assessed; and

- A process for assessing the current and desired degree of service integration maturity of an organization, using the model.

This version is provided, by IBM, as an input to the Open Group's OSIMM Working Group.

1.2 Overview

> *"The true value of Service Oriented Architecture is only achieved when Business Architecture is joined with IT Architecture. " [reference?]*

A Service-Oriented Architecture (SOA) is a set of business aligned IT services that collectively support an organization's business processes and goals. A service is a business task with an externalized service description that represents a contract between a Service Provider and Service Consumer.

As organizations are moving towards SOA and the use of services as the fundamental structuring element of their future state architecture, they increasingly encounter the need to assess where they are in this migration path to SOA and how to achieve greater benefits to support the organization, its business and systems.

The Open Group Service Integration Maturity Model (OSIMM) helps an organization create a roadmap for the incremental transformation of that organization towards more mature levels of service integration in order to achieve increasing business benefits associated with higher levels of maturity. OSIMM is used to determine which organizational characteristics are desirable in order to attain a new level of maturity. This will determine whether problems occurring at the current level can be solved by evolving to a higher level of service integration maturity.

OSIMM is offered to the industry as a standardized model for organizations to guide their SOA transformation journey. By having a standard maturity model, it becomes possible for the industry to benchmark their SOA levels, to have a roadmap for transformation to assist their planning and for vendors to offer services and software against these benchmarks. OSIMM may also serve as a framework for the transformation process that can be customized to suit the specific needs of organizations and assessments. This process is a simple sequence of steps:

configure the assessment framework, determine the initial level of maturity, determine the target level of maturity and a transformation path from initial to target level.

OSIMM structures the assessment of the organization’s current state in service integration and flexibility (including services orientation) and of their desired or future state for different lines of business or enterprise, taking account of pain points in flexibility or integration that need to be improved. It provides a model for assisting the organization in determining its architectural strategy when adopting service orientation, including the creation of an architectural roadmap for initiatives in legacy transformation, integration with one or more packaged applications, application renovation and development, and systems integration. This roadmap helps to determine the scope, focus and incremental steps for different parts of the organization in order to transform them towards a higher level of service orientation and service integration, with justifications in terms of anticipated business benefits. OSIMM provides a framework for surfacing insights and identifying IT improvements in terms of component development, service integration, SOA and IT governance.

OSIMM focuses on increasing levels of flexibility in seven aspects of an organization or enterprise: business, organization, methods and processes, application portfolio, architecture, information and infrastructure. Focus on these aspects aids the adoption of a more flexible business by planning integration in advance and constructing business models, processes, applications and infrastructure mindful of flexibility.

OSIMM may be used to conduct assessments of the current and desired levels of maturity for an enterprise or line of business within an organization and design a plan of action to transform from the current to the desired levels. For example, an organization may wish to apply OSIMM to a particular set of applications in the organization’s portfolio. A decision is made to partition the large number of applications into a small number of partitions, based upon affinity to business function. The current state of each partition is then assessed using the maturity model. Based upon the pain points, business drivers and goals, the target state for each partition is established. The transformation increment for each partition (which may be different for each partition) is then defined in order to achieve the target state for that partition.

1.3 Conformance

This specification describes a model and a process. It does not define what it means for an organization’s architecture to conform to the model. Rather, it defines what it means for an assessment to conform to the process.

An assessment that conforms to this specification must meet the mandatory provisions of Section 3.

1.4 Terminology

This glossary provides definition for terms that have a specialized meaning within OSIMM or are prone to alternative interpretations.

SOA Service-Oriented Architecture (SOA) is an architectural style that supports service orientation.

Service Orientation

is a way of thinking in terms of services and service-based development and the outcomes of services.

A Service:

- Is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit; provide weather data, consolidate drilling reports)
- Is self-contained
- May be composed of other services
- Is a “black box” to consumers of the service

An architectural style

is the combination of distinctive features in which architecture is performed or expressed.

The SOA architectural style has the following distinctive features:

- It is based on the design of the services – which mirror real-world business activities – comprising the enterprise (or inter-enterprise) business processes.
- Service representation utilizes business descriptions to provide context (i.e., business process, goal, rule, policy, service interface, and service component) and implements services using service orchestration.
- It places unique requirements on the infrastructure – it is recommended that implementations use open standards to realize interoperability and location transparency.
- Implementations are environment-specific – they are constrained or enabled by context and must be described within that context.
- It requires strong governance of service representation and implementation.
- It requires a “Litmus Test", which determines a “good service”.

