The Practical Guide for SOA in Health Care
A real-world approach to planning, designing, and deploying SOA.

Version 1.0

An informative reference for leaders and decision-makers. Produced by the Healthcare Services Specification Project (HSSP): A collaborative effort between Health Level Seven (HL7) and the Object Management Group.

This document is an informative reference and it is not, nor is it intended to be, an industry standard. Free distribution and replication of this document is permitted, as is the reuse of content when appropriately attributed. The content of this document was collected during standards working meeting and is copyright to the Healthcare Services Specification Project, Health Level Seven, and the Object Management Group. All Rights Reserved.
This page intentionally left blank
Executive Summary

Across the globe, healthcare organisations are increasingly making IT investments to improve their operations, efficiency, more effectively manage costs, and improve their operational capabilities. IT shops can be hampered in their efforts to evolve due to extensive existing investments in hardware, software, and medical devices that they must continue to support, while coming under increasing pressure to modernise systems. The need to change is pushed by technologies are constantly evolving. IT planning processes are also complicated by “industry recommended practices” that are hyped by the technology journals.

One such area receiving significant amounts of this hype is service-oriented architecture (SOA). In a nutshell, SOA provides an approach for business transformation based on dividing complex environments into well defined, formally specified functions based on the activities they perform (services). Each service has well defined responsibilities and authority. These services then work together in collaboration to support the workflow of the business, all within the context of governance and oversight that manages their coordination and performance. It is being touted as everything from “the next silver bullet” to a technology platform to an enterprise change management strategy.

Effective SOA programmes involve cooperation and coordination among a wide variety of participants from across an organisation. Whether senior management sponsorship, business community ownership, program management governance, or project level execution and sustainment, SOA involves a broad variety of participants all of whom must actively engage to realise organisational success.

Why SOA? Why Act Now?

Among the most difficult challenges facing healthcare organisations making IT investments today comes from deciding whether to go “all-in” with any particular vendor, or whether to self-integrate components from multiple vendors. The appeal of the single-vendor solution is strong – no “finger-pointing”, out-of-the-box integration, [US-based] EHR certification via the Certification Commission for Healthcare IT (CCHIT), and so on. This is contrasted with seemingly increased risk and work involved in a multi-vendor situation and involving integration. The tradeoff is that a multi-vendor solution offers best-of-breed options, and a SOA promotes the easy integration and alignment across suppliers into a cohesive architecture.

In a nutshell, there are a few fundamental truths to consider to get a clear picture of your return-on-investment (ROI):

1. **Legacy systems comprise a significant investment.** Legacy systems make up the core of any business’ portfolio. While some of these systems may require replacement that may not be true for the entire portfolio. In many cases, there is either too little benefit or too much cost in

---

1 The term platform is frequently used and means different things. In this context, platform represents a software layer in a technology stack which is the manifestation of an architecture. In other words, while SOA itself is an architecture, the SOA platform comprises all of the software that implements that architecture.
replacing these systems. As a result, legacy systems will continue to retain data that is too expensive or “dirty” to migrate.

**No single vendor is best-of-breed at everything.** The reality is that heterogeneous environments are here to stay. Whether required by a clinical subspecialty or mandated by government authorities, integration needs with unexpected systems and platforms will always exist.

**The need to exchange health information across organisational boundaries is growing.** Information sharing is happening between healthcare institutions that was unforeseen only a few years ago (e.g., shared care delivery, public health reporting, chronic disease management, biosurveillance). The need to expose access to service capabilities is being increasingly driven by patient-directed and community-based care.

**Clinical medicine, workflow, and regulatory environment are ever-changing.** The rules are changing, faster all the time, placing demands on healthcare organisations to more quickly react and support the needs of an evolving landscape, fostering clinical quality and consistency.

SOA provides some unique abilities to more quickly react, adapt, and institute changes within an organisation and its IT landscape due to the modular structure of services and ability to alter interactions among those service components. SOA is a proven architectural approach that is mature in many other market sectors and has shown benefit it healthcare organisations as well. Ultimately, organisations that have elected to utilise SOA solutions have done so to improve their agility (ability to respond to changing requirements), to more effectively develop and deploy IT systems, and to improve business ownership, accountability, and consistency.

**Principal Findings**

Although SOA is a relatively new approach within the healthcare sector, it is an established, proven, and reliable approach that has realised business benefit in other industries. **SOA is an architectural style (as is client/server, hub-and-spoke, etc.)** that may be realised in a number of implementation forms. That said, do not confuse SOA with web-services, which is technology platform. Just because something is implemented using web services does NOT make it a SOA.

SOA provides some unique abilities to more quickly react, adapt, and institute changes within an organisation and its IT landscape. SOA is a proven architectural approach that is mature in many other market sectors and has shown benefit it healthcare organisations as well.

One challenge that few organisations realise in time is that SOA does not itself guarantee interoperability, nor do web services. Tooling in the marketplace has improved, resulting in improved ability for organisations to build new “stovepipe systems” that do not necessarily interoperate. Overall, SOA successes share several core tenets, summarized below:

- Based in Enterprise Architecture, fostering IT alignment with business needs
- Supported by the business community under the authority of an Executive Sponsor as SOA often alters business responsibilities and organisational boundaries. This includes authority and governance to manage the overall programme.
Encompassing of legacy systems, breaking up what had been large, monolithic applications into service capabilities, and fostering their co-existence with current infrastructure and investments

Fosters discrete components with well defined responsibilities and interfaces, establishing authoritative ownership within the business and IT landscape.

Orthogonal to existing systems view, where focus is on identifying shared need across systems and organisations (e.g. not 1:1 between SOA services and department systems)

The figure (right) illustrates a macro-view of a SOA. It depicts five overlapping boundaries, each of which represents a context within the business and architecture:

- The **Inter-organisational Boundary** (outermost) represents inter-organisational considerations, such as policies, sharing agreements, and business partners.

- The **System Boundary** represents the physical platforms on which software and systems run, including servers, networks, and so on.

- The **Application Boundary** represents the software running on those platforms, inclusive of applications and data.

- The **Business Process / Orchestration Boundary** manages the intersection between software and workflow, and would manage coordination among multiple software components that all must interact to satisfy business needs.

- Finally, the **Service Implementation Boundary** depicts the implementations themselves, interacting across a service bus, and realising the architecture.

This diagram, more richly elaborated later in the document, forms the basis to demonstrate how a SOA can co-exist with legacy and commercial applications within an organisation.

---

Today, it is prevalent to have dedicated systems devoted to a particular functional line. Slowly, the marketplace has been identifying areas of shared service need and providing SOA capabilities to address them. This shift has been occurring, driven by purchasers whom are themselves implanting SOA.
Approaches to SOA

There are many viable methods for approaching SOA implementations. Some organisations elect to follow a “top-down” method, functionally breaking-down the business processes to identify the common services needed across the organisation. Other organisations approach SOA more opportunistically, starting from one easily-identified service need and growing the architecture from there. Finally, SOA-enabling “legacy” applications is one way to further the useful life of existing IT investments while taking a step into the SOA world.

There is no right or wrong approach, despite industry literature that may indicate otherwise. The right decision for your circumstance will likely depend on your own organisational culture, willingness to embrace change, planning and budgetary cycles, and existing investments. Throughout this document, we have centered the discussion around the “top down” approach as it is a bit more systematic and comprehensive than the alternatives. That said, a hybrid approach employing a top-down method coupled with opportunistic bottom-up instances that can grow organically is an approach many of the authors have used within our own organisations with success.

Health Industry SOA Standards

In an effort to help promote alignment among SOA implementations, and to further interoperability among organisations as they seek to realise SOA-based architectures, the Healthcare Services Specification Project (HSSP) was formed. This effort is a joint collaboration among standards groups--specifically Health Level Seven (HL7) and the Object Management Group (OMG)—developing health industry SOA standards. The intent of HSSP is to produce standard services that define services’ responsibilities, behavior, and interfaces so that ubiquity can be achieved across implementations and vendor products. HSSP services align with the architectural diagram above, and have a role in assuring that SOA’s for healthcare organisations are interoperable and standards-based.
Executive Summary............................................................................................................................i
Principal Findings...........................................................................................................................ii

1 Overview ........................................................................................................................................1
2 Document Scope ..........................................................................................................................1
  2.1 Document “Tour” and Structure.............................................................................................2
  2.2 How to use this document. .......................................................................................................2
  2.3 SOA-101.................................................................................................................................3
  2.4 Healthcare SOA Standards and HSSP....................................................................................3
  2.5 Healthcare-specific SOA Challenges.....................................................................................4
  2.6 Core Principles.......................................................................................................................5
3 Establishing a Healthcare SOA ...................................................................................................7
  3.1 Business case for SOA ..........................................................................................................7
  3.2 Introducing SampleHealth .....................................................................................................8
    3.2.1 Step 1: Enterprise Architecture .......................................................................................8
    3.2.2 Step 2: Define and Analyse As-Is State .........................................................................10
    3.2.3 Step 3: Identify Candidate Services ...............................................................................10
    3.2.4 Step 4: Define Future State ............................................................................................12
    3.2.5 Step 5: Specify Architecture and Services .....................................................................15
    3.2.6 Step 6: Build A Transition Plan .......................................................................................18
    3.2.7 Step 7: Realise the SOA ................................................................................................19
      Step 8: Deploy and Sustain ......................................................................................................21
  3.3 Lessons-Learned ....................................................................................................................23
    3.3.1 Design Principles .............................................................................................................24
    3.3.2 Enabling Legacy ..............................................................................................................25
    3.3.3 “Localising” to Your Organisation .................................................................................26
4 Acronym List ................................................................................................................................28
5 Glossary ........................................................................................................................................28
6 Acknowledgements ..................................................................................................................28
7 References ......................................................................................................................................29
1 Overview

This document considers service-oriented Architecture (SOA) from the perspective of a healthcare organisation considering a major information technology (IT) investment, recognising the value of existing systems, and mindful of the importance to demonstrate return-on-investment. If you are asking yourself how SOA fits within your organisation, whether healthcare SOA is any different from other industries, or how to approach a SOA implementation, then this document is for you.

