Hospital IT challenges: is SOA the cure?

John Kane
Architecture Management, University Hospital Zurich

Summary

Hospitals are challenged to reduce information technology (IT) costs, whilst supporting ever new business requirements and improving business effectiveness and competitiveness in a changing healthcare environment. Service-oriented architecture (SOA) offers an alternative to traditional IT solutions, and it is estimated that 77% of all companies with more than 1000 employees will adopt it in some form in 2012 [1]. This article investigates SOA, its applicability and the consequences of adopting it in hospitals.

Hospital IT challenges

Like most organisations, hospitals rely heavily on IT systems, infrastructure, and services to support them in their day-to-day business. Healthcare costs have risen steadily in Switzerland since 1960 [2], and the introduction of diagnosis related groups (DRG) adds further pressure on hospitals to be more cost-effective in an increasingly competitive and cost-driven market. IT is challenged to reduce costs by automating processes and providing timely and accurate information for doctors, carers, and management, to improve decision-making and provide better intelligence on the effectiveness of resources used. The introduction of e-Health also challenges hospitals to be more efficient in the patient care process, by improving integration with healthcare partners. Typical candidates for further process improvements include administrative and paper-based processes, as well as those that cross application system boundaries.

Hospital IT landscapes typically consist of numerous off-the-shelf (OTS) and tailored legacy systems, which compete for IT resources. Such systems often present the following challenges:

- redundancy of functionality and data;
- high infrastructure needs/costs;
- lack of support for some business functional and data needs (current and future);
- lack of agility and extendability;
- costly system changes;
- process logic is hidden inside the system;
- rigidity of the system’s own data model;
- old and heterogeneous technologies;
- limited technical means of integration;
- poor documentation or “black box” solutions.

OTS solutions are typically cheaper to buy than tailored systems are to build, but they pose the following challenges, in addition to the above:

- higher maintenance and operations costs over time;
- upgrades (content and date) are controlled by the supplier company;
- no guarantee of quality of solution;
- risk of vendor "lock-in"/dependency;
- insufficient level of security in the system.

Estimates of maintenance costs as a percentage of overall costs vary between approximately 50 and 90% [3]. The complexity of maintenance work is greatly influenced by the size, heterogeneity, transparency, and structure of the IT landscape and its components.

SOA is considered by some to be the answer to these challenges, and its adoption is the next logical step for many IT organisations that have evolved through stages of silo architectures.

Stages of enterprise architecture maturity

Enterprise architectures typically evolve through stages of maturity until the appropriate balance between efficiency and flexibility is achieved [4]. IT organisations often start with a silo approach ("business silos architecture") where the needs of individual business units (e.g. hospital clinics) are prioritised over those of the company. IT is typically decentralised here, allowing maximum flexibility for the clinics to focus on their local needs, but creating redundancy across the enterprise. The desire for increased cost efficiency results in the centralisation of IT and infrastructure and the definition of standard technologies, in order to reduce IT service and licence costs ("standard technology architecture"). This reduces local business unit flexibility in their IT choices, but encourages the basis for standard software solutions.
Further desire for cost efficiency drives the IT to reduce redundancies through centralised and consolidation of its application portfolio. In addition, processes and data are standardised for a more enterprise-oriented view ("optimised core architecture"). Here, companies typically implement large standard solutions, e.g. a data warehouse or an enterprise resource planning (ERP) system. Some companies strive for even greater levels of flexibility, reuse, and cost efficiency in their IT solutions. The final stage of architecture maturity ("business modularity architecture") ensures maximum efficiency for the organisation. The focus here is on building enterprise common components and not business unit-specific applications. The first three stages of architecture maturity are based on silo architectures of large, stand-alone applications: the “silo world”. The fourth stage is based on a different architecture model of reusable service components: the "SOA world". Enterprises tend to progress sequentially from one architecture maturity stage to the next. Such transitions involve changes in IT management, development processes, and existing systems.

**Enterprise operating models**

The appropriate stage of architecture maturity for a hospital depends upon the business strategy and operating model. Typically, four operating models are used to describe how the enterprise operates from a business integration and standardisation perspective [4].

A **diversification operating model** consists of isolated, autonomous business units, each focusing primarily on its local needs. This model applies to hospitals with decentralised clinics, which have unique processes and little or no need to share data. Such hospitals might progress to a standardised technology architecture, in order to reduce their overall operational costs.

A **coordination operating model** consists of autonomous business units, which have a need to cooperate with each other. This model applies to hospitals with decentralised clinics, which share patient and other data. Such integration creates the need for some common standards, e.g. in shared data structures. Hospitals with this operating model are typically suited to an optimised core architecture with (at least) some common IT structures and a standard integration technology.

A **replication operating model** consists of clone-like business units with the same or similar processes and products, but probably no common data (e.g. a chain of doctor’s practices). Such business units are unlikely to need to integrate with each other, but benefit from standard data structures and functionality, based on the principle of repeating locally the best innovations from the enterprise as a whole. Enterprises with this operating model are typically suited to an optimised core architecture.