Service Integration Maturity Level

An estimation of the degree to which an organization or enterprise has taken up the principles of SOA within their IT and business. There are 7 levels, level 1 being the least take up and level 7 being the greatest take up. Higher degrees of maturity are likely to lead to a higher degree of agility in the business, but are not necessarily “better”, as each organization may have an ideal level of maturity depending upon their business requirements and business and IT context.

Eco-system

A set of companies who are co-dependent on one another for achieving business goals by executing business processes that may leverage another company's business process.

Dimension (or View)

A major axis along which an organization may be measured as to its SOA maturity level. These represent significant views of the business and IT environment where the application of SOA principles can have a major effect. An organization may be at a different maturity level on each dimension, and the overall maturity level of the organization may be aggregated from each dimension's level. Dimensions are to a first approximation independent, but there are relationships between them.

Domain

A subdivision of a dimension, representing a more specific aspect of that dimension, along which the organization may be measured as to its SOA maturity level. Again these represent aspects where SOA principles can have an effect. Each domain has one or more maturity indicators at each maturity level, and the sequence of indicators identifies a pathway from less to more mature SOA. The overall maturity level of a dimension is aggregated from the individual maturity levels of each domain.

Maturity Indicator (or Characteristic)

A characteristic of the business or IT that may be measured and assessed by the asking of specific questions. Each maturity indicator is associated with a specific domain (and by implication a dimension) and maturity level; if the indicator is assessed as true then this is evidence for the domain being at that level of maturity.

Virtualized Service

A type of service that is hidden behind a "façade", so that the caller of the service does not call it directly but via a proxy that intercepts the call and routes it to a real service based upon considerations such as load and availability.

Dynamic Configuration

The ability of a system to look up new services, based upon the matching of a required specification, and to configure itself to call these new services without the development of new programming code.

Business Service

A self-contained piece of business functionality that may be called through a well-defined standard interface and protocol, independent of implementation platform, and managed under a contract specifying availability levels and quality of service.

BPEL Business Process Execution Language Standard. http://www-128.ibm.com/developerworks/library/specification/ws-bpel/

Transformation

Transformations define a high level change from one organizational state to another in order to support business imperatives and goals. Transformations may be business transformations (for example a reduction in the number of customer calls) or IT transformations (for example the introduction of support for markets in different geographies). It may be necessary to perform business and IT transformations in parallel in order to ensure that the business activities are aligned with the IT activities.

Adoption refers to the more detailed steps that are required to achieve the transformations. These steps may include the adoption of new technologies, methods, processes and integration techniques, and the establishing of corporate initiatives, IT directives, technical standards, Executive Councils, Architecture Boards and Governance.

Maturity is the creation of characteristics and behavior in an organization, as a result of transformation and adoption that permits them to operate better in accordance with their business goals. For example, an organization may have put in place processes for the identification of new services, which will facilitate the creation of services in the future. The nature of the characteristics and behavior created in the organization defines the service integration maturity level, and this is contained within the OSIMM model.

The concepts of transformation, adoption, and maturity are interrelated; transformations are broken down into adoptions, which create new characteristics: a sign of maturity.

Can Describes a permissible optional feature or behavior that an assessment may have.

Must Describes a feature or behavior that is mandatory for an assessment. An assessment that conforms to this document shall include this feature or behavior.

Should For an assessment that conforms to this document, describes a feature or behavior that is recommended but not mandatory.

Unspecified Describes the nature of a value or behavior not specified by this document that may vary among assessments that conform to this document.