Service-oriented architecture has received increasing interest and coverage from industry literature as the “next-best-thing” for organisations facing integration and modernisation challenges. Though these claims are clearly exaggerated, SOA has reached a maturity in the marketplace and is demonstrating business value in many market sectors.

Healthcare traditionally has been an IT laggard, making fewer investments than in comparable industries. That said, there has been tremendous interest in SOA within the health community, and it is increasingly garnering attention from all types of organisations, from governments, to healthcare providers to payers to vendors.

This document speaks to a business community, IT leadership community, and an IT application architecture/enterprise architecture community, and is not intended to be a detailed technical manual. It provides higher-level guidance outlining an approach for being successful using SOA, calling out industry best-practices and key factors for consideration in SOA implementation.

2 Document Scope

This document is being provided as a “jump-start” to help you and your organisation be more successful in your efforts to apply SOA. This document does not address every specialty or nuance that you are likely to face, instead focusing on the mainstream of how SOA may be used to address typical challenges faced in health care organisations and environments.

This document has a few simple objectives:

- To present, in plain English, the business case, context, and an approach for applying SOA into a healthcare organisation
- To provide a basic approach to use SOA, distilling many methodologies and industry-best-practices into a simple, concise process.
- To illustrate, by example, how healthcare industry SOA standards (such as HSSP standards) may be used within an organisation’s IT architecture.

To address these matters, the community elected to produce this informative guide, providing contextual information and implementation guidance without expectation of compliance. In other words, feel free to use the advice in this document that you find useful, and toss the bits that are not. Caveat emptor. The following will help you determine if this guide is applicable to your needs.

This Practitioner’s Guide is NOT…

- …a normative standard
- …detailed or granular enough to be an implementation guide on its own
- …a how-to guide for constructing profiles that are aligned to specific frameworks, data models, documents etc
- …an architecture to build a specific, given service (e.g., product architecture)
- …a methodology to support standards development
The stark reality is that architectures are best realised when business-driven, and the key design principles and environmental factors affect how that architecture is manifested. It is for this reason that this document is a “Practical Guide” and not a formal Healthcare SOA Standard. By creating the document in this light, we hope to convey many of the best-practices, lessons-learned, and community experiences to you in a coherent, cohesive example that you can adapt to your situation.

We hope that you find the document useful, and welcome your feedback.

2.1 Document “Tour” and Structure

This document is written to the mainstream 80%, intentionally omitting nuance exceptions and peculiarities that inevitably occur. This was done to keep the document easy to understand and very broadly applicable. To facilitate this discussion, an imaginary healthcare institution – SampleHealth—was invented. This provides a basis for discussion which includes a breakdown of their business functions, their existing architecture, legacy systems, and proposed new SOA.

The examples included in the document happen to be based on the United States, but the authors believe that none of the content would preclude its applicability to other nations or health systems. In fact, the authoring community contained several international participants.

This document considers SOA from two dimensions, each with a dedicated volume in the text. The static view [Volume I] represents the SOA services, their component parts, and how those pieces physically and logically relate [via dependency, topology, physical infrastructure, and so on]. This section discusses how to subdivide your problem space, identify SOA services of interest, specify interfaces, and so on.

Volume I of the Practical Guide outlines a nine-step plan that is an informative reference. This plan is based on accepted industry practices, and calls out key activities that are typically performed in successful SOA implementations. This plan is not the only way to apply SOA, nor is it a standard way of doing so. The guide is intended as a desk-reference, to be custom-tailored to your specific situation.

The dynamic view [Volume II] considers how the services interact, particularly when supporting the fulfillment of some business need. This section elaborates several scenarios [use-cases], illustrating how services coordinate in support of workflow [choreography, service orchestration].

2.2 How to use this document.

Recognise that this is a ‘practical guide’ and not an authoritative reference. The challenge we faced as authors was to consider the tradeoff between keeping things simple and useful versus the extensive depth that would be needed to address these concerns. We opted for useful.

As a result, we believe that many of the challenges that you are likely to face are addressed in these pages, but some of your situationally-specific needs might not be. We encourage you to extend this guide to make it your own. You are free to do so, and we only ask for attribution.

Finally, the document is structured around a set of core underlying principles, which are explained, followed by illustrative examples where those principles have been applied to an example situation to make the abstract concepts much more tangible. Volume I of this document addresses system and architectural structure from a “static” and topological view, focusing on how the pieces fit together.
This guide may be used:

- To provide an example of a SOA-based Enterprise Architecture in a health context using industry standard services
- To identify representative touch-points with existing standards and technologies (e.g., platforms)
- To review a suggested “minimum essential set” of layers and services needed for a successful SOA
- To provide helpful information to assist in localising / customising standard services to meet (your) needs
- To provide a framework setting the context in which future service planning (e.g., Roadmap) can occur

Volume II addresses system behavior and interactions, focusing on how the pieces work together (the system “dynamics”).

### 2.3 SOA-101

There is a plethora of industry analysis available which makes the distinction between services and “SOA services”, or at least good SOA services. Good SOA services are well designed and well defined, and achieve an appropriate balance of granularity, coupling, reusability, flexibility, simplicity and usability. Many of these trade-offs are an art rather than a science.

One can argue about what makes SOA different from some architectural styles of the past. Some will say that SOA concepts are not new, and may well be right. Perhaps the “new” piece of the story is the emphasis on standards based interfaces. To take it a step further, standards based interfaces that are well designed are a critical success factor for SOA.

A service, simply put, is comprised of an interface and an implementation. The interface outwardly defines how one interacts with the service, and the implementation is the coding that realises the expected behavior of the service. Put a bit more formally, service interfaces describe how the various actors in a SOA ecosystem interact.

Service interfaces utilise behavioral models (operations or actions ) and information models (structure, and semantics). Thus, information models that are also standardised and relevant are essential. Likewise, behavioral models that reflect business processes and business functions are also critical.

Ultimately, SOA is realised in the components from which it is comprised. A component is the piece of software that is actually running on a server. It is the implementation of the service that has been specified, exposing itself to its potential customers by means of a component (or service) interface. Within the component is the code to perform all of the functional capabilities it has. Components interact with other components, and can pass along work, collaborating to satisfy complex business needs.

### 2.4 Healthcare SOA Standards and HSSP

Standardised interfaces make it easier than in the past for disparate systems, with different programming languages, operating systems, and databases, to communicate with each other. SOA ecosystems can exist within an organisation as well as inter-organisational.

The Healthcare Services Specification Project (HSSP) was formed in 2005 as a joint activity of Health Level Seven and the Object Management Group, both established industry standards

---

3 Note that Volumes I and II of this document will be released separately. Volume II is expected in 3Q2009.

4 Adapted from OASIS SOA-RA 4.1.1.2.1
bodies, to create healthcare industry SOA standards. SOA is an architectural concept and can be realised in many technologies, approaches, and business strategies.

Recognising that SOA is an emerging part of that landscape, HSSP is defining health industry SOA standards that promote interoperability. Simply applying SOA isn’t enough, particularly if two “services” do similar things but do so differently.

Within the context of SOA, the mission of HSSP is to define and realise a key set of “SOA friendly” standard service interfaces for Healthcare. Many of the infrastructure services, and even some of the key business services are common to other industries, but many are Healthcare specific. So, there are two fairly orthogonal aspects to consider: healthcare uniqueness and standardisation.

Since the first HSSP work has been published, hundreds of inquiries have been brought forward from the healthcare community asking how that work relates to their business problem. Put another way, **SOA standards, absent architectural guidance, are difficult to consume** – especially because they require changes to the business and not just the technology.

This document is a product of the HSSP community, and attempts to bring together the health industry SOA needs in business terms, relating that work to HSSP-defined services where appropriate. That said, the document is intended to be standalone on its own, and is equally valid whether or not you elect to apply HSSP standards.  

---

5 This document is not intended explain or elaborate on HSSP, its methodology, or provide details about its standards. Those seeking more on these subjects are encouraged to visit [http://hssp.healthinterop.org](http://hssp.healthinterop.org).

### 2.5 Healthcare-specific SOA Challenges

Meaningful interchange of health information, and ultimately health system interoperability, has long been of strong interest to healthcare organisations. Given that many organisations the world over are making tremendous investments at present in healthcare IT, the risk is that each organisation “does its own thing” and that we might end up no better off as a community than we are now.