A **unification operating model** consists of closely cooperating, similar business units, with high levels of (data and process) standardisation and integration. This model applies to hospitals with a commitment for centralisation and a focus on maximising time and cost efficiency by standardising across clinics, whilst still allowing for unit-specific requirements as necessary. Such hospitals are best suited to an optimised core or business modularity architecture (and SOA), depending on the degree of efficiency, agility, and business differentiation required.

**SOA overview**

Figure 3 shows a simplified view of a hospital with “silo world” and “SOA world” systems and components. The SOA world is layered and consists of different types of components, which may provide (service providers) and/or use (service consumers) functionality in the form of services. **Client components** are responsible for displaying and capturing data, and reacting to user actions and requests. They use services from layers below, e.g. to start a process or save/retrieve data. **Process components** are responsible for managing business processes and human task lists, and they orchestrate the use of services from underlying components. **Business entity components** provide lower-level activities or business transactions for a business area, typically data-centric CRUD (create, read, update, delete) services. Business entity components control access to data in the layer below, and can be used as an integrating wrapper to hide the internal systems. **Data components** typically include a shared operational database. **Integration components**, such as an enterprise service bus (ESB), mediate and map data between applica-
Internal systems consist of standard OTS and tailored legacy systems. These are stand-alone, typically non-modular applications, which might make some of their functionality available via services.

The SOA world distinguishes between its two main reusable elements: data and service logic. Data is passed in structures, as part of the service requests and responses, and is visible across the layers. Service logic (functionality, processes, rules etc.) is hidden inside a service component behind a façade (see: [5]), thus protecting the service consumer from changes. The objective of the SOA world is to build enterprise common components, services and data, which can be used by all service consumers and business units, following the principle: “create once, use many”. This requires standard service logic and a common business data model.

The promise of SOA
SOA promises improvements in efficiency and business agility and, ultimately, in costs and business competitiveness, as a result of the following benefits:

- improved reliability from reuse and simplified testing of independent services;
- reduced unwanted redundancy through the reuse of centralised common services, data structures and operational database;
- reduced complexity due to the structuring, less redundancy, and fewer applications;
- reduced integration due to a shared operational database and fewer applications;
- reduced operations costs of a consolidated and less complex environment;
- simple maintenance of a clear structure with less complexity;
- quicker time to market changes due to reuse and reduced complexity.

The challenge of SOA
SOA brings new IT architecture concepts and methodologies, which demand changes to the software development process and probably also to the IT organisation. Adapting to a SOA/business modularity architecture and a focus on reusable, common components may require more effort than in the previous silo architecture stages, due to the higher level of business abstraction, the change of focus, and the necessary consolidations and coordination between business units. Also, a conflict with the previous (optimised core architecture) stage could arise if large, inflexible, standard solutions were acquired.

The success of SOA depends upon skilled project members, clearly defined development and governance processes, appropriate SOA projects, a suitable culture, and top management support. The quality and reusability of an SOA environment depends upon the quality of the architecture, analysis, and design work, in particular the quality of the data model (e.g. how well does it cover current and future business concepts?), component structuring (i.e., the patterns and layers of responsibilities defined), component design (i.e., cohesion and loose-coupling), and service design (e.g. business meaningfulness and granularity). Hospitals should consider the support of experienced IT professionals, in order to reduce the risks of failing or creating poor quality SOA solutions.

The case for SOA
The introduction of SOA in a company is typically driven by either IT needs to reduce complexity and development and operations costs, or business needs for improvements in process quality and efficiency. Research shows that SOA has been successful here [6]. SOA demands changes to the IT landscape, but this could be achieved partially (i.e., a combination of SOA and silo worlds) and incrementally (even one service at a time). The implementation of new SOA components and services, however, is only justified if service consumers exist which can make use of them. The following are example scenarios for new SOA projects:

1. A new central client web portal is required as a single point of access for the business. This would require new client components, and underlying components to support them.
2. Business processes should be automated for improved efficiency and quality. This would require new client and process components and underlying components to support them.

3. A new application is to be built using SOA, as a basis for the future and possible reuse. This would require new client components and underlying supporting components.

4. Existing application system functionality, which can be isolated and removed, is to be implemented centrally, e.g. for reuse purposes or as part of an incremental replacement of the system. This could require components across all the SOA layers.

Discussion

The value of SOA for a hospital depends upon its business strategy and operating model. The shift to a business modularity architecture (and SOA) involves costs and risks, but promises a return on investment, as a result of improved efficiency, agility, reuse, and maintainability.

Each hospital is unique in its IT landscape, capabilities, and culture. For each, the case for SOA varies, but still requires some leap of faith. Savings in hospitals through IT are achievable, particularly for the more IT advanced hospitals [7]. Whether and how far a leap from the “Silo world” to SOA is applicable for hospitals depends on questions such as:

- How advanced is our enterprise architecture and IT organisation?
- How much new software development do we envisage in the future?
- Where do we want to use tailored (SOA) components, instead of standard OTS (Silo) systems (e.g. for competitiveness)?
- What existing systems do we want to replace?
- How much potential for reuse do we have?
- How much potential for savings do we have?
- In the light of the promises of SOA, if other hospitals adopt it, can we afford not to?

References