1.5 Future Directions

TBD

2 The Model

2.1 Overview

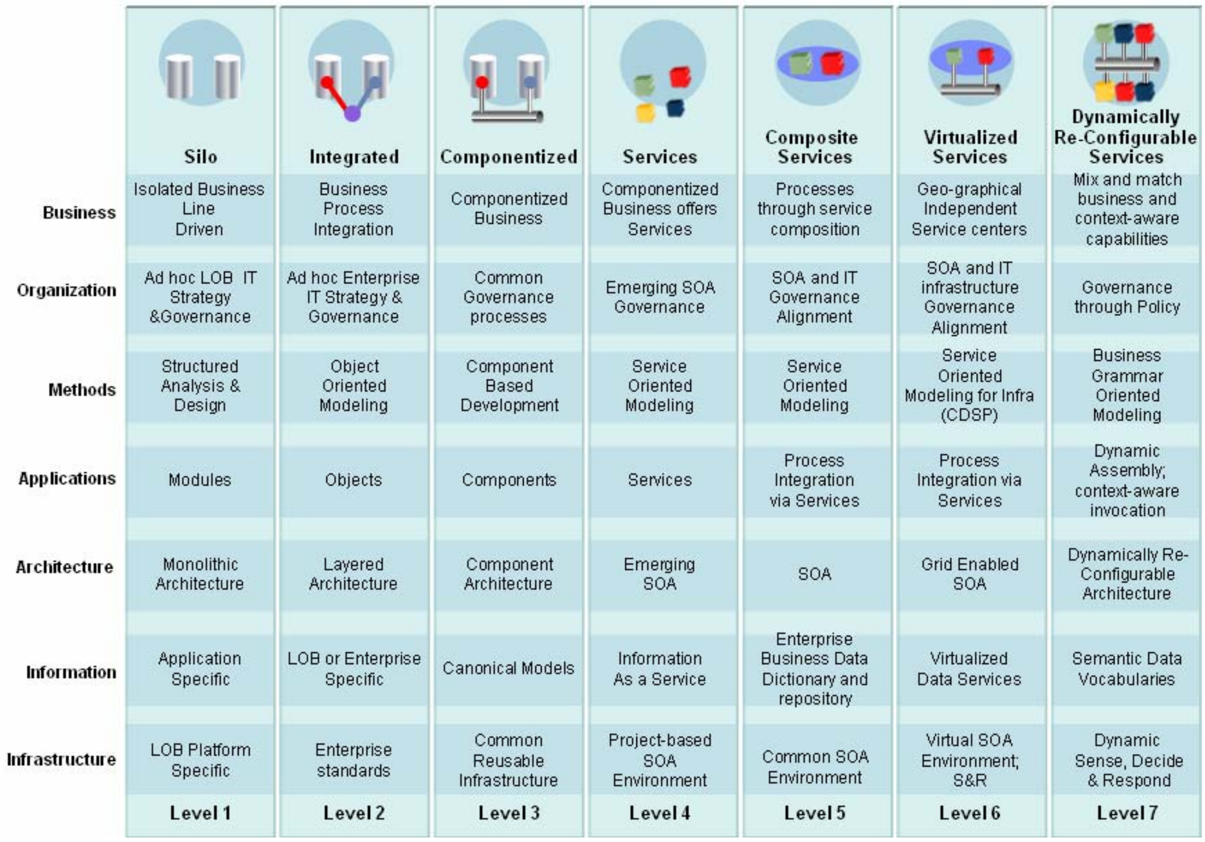
The OSIMM framework provides seven dimensions across seven maturity levels.

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 View, 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 modeled 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.

The complete matrix of dimensions and levels is shown in Figure 2:1.



	Silo	Integrated	Componentized	Services	Composite Services	Virtualized Services	Dynamically Re-Configurable Services
Business	Isolated Business Line Driven	Business Process Integration	Componentized Business	Componentized Business offers Services	Processes through service composition	Geo-graphical Independent Service centers	Mix and match business and context-aware capabilities
Organization	Ad hoc LOB IT Strategy &Governance	Ad hoc Enterprise IT Strategy & Governance	Common Governance processes	Emerging SOA Governance	SOA and IT Governance Alignment	SOA and IT infrastructure Governance Alignment	Governance through Policy
Methods	Structured Analysis & Design	Object Oriented Modeling	Component Based Development	Service Oriented Modeling	Service Oriented Modeling	Service Oriented Modeling for Infra (CDSP)	Business Grammar Oriented Modeling
Applications	Modules	Objects	Components	Services	Process Integration via Services	Process Integration via Services	Dynamic Assembly; context-aware invocation
Architecture	Monolithic Architecture	Layered Architecture	Component Architecture	Emerging SOA	SOA	Grid Enabled SOA	Dynamically Re-Configurable Architecture
Information	Application Specific	LOB or Enterprise Specific	Canonical Models	Information As a Service	Enterprise Business Data Dictionary and repository	Virtualized Data Services	Semantic Data Vocabularies
Infrastructure	LOB Platform Specific	Enterprise standards	Common Reusable Infrastructure	Project-based SOA Environment	Common SOA Environment	Virtual SOA Environment; S&R	Dynamic Sense, Decide & Respond
	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**	**Level 7**

Figure 2.1: OSIMM Overview

Each cell in the matrix corresponds to a summary of the maturity level for all the domains “underneath” the cell. For example, consider the cell Organization x Silo, with the label “Application Specific Skills”. The Organization 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 Organization Dimension for the organization is basically at the Silo level, and this is summarized by the label of “Application Specific Skills”.