Let us first consider healthcare uniqueness. While the health industry has many unique characteristics, many of those stem from the peculiarities of health information itself, and not necessarily from the behaviors exhibited on that information. For example, maintaining databases, locating information, managing finances, and so on are all generalisable needs not exclusive to healthcare, though health-specific business rules, policies, and regulations may impose constraints. For these reasons, there is a reuse potential for business services that are not healthcare-specific to be leverageable into health care SOA activities.

In other cases, healthcare needs will require their own, unique services due to the specialised nature of the problem, such as is the case with Clinical Decision Support. In these instances, reuse potential from elsewhere is not viable.

One of the key tenets within a SOA lies in the ability to adapt the architecture over time, adding new services, replacing existing services, and reconfiguring infrastructure all with minimal impacts to service consumers. SOA is intended to leverage existing resources – it is
not a “rip and replace” approach. Service oriented architectures can reduce the amount of custom point to point interfaces needed in a given environment.

This is an important characteristic since interface design, development, deployment and maintenance can be expensive. For these reasons, standards based interfaces add value as they promote consistency across vendor offerings, provide a common footing for custom development, and allow for business partners to interact based on shared capabilities while minimising the need for custom point-to-point interface development.

Implementations of standards based interfaces can reuse existing applications and software assets. HL7 has been providing clinical and administrative data standards for interoperability for many years. HSSP’s work builds upon this knowledge foundation, allowing information assets such as HL7 CDA documents or existing HL7 content leveraged as interface parameters for SOA services. One does not need to choose exclusively between these paradigms as implementations may take advantage of both.

An example of this in practice today is the HL7 CDA (Clinical Document Architecture) standard, which is based on HL7 v3 RIM. Service interfaces being defined and endorsed by organisations such as IHE (Integrating the Healthcare Enterprise) and Continua Health Alliance. In these examples, CDA-based clinical documents define information format (payload) passed when a service is invoked. HL7 V3 has been taking steps to extend beyond purely defining information to include behavioral characteristics. Ongoing work such as the HL7 Services-aware Enterprise Architecture Framework (SAEAF) illustrate this shift.

2.6 Core Principles

SOA projects tend to have many intricacies and complexities, often exacerbated due to the large number of services, service instances, and interrelationship among service components. Many organisations find it useful to establish a set of core, guiding principles that underpin their SOA activities. These principles serve as guideposts – overarching themes that assist when making detailed design and implementation decisions – and help organisations maintain focus on their overall programmatic objectives.

The principles listed here have underpinned the standards work reflected throughout this document, and are those tenets shared from across the community contributing to this document. Consider adopting or adapting these tenets for your own SOA activities:

- **SOA as an initiative needs to be owned by the business and not IT, and have executive sponsorship.** Designing, supporting, realising and adapting business processes is key in moving to a SOA environment. This is why senior sponsorship of SOA activities is essential, with representatives from the business as key stakeholders.

- **Purchasing decisions must conform to your [SOA] architectural plan, or the architecture is rendered irrelevant.**

- **SOA is not a replacement for “messaging” or “documents”, and these concepts are complementary to SOA implementations.**

- **Custom-defining your own SOA services will not enable interoperability with other organisations.** “Web-services” alone is not enough to ensure interoperability, which is why healthcare SOA standards are needed.

---

Adapted from Understanding Version 3 by Andrew Hinchley. Used with permission.
➢ **SOA, absent shared information meaning [shared semantics], is insufficient to achieve interoperability.** This is why the use of a common information model (such as the HL7 RIM) is important.

➢ **Services should be loosely coupled, “coarse-grained”, and place strong separation between service-providers and service-consumers.** This approach enables “separation of concerns” promoting reusability, cohesive functionality, and ability to customise implementation to meet local needs.

➢ **Service functional specifications should provide for platform and implementation independence.** This fosters their viability as technologies change.

➢ “**Make it as simple as possible, but not simpler**”

<table>
<thead>
<tr>
<th>Desired Business Impact</th>
<th>SOA Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foster business agility</td>
<td>Provides a foundation for rapid reconfiguration, responsiveness to changing needs, flexibility to alter workflow.</td>
</tr>
<tr>
<td>Retained value from existing health technology investments</td>
<td>SOA provides mechanisms to leverage existing investments by “wrappering” legacy applications using SOA interfaces extending their usable life while reducing lock-in risks associated with given platforms and tools.</td>
</tr>
<tr>
<td>Ability to deploy best-of-breed solutions</td>
<td>SOA allows products to be selected based on fitness to support business functions and integrate architecturally with other products.</td>
</tr>
<tr>
<td>Achieve consistency in business-facing services</td>
<td>Bringing together disparate delivery mechanisms into a service-architecture harmonizes inconsistencies and unifies customer-facing business capabilities.</td>
</tr>
<tr>
<td>Improve alignment between IT and business need</td>
<td>SOA applications and interactions more closely mimic real business processes and interactions, reducing the “impedance mismatch” between business and systems.</td>
</tr>
<tr>
<td>Foster interoperability through adoption of standard interfaces</td>
<td>Integrating business capabilities using a services-backplane, applications integrate into the bus and not each other. This reduces custom interfaces, and allows for system replacement/substitutability with lessened impact.</td>
</tr>
<tr>
<td>Improved ability to adapt workflow</td>
<td>Since workflow in SOA systems is usually dynamic, using business process management (engines) to coordinate activities, rules and flow changes can be made quickly and done by business staff/analysts and not programmers.</td>
</tr>
<tr>
<td>Improved visibility into business process</td>
<td>Use of dashboards / drilldown capability offers management insight into execution of business process, improving visibility to identify and correct concerns; transaction information is available in real-time.</td>
</tr>
<tr>
<td>Improved IT organisational responsiveness</td>
<td>Better tools and minimised integration dependencies allow IT organisation to be more responsive to business needs, reducing maintenance burdens.</td>
</tr>
</tbody>
</table>

**Practical Guide to SOA in Healthcare**

©2008, Healthcare Services Specification Project, Health Level Seven, Object Management Group. All rights reserved.
3 Establishing a Healthcare SOA

While healthcare has some very unique challenges, not every challenge is unique. As organisations consider the use of service-oriented architecture as a foundation, it is essential that the rationale behind this direction is grounded in business-need, and that industry best-practices are considered and applied to the modernisation effort. This section will highlight some of these best-practices as pertaining to a new healthcare-affiliated SOA implementation.

3.1 Business case for SOA

Fundamentally, perhaps the most important consideration to be given to SOA-based modernisation activities is that SOA is a business initiative and not a technology one. Most SOA projects that are considered to be technology-based fail. Simply put, service-orienting an Enterprise means bringing together business functions under common responsibility, governance, and ownership. As most IT organisations lack the authority to make these changes, IT-owned SOA initiatives fail. (Governance is discussed in additional detail in Section 3.2.5 and 3.2.8 of this document.)

That said, a common, successful business practice is to use IT modernisation as the catalyst for change, supported by the business owners, and with accountability to senior leadership sponsors. This model fosters the ability for the organisation to deploy the changes it needs to, on the heels of the analysis and systems implementation driven by the IT. The above table highlights implications of SOA in a healthcare environment.

This is not to say that SOA efforts are independent of Enterprise Architecture (EA) programs. Quite the contrary, SOA makes for a good complement to an EA program, and is an entirely viable solution architecture for an EA.

Inter-Organisational SOA. There is nothing inherent in SOA that limits its use to within one

---

---

Practice Guide to SOA in Healthcare

©2008, Healthcare Services Specification Project, Health Level Seven, Object Management Group. All rights reserved.
organisation. In fact, there are a number of benefits that SOA brings when used as an integration mechanisms between organisations. For instance, as the services that a given business provides are better understood, those services can be exposed to “consumers” very naturally using SOA. This approach often results in more natural business integration. Though a single organisational example is used as the basis for this document, the principles explained here are equally applicable for B2B, regional, and national integration.

3.2 Introducing SampleHealth

To assist in the understanding of SOA deployment, a fictitious health care organisation is used to demonstrate the methods and supporting processes. SampleHealth—a fictitious healthcare organisation invented as an illustrative example—is representative of healthcare institutions the world over, and should share many commonalities with your own organisation.

The method for deployment will be presented as a eight step process. By definition, the descriptions are high level and must be detailed to the specific institution and its specific EA.

3.2.1 Step 1: Enterprise Architecture

An accepted industry practice is to leverage your organisational Enterprise Architecture when establishing a SOA. In fact some make the case that SOA is a part of or an extension to enterprise architecture. The Open Group’s7 Enterprise Architecture Framework ( TOGAF ), for example, contains four architectural domains:

- Business
- Application
- Data
- Technology

At its most elemental level, an Enterprise Architecture starts with a model of your business and its functions, and then evidences traceability throughout all of your systems and IT investments to ensure that your business is being supported. An EA provides the bridge between today and tomorrow, painting the vision for the future of the organisation, and instituting the governance to ensure that investments progress the organisation to that vision.