Each Dimension may be assessed in a similar way, leading to a level assessment for each Dimension of Business View, Organization etc. The overall picture, in terms of the assessed maturity level for each Dimension may itself be assessed to provide a view of the overall maturity level of the organization.

2.2 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.

2.2.1 Silo

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.

2.2.2 Integrated

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

2.2.3 Componentized

The IT systems in the silos have been analyzed 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 "eco-system"), and causes difficulties in development of business processes that can be constructed across the parts of the organization.

2.2.4 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 underling 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 eco-system, and may be managed by assigning responsibilities for 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 analyzed 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.

2.2.5 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. Thus the design and development of business services is agile, and may be performed by developers under the close guidance of business analysts.

2.2.6 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.

2.2.7 Dynamically Re-Configurable Services

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.

2.3 Dimensions

An organization’s SOA or desired SOA scope can be assessed across the following complete set of dimensions.

2.3.1 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 maturity level.

2.3.2 Organization

The Organization dimension is focused on the structuring and design of organizations and resulting measures of organizational 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.

2.3.3 Methods

The Methods 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.

2.3.4 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.

2.3.5 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 artifacts produced.

2.3.6 Information

The Information dimension is focused on the information modeling 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.

2.3.7 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.

2.4 Domains and Indicators

The maturity indicators are assessed by a set of questions for each dimension. The standard set of questions is defined below, but it is anticipated that these questions may be added to as part of the customization of the assessment framework, at the start of an OSIMM assessment.

2.4.1 Business View

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 & up to date?
4. How is ROI measured in Business Process Management?
5. How agile are your current business processes?
6. What are the current funding practices?
7. What is the current cost model?
8. Who owns the portfolio of applications and services?
9. Do you have a cost model to charge service consumers for the use of the service?
10. How do you currently define TCO (including SW/HW & future maintenance)?
11. What level of partnership exists between the business stakeholders and the IT stakeholders?
12. How is the Business SLA measured currently?
13. What is the current practice to transform Business SLAs into IT SLAs?

2.4.2 Organization

1. What types of skills are common in your IT staff?
2. How does IT governance relate to your SOA?
3. How is the IT governance related or aligned with the corporate governance?
4. What are the governance functionalities and responsibilities?
5. How would you describe your IT cost model?
6. What type of SOA training is available in your IT organization?

7. How big is your development team and how is it organized?
8. How big is your Infrastructure team and how is it organized?
9. What is the relationship between you development team and the Infrastructure team?

2.4.3 Methods

1. What are the current requirements elicitations and requirement management practices?
2. What design methodologies and best practices are you currently adopting?
3. Do you practice any design techniques?
4. What design tools are in practice today?
5. What is the current practice for service development and management?
6. What is your current project management framework?
7. How is IT project management organized?
8. What is your current QA process?

2.4.4 Application

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?
6. How widely is XML used in your organization? How sophisticated is its use?

2.4.5 Architecture

1. How would you characterize your architectural topologies?
2. What type(s) of database do you use?
3. What is the standard communication style in your architecture?
4. How is integration achieved in your architecture?
5. Does your organization have or are you developing a Business Information Model to standardize data and message formats and concepts across the enterprise?
6. How mature are your Web services implementations?
7. How extensive is your SOA?
8. What architectural principles define your approach?

9. How extensive and sophisticated is your organization's use of frameworks in your architecture?

10. How are architectural decisions made in your organization?

11. Does your organization use reference architectures?

2.4.6 Information

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 from one application to another? For all applications? For only some applications?

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 modeling tools? By other non-computer-based modeling tools?

6. Are the Data Models in the form of Business Object Models, understandable to and owned by, the Business, or as IT object models, understandable only to, and owned by, the IT teams?

7. If there are mapping rules across different models, are these understandable to and maintained by the business or by IT staff?

8. Are such mapping rules performed by the infrastructure?

9. Are the data models defined by a language that includes taxonomies, ontologies, or other high-level logical representations?

10. Do you maintain data objects that are referenced by different programs?

11. Do you maintain a global directory or database of these objects, with global identifiers? Or do you have mechanisms for mapping these IDs between different databases/directories? Are these mechanisms electronic or manual? Are all such objects mapped, or is this done only for certain applications and sets of objects? Are these mappings undertaken automatically by the infrastructure?