As specifically pertaining to SOA, EA provides a few benefits. It helps improve understanding of business functions, thus defining the “services” that ultimately make-up the SOA. This understanding also helps differentiate between “core functions” – the functions that comprise the organisation, from supportive/enabling functions, which help the organisation do what it needs to. SOA has a role in both.8 It can also provide the basis for the separation of dynamically changing processes and more stable service capabilities.

Deciding on and governing consistency and usage of things like naming standards and technology standards are important for scaling across a SOA ecosystem. Effective governance models and ownership of services become as critical success factor for SOA since SOA services are provided and owned by various parties. There is a section later in this document devoted to the subject of SOA governance.

---

7 See http://www.theopengroup.org for more information

8 While this paper discusses a “top-down” methodology for identifying services, many organisations find it viable to approach this “bottom-up”. A bottom-up approach identifies self-evident areas of shared need with tangible business impact as a basis from which to grow the SOA programme.
Understanding Service “Granularity”

Before significant progress can be made in enumerating and specifying the service capabilities you will be supporting, consideration must be given to the “granularity” of your services. Fundamentally, granularity is nothing more than the tradeoff between specificity and genericism of your service. There is no right or wrong approach to service granularity. The following table contrasts coarse- and fine-grained services.

<table>
<thead>
<tr>
<th>Fine</th>
<th>Coarse</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Thousands of services</td>
<td>• Dozens of services</td>
</tr>
<tr>
<td>• Each service specific to one function</td>
<td>• Functional ‘overloading’ provides reuse in multiple contexts</td>
</tr>
<tr>
<td>• Services tend to be content-oriented and domain-aware</td>
<td>• Services tend to be content-agnostic</td>
</tr>
</tbody>
</table>

Service granularity refers to the scope of functionality and purpose of a service. Operation granularity refers to the functional scope and corresponding message size for single transactions. Within HSSP, a decision was made to specify coarse-grained service interfaces. This decision was made for several reasons. First, having broadly-applicable service capabilities allowed the standards being developed within HSSP to be used in multiple contexts, increasing their viability and usefulness. The level of granularity chosen for a given service and its operations is an important consideration since it affects the coupling, the cohesiveness, and the responsiveness of the service and the system that uses it.

Within a SOA implementation, however, organisations must worry about the implementation behind an interface, and how that implementation is deployed. The HSSP community felt that coarse-grained interface specification could best support these needs, allowing the users to make implementation decisions about the granularity of the service instance. Additionally, the service interfaces were leverageable atop many different service implementations (e.g., data-specific services, domain-specific services, content-specific services, and the like), effectively allowing them to be useful in support of fine-grained service needs as well. The following should be considered when defining the granularity of a new service and its operations:

- **Coarse-grained services** are often used for inter-enterprise communication and for intra-enterprise communication between business applications due to their focus on activities defined at the business process level. They require less communication than fine-grained services, making them well-suited for network performance, especially in transfers of accountability such as occur in B2B transactions.

- **Fine-grained services** are best used when communicating between parts of a composite application inside a given organisation where the internal network is faster and more stable. High level of interaction between fine-grained components can result in unacceptable overhead when used in a distributed, cross-enterprise environment.

- **User facing services** provide end-user functionality, such as portals, context management services, WSRP portlets. These can provide fine grained operations as opposed to the more coarse grained operations for application to application interactions. Note that many service implementations may also include finer grained local operations that are “private”, i.e. used by the external facing “public” operations.
3.2.2 Step 2: Define and Analyse As-Is State

Before we can effectively and appropriately identify what services are needed to support the Enterprise, we first must have a good understanding of the “as-is” environment. An analysis of the “as-is” environment captures several dimensions:

- the business functions being supported by the organisation
- inventory of existing systems (hardware, software, outsourced functions)
- identification of key business needs not being met
- identification of business objectives / quality attributes we seek to impact

Many organisations jump into solutioning too soon, without giving appropriate attention to the current landscape and organisational challenges. The result is often the automation of existing, broken processes.

One of the key benefits of SOA is that it provides a means to extract common functions from across multiple business lines, establish authoritative business ownership and accountability, and align IT assets to address those extracted business needs. A sufficient understanding of the present environment is essential to identify unplanned redundancies, business inconsistencies, and these shared functional needs/commonalities.

The SampleHealth Business Functions Map (facing page) identifies the business functions performed by SampleHealth. These high-level business lines are likely consistent with many mid-sized healthcare provider organisations. Note that they are not necessarily representative of organisational structure (e.g., personnel and physical assets within SampleHealth are not necessarily grouped according to these business lines). Ultimately this collection of information becomes the baseline against which gap analyses can be performed and the candidate list of “to-be” services identified.

3.2.3 Step 3: Identify Candidate Services

Many of the industry journals would have you believe that SOA can be purchased in a box, placed onto a server, and run. This is far from reality. What is real, however, is that SOA is an architectural concept, and many of the foundational parts of that architecture are indeed commercially available.

There are a host of available frameworks that break-down SOA into different capabilities and layers. This document does not take an opinion on which of these are preferred or not, instead focusing on commonalities that appear in common across the frameworks commonly cited. As such, we have characterised SOA services into three simplistic categories that follow. Note that this categorisation is not absolute but reflects recurring themes among several available industry reference frameworks::

- Technical/Infrastructure Services. Generally low-level utility capabilities, and are available in commercial offerings,
usually as part of an “Enterprise Service Bus.” These services are not vertical-market specific, and involve capabilities like service instance location, protocol/message routing, etc.

- Business Services. Business-services describe those capabilities that support business capabilities or processes, but are not so nuanced or specific that they are unique to any one vertical market. Payroll, accounting, human resource management, and demographics are examples.

- Healthcare-Unique Services. This category calls-out service capabilities that are either unique to healthcare, or for which healthcare has unique requirements. For instance, both clinical decision support and order management appear here. While order management may have non-healthcare-specific examples, we believe enough nuances exist in the health domain to warrant its inclusion as healthcare-unique.

The healthcare functions performed by SampleHealth across their care delivery network are enumerated in Table 3, following. [Note that the table below is based upon the business lines identified in the “As-Is” analysis of SampleHealth. Some business lines have been omitted to keep the example clear and concise.]

The table introduces a number of candidate “services” which are among the first stages of the “To-Be” SOA architectural design. The objective is for each of these services to have clearly-defined responsibility, unity of function (cohesion), and composability (e.g., minimum direct dependence on other services). Guidance about how to effectively identify and scope services follows in Section 3.2.5.

<table>
<thead>
<tr>
<th>Business Line</th>
<th>Healthcare-Unique Services</th>
<th>Business Svs</th>
<th>SOA Infrastructure Svs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pharmacy</td>
<td>X X X · X X X X X X · X X X X · X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Laboratory</td>
<td>X X X X X · X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Patient Administration</td>
<td>· · · · · · · · X X · · · · X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Order Entry/Mgmt</td>
<td>X X · · · · · X X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Scheduling</td>
<td>· · · · · · · · X X · · · · X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Registration</td>
<td>· · · · · · · · X X · · · · X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Care Management</td>
<td>X X X X X X X X X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Referrals/Referral Mgmt</td>
<td>X X X X X X X X X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Nursing</td>
<td>X X X X X X X X X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Emergency Department</td>
<td>X X X X X X X X X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Patient Billing</td>
<td>· X · X X X X X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Imaging/Radiology</td>
<td>· X · · · · · X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Clinical Decision Support</td>
<td>· · X X X X X X X X X X · · · · X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Facilities Management</td>
<td>· · · · · · · · · · · · · · X · · · · X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
<tr>
<td>Nutrition Mgmt (Dietetics)</td>
<td>X X · X X X X X X X X X X X X X X X X</td>
<td>· X X X X</td>
<td>X X X X</td>
</tr>
</tbody>
</table>

Table 3. SampleHealth Business Function Map

©2008, Healthcare Services Specification Project, Health Level Seven, Object Management Group. All rights reserved.
More importantly, this table allows a cross-reference to be performed relating each business line to the SOA infrastructure. A mistake often made by new practitioners exploring SOA is to create SOA services in a one-to-one relationship to existing systems – making each “As-Is” system its own service. The key to effective SOA services is to bring together into one construct all of the related data, business rules, and ownership of a given responsibility that can then be leveraged across multiple systems and/or business-lines.

Note that service candidates can be identified “top down”, as elaborated below, or bottom-up. A bottom-up approach looks for areas of shared need in the form of existing common functions, subroutines, or needs. These also service as good service candidates.

As an example of this, consider the Master Person Index “Service” column, which touches almost every business line SampleHealth performs. Why is that? Simply put, when performing activities such as nursing, scheduling, or care management, we must authoritatively identify which person is the target subject. The Master Person Index has authoritative responsibility for these identities, so we need to leverage that function across multiple business lines.

Conversely, Care Management, touches multiple services. To manage patient care for a given patient, it is not sufficient to simply know their identity. We must also review their electronic health record (EHR), retrieve laboratory results, evaluate the patient’s status (including navigation across codesets and terminologies) and so on. Multiple services must all work together (e.g., a choreographed workflow) to achieve our intended objective. Examples of how services interact to support use cases (e.g., the dynamic model) will be included in Volume II (to be published soon) of this document (to be published soon).