12. Do you have standards for representing these IDs in calls to services?

13. Do you have standards for representing these IDs in business processes defined by flow languages?

14. Do you have mechanisms for looking up global objects by a search on their characteristics?

15. Are the security credentials (e.g. user name, password) different for different applications? Are they shared across some applications? Are they shared across all applications?

16. Do you pass security credentials in service calls? Are these represented according to an open standard? Is there a security infrastructure to support these standards and service calls?
17. Do the business services provide definitions for failures of access and other security issues?
18. Is there need for transformation of data from one application to another? If so is this done by IT or manually (e.g. by multiple keying of data)?
19. If by IT, is this done by bespoke adapters as required? Or via a comprehensive set of APIs? Or via services?
20. Do the services use bespoke code to perform mapping or is there a standard infrastructure (e.g. an ESB) providing general mechanisms for performing mapping, that are driven by meta-data defining the mappings?
21. Are there facilities for performing complex inference in order to map data defined in ontologies from one form to another?
22. Are there well defined, unambiguous, specifications of components?
23. Are these definitions in a standard, programmer neutral form, as services, permitting their use from different platforms?
24. Are these definitions stored and searchable electronically? Via a standard interface?
25. Are business processes defined as services? Is this in the same form as the basic services above?
26. Is the addressing of the services via a virtual addressing mechanism, permitting the change of the service implementation without affecting the consumers?
27. Is there a rich definition language for the service definitions, permitting the lookup of the services based on the purpose, effects and actions of the service?
28. Are business processes defined in any formal manner? Are all processes defined, or just those as required?
29. Are there any common methods for representing business processes?
30. Are these definitions understandable by business or IT or both?
31. Are these definitions using an open standard or via a bespoke representation?
32. Are there transformations and mappings performed by the infrastructure that may be used as part of the business process flows?
33. Is an explicit flow language used? Or is the language based upon goals, actions and their effects?
34. Is data accessed in a bespoke manner different for each application?
35. Are there some common mechanisms for accessing data, as required?
36. Is all data accessed in a common manner?

37. Is this access defined via program level APIs or via services?

38. Is transactional support offered for data access and update? Are mechanisms in place for "locking/reserving" data across long running processes

39. Are there virtual data access mechanisms based upon meta data describing the information?

40. Are the data access services set up so that they may participate in business flows?

41. Is there a facility to query repositories of information represented by ontologies?

42. Do you perform data analysis?

43. Is this performed by bespoke code as required with individual metrics?

44. Is there a common set of metrics?

45. Are these analyzed by common code?

46. Are the results integrated across different parts of the organization?

47. Is the common code defined as services? Are there industry standards for the analysis and are they used?

48. Are there metrics for business process level data?

49. Are there analytic services that can be used in the assembly of business processes? Are these performed by the infrastructure?

50. Are the metrics defined in terms of the objects defined in ontologies?

2.4.7 Infrastructure

1. What are your current infrastructure usage guidelines?

2. How are your IT SLAs transformed from the business SLAs?

3. Have you defined SLAs around Quality of Service? How is this monitored and measured?

4. Have you defined any SLAs around security and privacy? How is this measured and monitored?

5. What level of monitoring is in place today? What management tools are in place today?

6. What platforms are currently in use for integration?

7. Which objects are placed under version control?

8. What is your current change management process?

9. What tools are used for configuration management?

10. What are considered as your organization's IT assets? (excluding human resource) How are these assets managed?

11. What does your current operational architecture look like?

12. How does your operational architecture support the NFRs for applications and services?

3 The Assessment Method

OSIMM may be 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 eco-system.

3.1 Overview

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 Services Oriented Architecture
- 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 Services Oriented Architecture
- production of the recommendation report which 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.

These activities are performed in an OSIMM analysis. An OSIMM analysis must be conducted using the following steps.

- The business identifies the pain points
- The pain points focus the assessment framework and maturity indicators to those relevant to the pain points.
- The assessor determines the current maturity level by comparing the current state of the organization against the maturity indicators.
- The reasons for the pain points are determined by considering the maturity indicators and their effects
- By considering the characteristics of the maturity indicators in higher levels, consideration is given to how the pain point reasons may be alleviated, and which target level is suitable.
- By comparing the current and target level maturity indicators, the steps in the roadmap from current to target may be determined.

The logic behind these steps is illustrated in Figure 3:1.