The table also classifies our proposed services into our high-level categories: healthcare-unique services, business services, and infrastructure services. While these classifications are not absolute, they do indicate expectation in terms of commercial availability of off-the-shelf solutions to potentially fit our needs, and help us anticipate the degree of customisation we might expect to perform to attain and deploy those capabilities.

For example, the Infrastructure Services tend to be well supported by the marketplace, are not industry-unique, and we would expect commercial offerings such as Enterprise Service Bus (ESB) products to bring many or all of those capabilities. Further, as we place an ESB into production, we would expect limited need to customise or tailor those products to fit our environment (outside of performance/platform/scalability concerns). This would likely be a significant contrast from a healthcare-specific service, such as Patient Evaluation (a form of clinical decision support) which may require extensive customisations to support our preferred clinical protocols. Note that commercial marketplace offerings for all types of services are increasingly becoming available, including healthcare-specific SOA services.

3.2.4 Step 4: Define Future State

There are many outstanding resources available detailing methodologies that may be followed to identify good future-state service architectures. Defining a future-state architecture is establishing the target environment into which you will be transitioning your systems, and around which you will be re-aligning infrastructure to reduce unplanned redundancy, maximise reuse, and foster business-consistency.

While these objectives are common, ultimately the future-state architecture should be designed as a platform targeted to realise your intended business objectives. In defining this future state,
there are several steps that appear in common among multiple methodologies and approaches:

Assess your target business environment. A mistake often made is to develop a target architecture without consideration for changes in the business environment or the way the Enterprise would like to operate. This would include changes to the organisation’s operating model (such as a shift to wellness/preventative care or outpatient care), disruptive technologies (such as mobile devices and personal health records), legislative influences, and best-practices.

Assets such as the HL7 Electronic Health Record System Functional Model (HL7 EHR System Functional Model) may be useful to conduct gap-analyses between the current organisational practice and commercial best-practices. Regardless of the assets used, the core need to ascertain how the Enterprise will be affected by anticipated business needs and environment is of paramount importance in designing a suitable architecture to address those needs.

Identify areas of shared function, data, or need. A key element to remember when designing a SOA is that services must be shared for an organisation to realise a return-on-investment. When functions are needed by multiple parts of the business (and are often implemented in multiple places), this is often done with increased expense and inconsistency.

Bringing shared functions under one programme, governance, and implementation in the form of a “common service” within a SOA affords economy-of-scale benefits, fostering business consistency, minimising unplanned redundancy, and promoting reuse. Perhaps most importantly, it creates an authoritative source of information within the Enterprise – the single “source of truth” to which systems can turn for accurate information.

Identify business “actors.” An effective vehicle for determining how the future state architecture fits within the organisation is to conduct an analysis based upon the business entities that will interact with the architecture. We recommend distinguishing between business actors and system actors as the nature of their interactions with the architecture are commonly very different. In this analysis, placing focus on information access and exchange offers insight into the services that need to be provided and how they support business requirements.

Consider the organisation and its policies. One important objective of a future-state architecture is to design for and mitigate anticipated changes. One of the most significant challenges for any architecture is to provide significant flexibility to accommodate change without “breaking” the architecture, and to do so in a way that enables IT development and evolution without being overly onerous.

A review of organisational policy (both internal policy and external policies that affect data exchange) is useful to highlight requirements and to identify volatility. Policies can and will change. A careful consideration of policies and their architectural impact is useful in determining how and where to address those requirements architecturally.

---

9 Actor is a term use to represent any stakeholder (organisation, person, system, or other business entity) with which the architecture and its systems interact.
10 Simply put, business actors represent real-world business entities, such as users, user-communities, organisational entities (such as Department and Business Partners) and so on. System actors would be more IT-oriented, such as commercial software applications and software services.
A useful “rule-of-thumb” is to establish constructs that allow for the separation of policies from the infrastructure itself. This minimises the impact of changed policy on the architecture.

**Plan your Governance Programme.**

Governance is an absolutely critical dimension to SOA implementations, as it becomes the guidance, the oversight, the enforcement, and ultimately the compliance mechanism needed for a service-oriented architecture to be successful. Governance should be considered and planned for throughout the life cycle evolution of your SOA effort. In other words, it cannot be done as an afterthought after your architecture is built and is deployed.

While Governance is discussed in significantly more detail later in this paper, the issue is raised here so that factors such as organisational ownership, compliance reviews, service contracts, service-level agreements, and these other governance-related issues are planned for and considered throughout the entirety of the process.

**Identify your candidate services.** Several dimensions are useful when considering which candidate services are appropriate for a target-state architecture. A shared need across multiple parts of the organisation, especially when data and business-rules commonality exist, is often a good indicator of a candidate service. One approach commonly taken is to develop business process models, which give insight into the workings of the organisation and highlight candidate services from the “top-down.”

Similarly, areas of high volatility and/or functional cohesion (e.g., performing a closely-related set of operations) are good indicators as well.

Work is underway to elaborate healthcare service catalogues (vis-à-vis a formally defined ontology of services), in addition to richly-populated catalogues that are publicly available from large healthcare organisations, such as Canada Health Infoway and the [US] Veterans Health Administration. Additionally, work products such as the HSSP “Roadmap” may be useful when considering future-state candidate services.

**3.2.5 Step 5: Specify Architecture and Services**

As the business needs have been understood and the candidate services identified, operational and design considerations affecting the architecture come into play as the SOA is specified and designed. It is in this stage that the so-called “non-functional” requirements, such as the performance considerations, qualities (e.g., the “ilities” such as availability, reliability, scalability, etc.), and security/privacy come into...
play. The major steps involved in architectural specification follow:

Establish architectural framework. The “Service Bus” is perhaps the most common service-oriented architecture in use. What then needs to be specified? Selecting a bus-based architecture alone is not a technical specification. Our technical specification must address a number of open questions, including:

- Is a commercial “Enterprise Service Bus” to be leveraged? For which services?
- What is the preferred underlying messaging protocol? Web Services/SOAP? Others?
- What service-level agreements are to be supported? What is acceptable downtime? What is acceptable latency?
- What topology is to be supported? Centralised? Federated? Peer-to-Peer?
- To what granularity do we wish to manage and audit our security?
- How will existing/legacy applications integrate into the framework? Will they be side-by-side deployments? Will we implement “SOA-wrappers” and integrate them? Will the new applications delegate responsibilities to them outside of the core architecture?

Perform “Black Box” Specification. One of the primary benefits of a SOA lies in its ability to encapsulate (or hide) unnecessary detail. Simply put, the focus is on what the “black box” is and does, and how a customer uses it. That said, customers of the “black box” deliberately remain ignorant as to how that service is provided. This encapsulation provides flexibility to evolve and improve implementations without affecting service consumers.

Each black box “unit of composition” is based on clearly defined behavior and is described and used via a formal interface. This interface includes the precise definition of inputs and outputs, anticipated behavior, and exception handling. This stage is also where candidate industry standards (such as those produced by HSSP) are assessed for applicability.

Identify assets that can be leveraged to realise your services. Upon completion of the black-box specification process, you will have a detailed inventory of the services that you need to realise in your implementation. Armed with these specifications, the next stage is to conduct an assessment to determine the approach for “realisation” of each of these services.

Since SOA is an architectural construct and not itself a product, one of the challenges faced at this stage is in identifying which components are to be purchased and deployed as-is, which are to be purchased and customised, and what needs to be developed outright. There is no clear-cut answer, and the needs will vary from institution to institution.

11 “Realisation” is an industry term used to represent the purchase and/or development and the deployment of a service into an operational environment.
A key element often forgotten at this stage is the role of legacy applications. Many organisations overlook the value of their legacy apps, electing to replace aging systems with newer technologies. Legacy applications can play two roles in service identification. First, they can be analysed and parsed to identify existing candidate service functions. Second, in some cases existing legacy investment is best leveraged by developing a new SOA interface to sit atop them, providing access to their information content without necessarily using the remainder of the application logic. This approach – “wrappering” – extends the viability and lifespan of these historical applications without making architectural compromises.

As each specified “black-box” is considered, an assessment of commercially-available and open source products can be performed to determine their fitness to the solution and the architecture. Fitness can be assessed from functional, technical, performance, and integration perspectives. The “ideal” application that meets and exceeds all technical and performance considerations but does not support your domestic language (English, Finnish, etc.) is not ideal. Similarly, the “perfect” application that meets all your functional needs but cannot integrate with the remainder of your Enterprise is similarly unsuitable.

With regard to integrating these SOA components, vendor support for open standards is essential. Where your black-box specification has called-out for compliance to standards, ensure that your purchasing processes mandate that, either via out-of-the-box support from the vendor, or via supplier-commitment to provide support. Even when product vendors do not natively support industry standards, these are generally negotiable items in advance of your purchase. If you have not negotiated this as part of the purchasing agreement, you will likely absorb these integration costs directly and will have to custom integrate these components.

**Perform “White Box” Specification.** Once a marketplace assessment has been performed and it is determined that a custom-development or off-the-shelf customisation is required, the next stage is white-box specification. Where black-box specification is an externally-facing view of the service from which the internals have been hidden, a white-box specification exists precisely to address those hidden details.