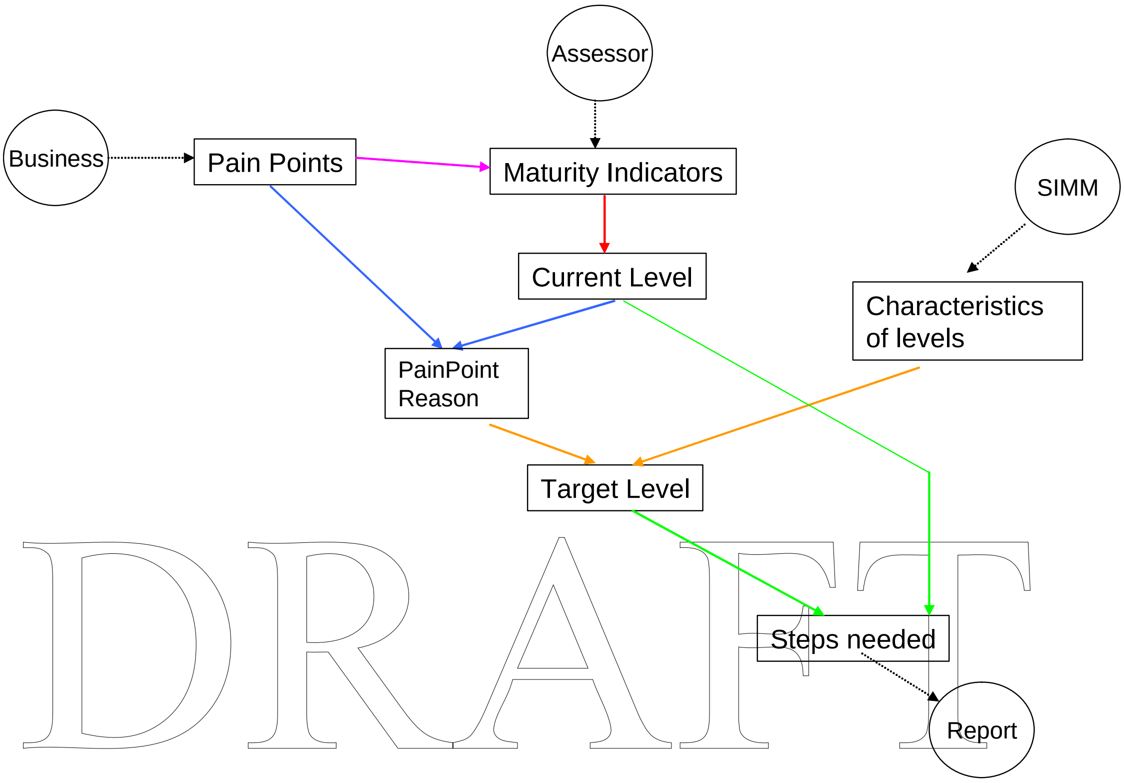


Figure 3:1 – OSIMM Analysis Logic

The steps are described in the following subsections.

3.2 Identify the Pain Points and Define Scope

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.

3.3 Configure Assessment Framework

On the basis of the agreed scope, an assessment matrix must be created, based upon the full OSIMM matrices, but tailored to focus on the key pain points. The OSIMM practitioner can select from the existing set of dimensions, domains, maturity indicators and questions, or can develop new indicators or questions as appropriate.

3.4 Assess Current State

Using the assessment matrix developed in the previous step, the OSIMM practitioner must interview key staff from the organization in order 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. The standard set of questions is provided in section 5. 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. In addition, 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.

3.5 Determine Future State

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.

3.6 Identify the Gaps and Determine the Road map

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.

3.7 Develop the final report

The conclusions of the OSIMM assessment, including pain points, assessment matrix, current maturity level, future maturity level, alleviation of pain points and road map, must be presented as a report to the organization. This can be used to guide the next stage of analysis and planning.

A Appendix A – Example Assessment

A.1 Business Objective

HEALTHCO, a company providing health care services, envisioned a Service-Oriented Architecture to drive integration, promote a common business and IT vision, and optimize IT spending to support HEALTHCO's business goals. To accomplish this vision, HEALTHCO needed to identify gaps in its current IT environment from the service integration maturity perspective. OSIMM was used to assess the current state, determine the target state and develop recommendations across the OSIMM dimensions.

A.2 Analysis

In the example, a number of applications were divided into two groups, front end and legacy, and an OSIMM Analysis was performed. The steps, focusing on the Business dimension, are summarized below, and illustrated in Figure A:1.