The purpose of a white-box specification is to address the design considerations necessary when building or customising the service. In custom development, this would include dimensions such as topology support (e.g., will the implementation support federation or peer-to-peer deployment), scalability considerations, performance engineering, and so on. For off-the-shelf customisations, this

---

**Practical Guide to SOA in Healthcare**

©2008, Healthcare Services Specification Project, Health Level Seven, Object Management Group. All rights reserved.
addresses issues such as information/semantic mappings, protocol transformations, configuration settings, etc.

Irrespective of the approach being taken, the result of this step must be a fully-specified, elaborated, implementable design. The target audience of this artifact is a developer or implementer, and the specification must be sufficiently robust to ensure that when built the end-product will properly integrate into the architecture.

3.2.6 Step 6: Build A Transition Plan.

It can be argued that the most important step in realising any sort of technical or organisational transformation lies in the ability to successfully develop and deploy transition planning. For example, even if the perfect future-state architecture has been designed and is deployed, a transition plan is essential to plan for all of the contextual needs to operationalising the new architecture. This may involve migration of data from legacy to new applications, redirecting sites to the new infrastructure, any changes in end-user affected applications and the corresponding training, and so on.

Equally important, transition planning is not a one-time event, as every architecture is a living artifact which will evolve and adapt to changing business environments. Over time, new components and services will be added to the SOA, new foundational technologies (such as communication protocols and operating platforms) will change, and so-too will information content being carried across the infrastructure. Establishing discipline and effectively planning, managing, and executing these changes as they occur is a key element in achieving the target architectural state.

The following are key steps to consider when developing your transition plans:

**Establish increments/phases of progress based on priorities.** Transformations such as the establishment of a SOA are large, complex undertakings that have a significant number of dependencies. These complexities can be managed by establishing an architectural “critical path,” determining the sequencing of components that must be present, and then subdividing that critical path into manageable increments of progress.

Each increment should consider the following qualities:

- Demonstrated business value
- Achievable in a “reasonable” period of time (6 months, 12 months, etc.)
- Objective, measurable “exit criteria” (to demonstrate that the increment is complete)
- Rigorously manage dependencies

Collectively, the above (though an incomplete list) are based on established industry best-practices around iterative development. Incremental development is ideally suited to service architectures as SOA is component based with loose coupling among the pieces. Using incremental techniques, different components of the SOA may be developed to different levels of depth and maturity, all aligned with an Enterprise Sequencing Plan.

**Establish an Investment Approach.** Effective transition planning recognises ties to the business, and must ultimately add value back to the organisation. Many organisations lack the appetite, endurance, and resolve to see large-scale technology transformations through to their completion. This is often due to poor planning, especially with respect to demonstrated returns-on-investment and added incremental business value resulting from the investment.
As the phases are determined and ultimately executed upon, ensure that each major stage of the effort yields demonstrable business value. This provides many benefits. First, it addresses the “so what?” question, evidencing how monies are being appropriately spent. Secondly, it provides incremental progress and the corresponding confidence in the IT organisation’s ability to execute.

Plan Enterprise Sequencing based on critical path and dependencies. A particular challenge in managing a transition to a new architectural construct is to effectively identify dependencies and the critical path needed to deploy. Service-based architectures have additional complexities in this regard, as they are based on component parts and not applications themselves. For example, if a particular application is planned to “go live” on a certain date, it is essential that the SOA components upon which it relies are available and deployed before that date.

That said, similar to standing-up any new application, ancillary issues such as data conversion, availability, and system performance considerations must also be taken into account as they are factors affecting the use of the component SOA service in production.

In “Step 4” of this document (above), three broad categorisations of service types were identified: infrastructure services, business-services, and healthcare-unique services. As the early phases of the critical-path sequencing is determined, we recommend particular consideration be given to the infrastructure services. It is common that these provide core capabilities that are de-facto dependencies upon which most other services rely: message transport, service discovery, security/privacy services, and so on. When developing your sequencing plan, the following are activities to consider:

- Produce a visually-integrated view of the enterprise to analyse and assess dependencies and interrelationships between business initiatives, stakeholders, and systems
- Conduct a review of major business initiatives and the impacts they are anticipated to have on IT investments and the architecture
- Define the factors critical to the successful deployment of each SOA service (and the whole architecture)
- Develop mitigation plans to address concerns affecting the critical path
- Delineate clear phasing with articulated milestones and success criteria
- Validate that identified project phases align to incremental returned business value
- Establish a governance authority (and procedures) to manage the change process

3.2.7 Step 7: Realise the SOA
Realisation of SOA is, in a large part, executing against the Enterprise Sequencing plan that was identified in the step above. In this stage, the “services backplane” is created, based upon your determined collection of commercial packages, customised off-the-shelf packages, and custom development.

As an organisation’s SOA expertise and capability matures, additional functional and non-functional capabilities will likely be required. These considerations should be reflected in your sequencing plan, which implemented stage-by-stage will increasingly take on complexities that result from large-scale Enterprise deployments (such as achieving desired service performance with a potentially unknown number of service requestors and demand spikes) [Zapthink].

Practical Guide to SOA in Healthcare
©2008, Healthcare Services Specification Project, Health Level Seven, Object Management Group. All rights reserved.
Procurement of products and services that align with emerging standards and technologies can be a major challenge. One approach is to allow a staggered conformance, allowing a supplier to win the contract with a contingency that they come into compliance within a specified period of time.

This approach both can stimulate the number of candidate respondents you receive, as well as incenting vendors to provide marketplace support for emerging standards.

Speaking broadly, and harking back to our differentiation of service types, infrastructure services are foundational and allow the Services Bus itself to operate. Business- and healthcare-services are outwardly facing, interacting with stakeholder applications and other services in support of dynamic or choreographed interactions fulfilling business needs such as workflow and processes.

Procuring software.
Advances in available off-the-shelf commercial offerings, often in the form of an Enterprise Service Bus (ESB), provides a reasonable starting point for establishing much of the infrastructure needed to support your Enterprise. Enforcing architectural decisions and direction at the point-of-purchase is a step often overlooked or underspecified, a contributing factor to significant integration challenges and ultimately affecting compliance.

It is essential that the black-box and white-box specification work that has been done is affected in the purchasing. This may take the form of mandated requirements as part of your organisational acquisition strategy, or it may be part of a contractual obligation to either vendors directly or integration contractors (such as in the development of SOA-wrappers).

Using Commercial ESBs.
The Enterprise Service Bus (ESB) is probably the most commonly used term and concept when describing SOA infrastructure [Gartner, June 2006]. ESB’s may be characterised as a product, a group of products, or an architectural style, with the objective to act as a broker between service consumers and service providers. ESBs typically perform these functions, thus reducing the number of point-to-point interfaces while promoting scalability and reuse. Typical among the functions found within commercial ESBs are:

- **Routing.** By managing service endpoints (e.g., service locations), the ESB eliminates the need for service consumers to know where services exist and how to reach them. This promotes flexibility, especially over time, and allows the ESB to serve as a gateway/entry point between organisations.

- **Protocol Conversion.** There may be many reasons why systems may more easily use certain protocols over others – historical, different organisational standards, different contexts, and so on. An ESB can provide a variety of conversion capabilities to allow service consumers and service providers to transport information in their preferred protocols, removing these burdens from the endpoints. [data transformation, message format changes, localisation, versioning, etc.] This assists not only in integration of legacy assets, but also fosters flexibility to add new technologies as they emerge.

- **Event Handling.** Sophisticated ESBs provide programming models and ability to introduce logic and rules to receive and process events received from service consumers and providers. Resulting actions may include exception handling, routing, converting, or transforming messages, accommodating changing
dynamics that may arise during the process of work streams of activity.

- **Security and Privacy.** Some ESB offerings also provide security capabilities which can offload some of this set of non-functional requirements from the service provider.

Managing Expectations.  

Institutionalising a new architecture, whether based on SOA or anything else, is going to be fraught with expectations. As part of your deployment strategy, we recommend a parallel communications effort to help set organisational expectations, collect organisational experiences, and overall contribute to the enterprise changes that will be occurring.

Part and parcel with managing expectations is to **establish formal, objective metrics** against which progress can be assessed. Evidence-based results that illustrate the impacts of your SOA programme are often essential in securing and maintaining long-term programme commitments, and in being able to demonstrate impacts of the architecture on the business.

Metrics that you establish should be meaningful to the objectives you were trying to achieve. For instance, counting the number of services that have been deployed or the number of applications integrated into the SOA are of little value if the business objective is improved system performance or reduced redundancy.

**Step 8: Deploy and Sustain**

Developing and maintaining services requires a more comprehensive approach to systems lifecycle management than traditional development. For example, before deploying a new service, it is essential that it comply with all mandated SOA standards (Naming, Contracts, Versioning, and so on) so that it integrates appropriately into the production environment.

Additionally, each service should have an identified ownership model. The owner of the service can be, depending on the organisation, various entities. Some companies have even broken down the ownership model to include business and technical support. Regardless, the point to be made is that each service requires a means to ensure care and feeding of the services deployed.

**Configuration/Version management in SOA systems is complex.** One of the most difficult and problematic areas facing organisations involved in significant IT changes lies in configuration management (CM) not only throughout the transformation process, but following it. CM is not inherently a SOA-based issue, as it affects any deployed IT systems, but it is exacerbated in SOA solutions as SOA tends to involve more components, more collaboration, and ultimately more interdependencies than do traditional “stovepipe” systems.