- A pain point is that the business perceive the IT as not being agile enough to support the introduction of new business processes
- By analyzing the maturity indicators, it is determined that the business see IT as applications rather than composite business services that can be created from other business services.
- This places the organization currently at level 2 on the business dimension
- The reason for the pain point is that applications are monolithic and are not easily composable from business components
- By considering the characteristics of different levels as defined in OSIMM, it can be seen that business at level 5 will alleviate the pain point reason by facilitating the design of new business processes from business services
- The need to go from level 2 to (at least) level 5 in the business dimension suggests a step in the road map of introducing business processes and business services to structure the functionality.

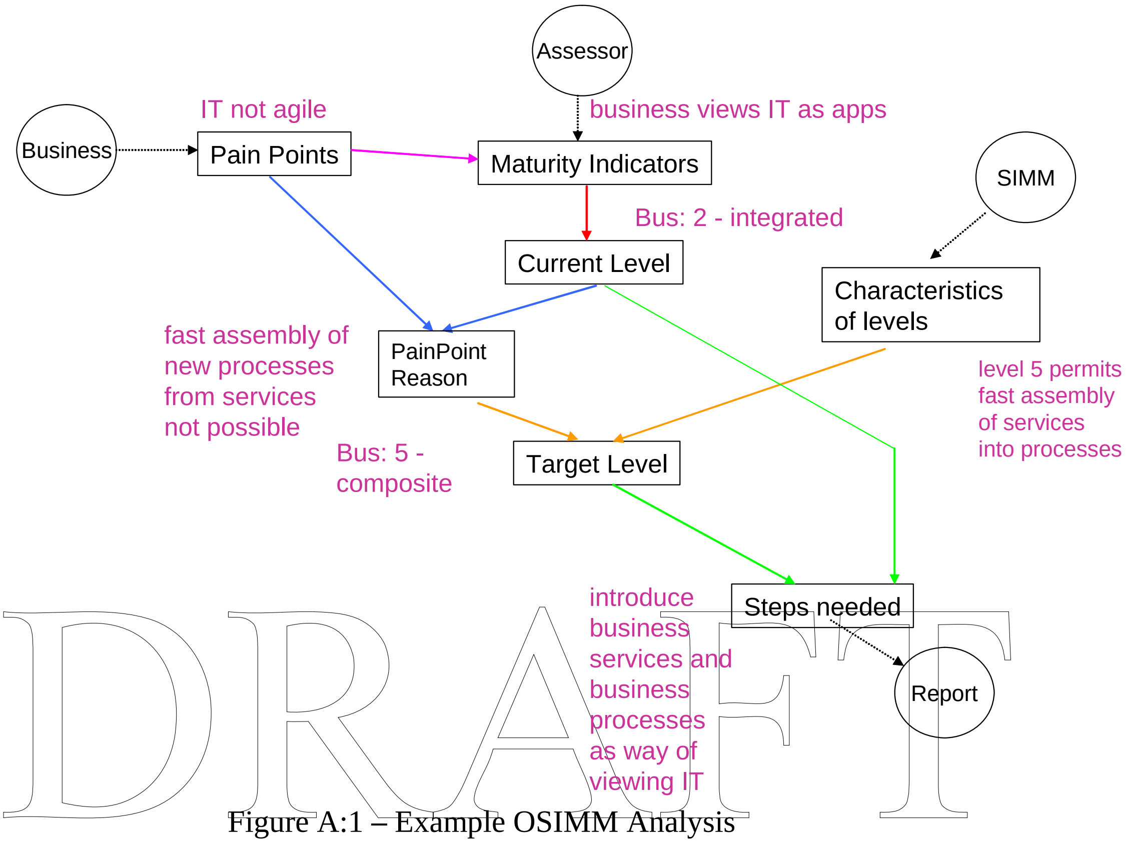

Assessor
IT not agile
business views IT as apps
Business
Pain Points
Maturity Indicators
SIMM
Bus: 2 - integrated
Current Level
Characteristics of levels
fast assembly of new processes from services not possible
PainPoint Reason
level 5 permits fast assembly of services into processes
Bus: 5 - composite
Target Level
introduce business services and business processes as way of viewing IT
Steps needed
Report

Figure A:1 – Example OSIMM Analysis

A.3 Recommendations

The recommendations are summarized in the following table, together with the current and target maturity levels for each of the dimensions:

OSIMM Dimension	**Current Maturity Level**	**Summarized Assessment**	**Target Maturity Level**	**Recommendations**
Business View	2	Strengths: • Business has good understanding of IT capabilities. Weaknesses: • Business views IT as a set of applications that deliver capabilities to support business processes. • Business capabilities are not aligned with	6	A componentized view of the business capabilities should be developed in which business views services as assets that transcend the current application-centric views.