Any changes to a service can have impacts on the consumers using that service. For example, you cannot replace an Enterprise Master Patient Index with a non-compatible version without considering how other applications and services interact with that capability. Changes in the service code (programmatic), operations (method calls), parameters (payload) each may have impacts on the service contract. Contract changes do not affect the functionality of the service, but rather the means in which it can be consumed.

A number of considerations will affect your ability to successfully address version management in a SOA environment:
Identify and manage metadata effectively. Each service has a purpose, a context, a version, and dependencies. Establish an organisational strategy to accurately and rigorously capture these and maintain this information about your deployed system components. This is essential not only for maintenance and management, but also for service discovery.

Configuration management is not limited to software. Many organisations perform due-diligence regarding software versions, but neglect data versioning. For example, codesets change over time (such as the switch from ICD-9 to ICD-10), and software may have specific dependencies-on or support-for designated terminologies. Similarly, data representations and industry standards (such as the Continuity of Care Document—CCD) are also versioned and evolve. These must be considered as you manage and track versions within your organisation.

Topology affects configuration management. Organisational decisions such as how services are deployed—nationally or regionally—have strong ties to CM. The bigger the organisation and the more complexities associated with it, the stronger the CM plan must be to accommodate. For example, is it realistic for a regionally-based deployment to be updated simultaneously across all regions, especially if data scrubbing must occur? Can we support side-by-side instances of different versions of a service? Do newer services support backward compatibility? Is this managed at the service-instance?

Enforce Service Naming Standards, Contracts. Naming standards will vary, but it is considered a best-practice to promote consistency in service names, methods, and parameters. Among other benefits, having standardised service names improves the ability to “discover” services, fostering reuse.

Contracts are a set of promises that a service provider makes to consumers. Contracts can vary in information and can include information regarding operations, pre & post conditions, and error handling. Each service should specify a default (or unilateral) contract, and each service consumer should also have a binding contract, with service contracts being tracked and managed in each environment in which they are deployed (e.g., test, production).

Service Directories play a key role in Deployment. Many organisations fail to realise the role that Service Directories can play in effective deployment. A service directory provides the ability to “look up” services that can satisfy a need. This step is essential in growing beyond a “hard-wired” deployment where services sit specifically on designated hardware boxes, allowing new services to come online and be removed dynamically, to support dynamic load-balancing and workload distribution, to enforce security/privacy policy, and to support transition management and product versioning.

Services directories in a SOA is essentially a catalogue of services, service instances, and their locations within the network of the organisation being served. During runtime, they are accessed to identify what type of service capability is being sought, which version of the service is needed (allowing side-by-side deployments of different versions) and where that service is located. With processes such as a workflow engine or service orchestrator then taking that information and routing processing as appropriate.

Since this step can be applied at runtime, governance responsibilities such as organisational policy, user permissions, and session

---

Practical Guide to SOA in Healthcare - 22 -

©2008, Healthcare Services Specification Project, Health Level Seven, Object Management Group. All rights reserved.
Testing SOA-based solutions differs greatly from traditional applications because services collaborate and interact dynamically based upon situational need and workflow. Whether deploying a new, updated, or corrected service, it is imperative to employ stringent testing and deployment processes.

Testing tools can be configured to mimic the known consumers of a service. Additionally, testing should also mimic true production scenarios and use cases. Maintenance of a distinct testing environment is a best-practice as it allows for improved understanding of the impacts of system stressing, side-by-side versioning, retirement, and other potential events that could create instabilities.

Understanding service-to-service dependencies is critical in an operational environment. Maintaining organisational rigor around major and minor releases of services, (and the use of Service Lifecycle Management tools) aids in the institutional control to introduce, version, manage, measure, assess, and retire services as part of a comprehensive SOA programme.

**Maintain Business-line Accountability.** As SOA initiatives are ultimately reflections of the business units they support, tracking metrics, measuring impact, and ultimately reporting to business stakeholders is essential in quantifying value of the organisational SOA investment.

**Establish and Sustain Strong Governance.** Governance and oversight form the solid foundation needed for SOA success. Without governance, projects are not held accountable for architectural decisions, promoting inconsistencies in implementation and potential duplication of effort. Governance assures that the enterprise is aligning with the future-state vision, and is a critical component of the business-accountability already described. A good governance program includes oversight of project initiation, ongoing reviews of activities, and monitoring of outcome to assess business impact.

### 3.3 Lessons-Learned

Organisations leveraging service-oriented architecture have an increasing number of places to look to share lessons-learned, benefit from the experiences of others, and to harvest best-practices. In tandem with this document, HSSP has established a forum to further this interaction ([http://hsspforum.org](http://hsspforum.org)).

A common challenge faced by organisations embarking down a SOA path is to determine how to get started. SOA can be daunting, and it is hard to determine where to start, especially in establishing the initial business case to proceed.

One way to start your SOA efforts is with limited scope pilots that provide definitive business value and become proof points upon which to build further. Over time the value of SOA realised as well as the experience gained can become...
foundational elements to build upon. Here are a few suggestions:

- **The “Focus Area” Approach.** One approach to getting started with SOA is to identify an area of the organisation on which to focus, perhaps because there are already plans forming for a new business strategy or initiatives that will drive changes. This can become the opportunity to identify, design, and realise services based on a “top-down” understanding of business needs and processes as well as a “bottom-up” IT systems view.

- **The “Integrate Legacy” Approach.** Many organisations have challenges in integrating legacy systems with newly purchased environments or applications. This approach entails providing a new, standards based interface and implementation for a legacy system. Consider the type of system that contains important function and/or data that is not easily accessible by other systems because of differences in platform, lack of knowledge about the functions and data. This encapsulation, or wrapping, of existing logic and data, enables reuse by existing and future systems.

- **The “Information Access/Quality” Approach.** Consider an existing database that contains important information but is not easily accessible by some systems (whether due to differences in platform, complexities in database structure, or lack of authoritative data ownership/consistency within the organisation). Concerns about data interpretation, consistency, and performance may have led to decisions in the past to restrict data access. Wrapping data access with a standards-based service provides controlled access to data, clear ownership (e.g., a single “answer” for the enterprise), and can lead to new value derived from existing data sources.

A SOA best practice is to approach SOA design from a business centric perspective. Services, and their behavioral and information models will reflect the business domain.

### 3.3.1 Design Principles

The value of a SOA is directly related to the usability, relevance, value, and flexibility of the services it provides. The “right” service interfaces, in concert with the associated behavioral and informational models, are critical for success. Best practices for service design are readily available and extend the architectural principles described earlier. A few such design principles follow:

- **The business perspective is critical.** Although IT plays a critical role, SOA efforts led by IT are rarely successful. SOA designs should reflect the business domain, with service names, operation names and behavior that are meaningful. Associated information models and messages [GetLabReport(…)] may be a meaningful service operation, where GetRecord(…) likely is not.

- **Granularity matters.** Perhaps the most oft discussed service design consideration, service granularity applies at several levels in a design. For instance, the number of services in your SOA, and the number of operations within a given service (and the scope of function that those operations provide) are both affected by granularity.

It is generally advisable to avoid extremes—a course-grained approach with too few services (or operations within services) can lead to bulky and overloaded services.
with more function than may be needed in many scenarios. Conversely, a fine-grained approach with too many operations and services means they are potentially harder to consume and less reusable, more prone to modifications and cause performance issues. For example, you may have a fine-grained service to specifically update exclusively the enrollment status of a beneficiary.

Avoid Stateful interfaces. The need for state management developed over time as web applications evolved and needs increased to support increasingly sophisticated scenarios. An example might be the need to maintain user information and choices / preferences in the context of a transaction evolving over several interactions between browser and server. These general principles apply equally well to SOA.

Cookies are a well known solution to maintaining state in web applications, where user or session-based information is maintained within a service and stored locally on the client. There is tremendous value in this approach (evidenced by its popularity), particularly in its ability to scale and avoid performance concerns.

The more contextual state information is needed at a SOA interface, the less generally-applicable and reusable a service is. Given that a core objective of SOA is to allow for highly-independent components to collaborate dynamically, this removes some of the inherent benefits you would realise with SOA.

Avoid indiscriminant use of CRUD interfaces. Related to the granularity and business perspective topics, CRUD (Create, read, update, delete) operations appearing pervasively across a SOA generally serves as a warning sign of a poorly designed and implemented SOA. While service implementations may indeed perform CRUD functions, these are not typically exposed to service consumers.

As described above, it is good SOA practice to keep interfaces stateless, and use of CRUD-type interfaces often require detailed state understanding of underlying data content, putting undue burden on the service consumer. While there are situations that warrant this, such as “Master” data sources, the approach is not readily extensible to other services and should be considered carefully before adopted.

Note that the service design principles here are based on general industry best-practices, and are not healthcare-specific. A number of good reference sources are available in the References section of this Volume in Section 6.

3.3.2 Enabling Legacy

Legacy applications play a vital role within most organisations, and architectural initiatives such as SOA must take that value into account to realise and incorporate their business benefit into the design. The term “legacy” has come to have an inherent stigma, often associated with older technology. The practicalities, however, often overshadow these issues.