		IT. • Application interdependencies and complexities affect business agility.		
Organization	3	Strengths: • Cross-application organization is in place. • Responsibility for service enablement is managed Weaknesses: • The IT organization is mostly application-centric with specialized skills to manage the development and support for specific applications.	4	Business owners should drive changes to business services, business processes, and the component architecture to meet changing business needs. IT owners should be assigned to support specific business capability areas and their business owners. Business capability owners should be enabled to focus more on sustaining and improving specific capabilities.
Methods	2	Strengths: • IT planning and governance process in place. • Consistent Development methodology followed. • OO design and development practices are in place for front-end applications • Services standards and guidelines are published. Weaknesses: • Planning and development process does not support services modeling or code reuse, with limited support for business process modeling • Planning and development process is heavyweight and	4	Enhance planning and development methods to support services identification, design and development. Introduce services governance process. Enhance planning and development processes to encourage and promote code reuse. Enhance planning and development processes to support iterative development. Enhance the software development method to support Business Process Modeling and implementation.

		waterfall-based.		
Infrastructure	3	Strengths: • System management software is in place. • Security infrastructure is in place. Weaknesses: • SOA-specific infrastructure (Services management, Business process management) is absent.	5	Deploy web services management infrastructure to support enterprise-scale SOA deployment. Deploy Business Process Management infrastructure. Deploy SOA security infrastructure to be able to support security policies defined at the service level.
Applications (Front End)	3	Strengths: • Architecture is componentized and layered • Object models used Weaknesses: • Minimal code reuse. • Object models not shared and are developed independently • BPM/Workflow is custom or not in place. • Application architecture is not standardized.	5	Implement enterprise domain object model. Introduce code reuse at the component and library level. Standardize reference application architecture, design patterns and best practices. Implement business rules engine. Modernize and componentize COBOL applications.

Applications (Legacy)	2	Strengths: • Efforts are in place to modernize the application architecture. • Legacy Access views provide a consistent approach for code reuse. Weaknesses: • COBOL legacy architecture difficult to change. • No consistent approach for system componentization. • BPM/Workflow is custom or not in place. • Application architecture is not standardized and does not address back-end applications.		
Architecture Integration and Services (Front End)	3	Strengths: • Most applications consume Legacy access views using standard approach. • Some applications act as service providers. • WSDL files published within each application. Weaknesses: • Point-to-point integration. • Different protocols and translation mechanisms used for mainframe integration. • Security architecture is inconsistent	5	Implement reusable business services. Implement Enterprise Integration Data Model (Canonical Data Model). Implement uniform transport protocol for Web services. All communications with internal and external systems should be handled by ESB Support legacy consumers using ESB. Implement some of the application components as coarse-grained service components where component's interfaces are exposed using Web services. All applications, including

Architecture Integration and Services (Legacy)	3	Strengths: • Legacy Access views are used to provide services to other applications and are documented • An approach is in place to make the Legacy Access views available to consuming systems. • ESB implemented. Weaknesses: • Back-end systems tightly coupled. • Some Legacy Access views not generic. • No enterprise data model for system integration. • Business functions are duplicated in multiple systems. • Heavy reliance on batch feeds. • Security architecture is inconsistent		mainframe back-end systems, communicate via Web services as opposed to reusing copybooks and Legacy Access views directly. COBOL applications should be able to act as consumers of Web services provided by other back-end systems. SOA must provide the support for batch processing; batch processing should be implemented "on the side." Design and implement security policies at the service level.

B APPENDIX B – Benefits of Moving to Higher Maturity Levels

B.1 Benefits of moving from Silo to Integrated

Organizations transforming from a Silo maturity level to an Integrated maturity level will significantly reduce operational and maintenance cost. 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 OO 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.

B.2 Benefits of moving from Integrated to Componentized

Organizations transforming from an Integrated maturity level 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 and new functionality is achieved through refactoring of the existing applications into smaller re-useable 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.

B.3 Benefits of moving from Componentized to Services

The transformation from a Componentized maturity level 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 re-useable business services called through a standardised 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 which 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.

B.4 Benefits of moving from Services to Composite Services

Organizations transforming from a Service maturity 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 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.

B.5 Benefits of moving from Services to Virtualized Services

Organizations transforming from a Composite Service maturity 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, system 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.

B.6 Benefits of moving 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.

Index