Few organisations have the budget to replace all of these systems, and fewer still have the ability to extract and convert all of the data resident within them to new systems. Frequently, a hybrid approach is the viable one, where new applications become the system-of-record from the day they are deployed, with historical data continuing to reside in these legacy applications.

SOA provides an effective approach for providing access to these historical systems,
without compromising architectural integrity or the to-be systems design. The use of wrappering (encapsulating application-specific interface work from the remainder of the architecture, exposed via a SOA interface) allows these applications to continue to live within the new architecture without the need for brittle integration or significant one-off development for new commercial or custom applications.

Let’s explore how this approach affects several components of what is found in a typical Enterprise SOA environment:

For the Legacy Application, there are few or no changes. Legacy applications typically interact with external systems using point-to-point messaging protocols, such as HL7 2.x. Within a SOA, they continue to do so, with their outbound messaging traffic either redirected to the SOA wrapper layer, or that redirection being managed in an interface engine. Similarly, inbound message traffic would continue to arrive as messages, though the sources of those messages will change.

For commercial applications (COTS), a mixture of use of direct SOA service calls and legacy messaging is likely, depending on vendor product maturity and organisational implementation guidance. Packages that do not natively support SOA can be wrapped in the same manner as legacy applications, with a specific custom wrapper forming the bridge between application and the SOA.

For the SOA Services Bus, new wrappers will be registered as part of the bus with the capabilities they provide, hiding any data and messaging protocol transformation needed to realise those capabilities. Service consumers would make SOA service calls in the manner prescribed within the architecture, ignorant to the fact that those service providers happen to be legacy applications.

For the interface engine, transformations will have to be created to perform the protocol and data mappings needed to support the legacy and target environments. In some implementations, these transformations would be masked and applied “within” the SOA wrapper. In others, the interface engine itself becomes a “Transformation Service” with that dedicated charge. Depending upon the engine in use and its support for SOA, product enhancements or refresh is often likely to glean the Enterprise Service Bus capabilities of-of-the-box.

Version management allows for the side-by-side support of disparate versions of software within the architecture. For example, if a “new commercial package” contains all data from 2008 forward, and the legacy application contains data prior to that, the services directory could manage this, directing data queries to the right source depending upon the nature of the query. For example, a query for a full patient history may be forwarded to both applications, where a “recent medications history” would only go to the new application.

3.3.3 “Localising” to Your Organisation

When implementing within an organisation that collaborates with other business partners and institutions, there is a natural tension between designing to specifically suit your situation versus aligning with industry trends and standards. Both alternatives involve compromise and can be painful.

Organisations that “go it alone” are able to custom-fit their IT systems to maximally align with their internal needs, but incur significant pain when technologies age, new products are to be integrated, and when interacting with business partners.
Conversely, organisations that seek to align with industry standards may incur additional costs (time and money) to make the standards work for them, must justify these expenditures in financial and business terms, and often end up with designs that appear counterintuitive or outside of the organisational culture.

One approach to managing these tensions is through the use of profiling—where generally-applicable, broad standards are constrained for specific use within an implementation. By doing so, implementations are interoperable as they conform to industry specifications, while supporting local needs by providing customisation detail specific to the organisation. This approach to standards has precedent, particularly in Integrating the Healthcare Enterprise (IHE)\(^\text{13}\), and is the model being followed in HSSP.

To maximise their utility and flexibility, HSSP has approached SOA service standardisation by developing coarse-grained, generic services with mechanisms for profiling to address local needs and customisation concerns.

On first glance, a review of HSSP’s Entity Identification Service (EIS) or the Retrieve Locate Update Service (RLUS) appears very generic. For example, EIS is used to help manage identities, but without a “profile” it does not address whether it is identifying patients, persons, physical equipment, or anything else.

Why? The challenge is that there are any number of things that may need identifying, and the core behaviors needed of a service are the same irrespective of who (or what) is being identified.

By using profiling to address these differences, EIS is broadly applicable, and can be used to specify Patient Identity services, Healthcare Provider Identity Services, and almost anything that needs identifying.

But what about standardisation? These specifications include Conformance Profiles. Conformance profiles define a name, a version, and the data elements that must be supported in the implementations that assert conformance to that profile. For instance, the Patient Identification Profile specifies which identifying traits are mandatory (for example, Date of Birth) and which are optional. Further, it designates which operations of the service must be implemented, as these standards define many operations, not all of which apply in every situation.

Taken together, the information semantics (inputs and outputs, with corresponding clinical terminology where appropriate) and the behavioural semantics (mandatory operations) form the conformance profile.

Referring back to the Oval diagram, note that interface specifications such as EIS and RLUS appear atop multiple services within the diagram. This is possible because of this profiling mechanism, allowing one specification to be used for multiple purposes.

In fact, depending upon how the service has been implemented, it is possible to use the exact same implementation to be used for multiple purposes, with the distinctions being managed in your service registry.

\(^{13}\) For more on IHE, visit their site at [http://www.ihe.net](http://www.ihe.net)
4 Acronym List

B2B: Business-to-Business
BPMN: Business Process Management Notation
BPEL: Business Process Execution Language
CCD: Continuity of Care Document
CDA: Clinical Document Architecture
CCR: Continuity of Care Record
CM: Configuration Management
CRFQ: Clinical Research Filtered Query
CTS/CTS2: Common Terminology Service
DSS: Decision Support Service
EA: Enterprise Architecture
ED: Emergency Department
EHR: Electronic Health Record
EIS: Entity Identification Service
ESB: Enterprise Service Bus
HL7: Health Level Seven
HSD: Human Services Directory
HSSP: Healthcare Services Specification Project
OMG: Object Management Group
PASS: Privacy, Access, Security Services
RLUS: Retrieve, Locate, Update Service
SOA: Service-oriented Architecture

5 Glossary

Capability, Service
CRFQ: Clinical Research Filtered Query
CTS/CTS2: Common Terminology Service
DSS: Decision Support Service
EA: Enterprise Architecture
EIS: Entity Identification Service
ESB: Enterprise Service Bus
HL7: Health Level Seven
HSD: Human Services Directory
HSSP: Healthcare Services Specification Project
OMG: Object Management Group
PASS: Privacy, Access, Security Services
RLUS: Retrieve, Locate, Update Service
SOA: Service-oriented Architecture
Web Service(s)

6 Acknowledgements

This document reflects the collective input from across the HSSP community, and was developed in collaboration during a series of teleconference calls and offline work from a host of contributors. In particular, the following individuals (listed alphabetically) and organisations made significant contributions to this work:

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Org. Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alex DeJong</td>
<td>Siemens</td>
</tr>
<tr>
<td>Alan Honey</td>
<td>A.P. Honey Systems</td>
</tr>
<tr>
<td>Steve Hufnagel</td>
<td>Military Health System (TIAG)</td>
</tr>
<tr>
<td>Don Jorgenson</td>
<td>Inpriva</td>
</tr>
<tr>
<td>Alean Kirnak</td>
<td>Kaiser-Permanente</td>
</tr>
<tr>
<td>Bruce Laidlaw</td>
<td>Sun Microsystems</td>
</tr>
<tr>
<td>Thad Marcelli</td>
<td>MomentumSI</td>
</tr>
<tr>
<td>Galen Mulrooney</td>
<td>Veterans Health Administration</td>
</tr>
<tr>
<td>Marc Newman</td>
<td>EDS, an HP Company</td>
</tr>
<tr>
<td>Rich Rogers</td>
<td>IBM</td>
</tr>
<tr>
<td>Ken Rubin</td>
<td>EDS, an HP Company</td>
</tr>
<tr>
<td>Ann Wrightson</td>
<td>Informing Healthcare</td>
</tr>
<tr>
<td>Ginna Yost</td>
<td>VisumPoint</td>
</tr>
</tbody>
</table>

* Indicates a major contributor

Special thanks to all who contributed to the authoring and review of this document.


## References

The following are industry sources that may prove useful

<table>
<thead>
<tr>
<th>Web Resources</th>
<th>Citation</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.soaconsortium.org">http://www.soaconsortium.org</a></td>
<td>The SOA Consortium</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Print Resources</th>
<th>Citation</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Building the Agile Enterprise: With SOA, BPM and MBM (2008)</td>
<td>Fred Cummins</td>
</tr>
<tr>
<td></td>
<td>Service-Oriented Architecture, Concepts Technology, and Design (2005)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference Implementations</th>
<th>Citation</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.gov.bc.ca/health/">http://www.gov.bc.ca/health/</a></td>
<td>BC Ministry of Health</td>
<td></td>
</tr>
<tr>
<td><a href="http://www1.va.gov/VHA_OI/page.cfm?page=42">http://www1.va.gov/VHA_OI/page.cfm?page=42</a></td>
<td>US Department of Veterans Affairs</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.infoway-inforoute.ca/">http://www.infoway-inforoute.ca/</a></td>
<td>Canada Health Infoway</td>
<td></td>
</tr>
</tbody>
</table>

---

14 Although many good vendor-based resources are available, none are cited to maintain neutrality and avoid favoritism.