Balancing between Local Requirements, Interoperability Standards, and SOA principles - Case eBooking of Health Services

Juha Mykkänen, PhD, Research director, Mika Tuomainen, Project manager

University of Eastern Finland, School of Computing, HIS R&D

Juha Mykkänen, University of Eastern Finland, School of Computing, HIS R&D, P.O.B. 1627, FI-70211 Kuopio, FINLAND. Email: juha.mykkanen@uef.fi.

Abstract

Service-oriented architecture is an attractive development approach for flexible and reusable healthcare IT solutions. However, there are many practical architectural challenges in developing Service-Oriented Architectures (SOA) in organizations. In practice, not all basic SOA principles can be easily followed using vertical standards and local adaptation is typically needed. In this paper, we discuss balancing between vertical interoperability standards, local requirements and SOA principles. We classify different types of conflicts between these elements and analyze healthcare electronic booking solutions as a case example. The establishment of inter-organizational interoperability solutions requires agreements on many levels, and open vertical standards such as HL7 combined with horizontal industry standards provide solutions to many of these levels. SOA based interfaces using vertical industry standards and models are good starting points, but they must be further refined to guarantee interoperability and fit for local requirements.

Keywords: interoperability, standards, SOA, appointment booking, requirements management, HL7

Introduction

Service-oriented architecture (SOA) [1] has been used as a development approach which aims to enhance business and IT alignment in many domains, including healthcare. Achieving this alignment in practice is still not easy and there are many misconceptions that that lead to oversimplifying the effort required implementing SOA [2]. For accomplishing SOA goals of business agility and IT effectiveness, the use or standards is essential for reuse and interoperability. The standards have often been developed to support many different types of use cases as the standardization process mandates multilateral and wide utilization. However, this may be problematic especially for vertical standards which contain features which are close to organizational and user processes, where several different variations need to be supported. The case studies from Ross [3], illustrate that organizations learn in stages the ability to define and align IT and business strategy by accumulating architecture-related experiences. Ross lists four stages of (1) application silo, (2) standardized technology, (3) rationalized data and (4) modular architectures. Stages 2 to 4 can be supported by various generic and domain-specific standards. Stage 4 can be interpreted as a stage where SOA is widely adopted in the organization's IT. This causes the need for locally adapting the use of the standards and may lead to problems in reusability and interoperability. In this paper, we study this dilemma through the relationship between project-specific or local requirements, interoperability standard specifications, and SOA principles and technologies.

Material and methods

The aim of this study is to study challenges and support utilization of vertical standards with SOA principles to satisfy local requirements for health IT systems. We retrospectively analyze solutions and specifications related to health services eBooking in relation to SOA principles and identified interoperability conflicts. The results are based on conceptual analysis of a case example in relation to principles suggested in literature. The case material is based on documentation and authors' experience from projects dealing with open interface specifications, SOA and eBooking. The first project is the SerAPI project where open eBooking interfaces were specified between regional and citizen-oriented health IT services and core administrative systems of health service providers. The second project is the eKat project whose one task was to coordinate regional projects and provide guidelines for electronic booking of health services on national level. In addition to these projects, the SOA approaches from other projects (MyWellbeing, the National Project for Social Services IT, Healthcare Services Specification Project) as well as HL7 standardization projects for national ePrescription and eArchive and two projects related to IHE (Integrating the Healthcare Enterprise) integration profiles have been used as a conceptual background. The analysis was performed as part of the SOLEA project dealing with enterprise architectures and SOA in domains such as healthcare, finance and machine engineering, and including several different application domain cases for SOA. Similar analysis was performed in SOLEA project for RosettaNet supply chain integration standards, but was not completed and is outside the wellbeing scope of this paper.

The concrete result material of the scheduling projects is available in Finnish [4,5]. In addition, the analysis is based on SOA and integration methods [6,7] and material on standards and their evaluation [8-12].

SOA Principles

Services in SOA are accessible through well-defined, standardized interface in a platform-neutral way. SOA is often associated with Web Service technologies, but SOA can be implemented with other technologies as well. Main design principles and best practices associated with SOA are [6]:

- Service Contracts: service has a well-defined, standardized interface which provides a formal definition (contract) of the operation of the service and its inputs and outputs.
- Loose Coupling: Service is independent from the state and context of other services and consumers using it. It is easier to change service implementation without need to change the other services that use it as they are not coupled to implementation details.
- Service Abstraction: Services are abstracted from specific software resources, non-essential information is abstracted and service consumers do not know the inner workings of the service, focusing only on the interface.
- Service Reusability: Service logic can be reused by other processes and services, in different contexts and by consumers not known at the design time.
- Service Autonomy: Services exercise a high level of control over their underlying runtime environment and have minimal dependencies to other services.
- Service Statelessness: The amount and duration of state information managed by service is minimized and limited only to current service invocation after which it returns to passive state.
- Service discoverability: Services have metadata to facilitate effective discovery and interpretation.
- Service composability: services can be combined with other services to form composite services.

In addition to these basic principles, many SOA approaches emphasize the definition of business-level services which provide interfaces with operations grouping related messages between services and their consumers. Web service technologies including WSDL and SOAP (and their extensions) are often used for interfaces and messaging. There are also various infrastructure offerings available supporting SOA, most notably Enterprise Service Buses (ESBs). Separation of architectural concerns such as functionality and informational semantics, service provision and consumption, process definitions and service implementations, are also emphasized as best practices, respectively [12].

Horizontal and Vertical standards in SOA

There is a number of interoperability related open standards and specifications for software applications. Interoperability standards include official and industry standards and can also be classified into domain-neutral (horizontal) and domain-specific (vertical) standards [13]. In this paper, we concentrate on vertical interoperability standards in healthcare. To set up integration between software applications, agreements are needed on interface and transport technologies as well as specification mechanisms for data contents exchanged and for the processes in which the data is communicated [12]. Many of these aspects are supported by horizontal and generic XML and SOA technologies.

Vertical standards have been developed over a long period of time to include many messaging-related aspects for integration solutions, in addition to the business data and business interactions. SOA solutions, on the other hand, build upon generic technology and process mechanisms to support these business interactions. Vertical standards provide a valuable asset for SOA solutions, containing many readily-specified aspects. Thus, the mediation of local requirements, SOA principles and vertical standards is a central challenge for various SOA initiatives and projects.

Main approaches for interfacing SOA applications include functionally oriented web services and document-oriented interactions [1]. Functionally oriented specifications rely on the (vertical) definition of operations and their input and output messages using (horizontal) interface technologies. Data in document-oriented SOA and interoperability standards is typically specified as (vertical) business documents. Parameters of messages or specifications of business documents aim to standardize the structure of the exchanged data.

On technology level, horizontal messaging standards also specify the technical interfaces, enveloping, security mechanisms and other technical details in communicating the data. XML schema and related technologies are often used as a primary enabler for domain-specific solutions. The use of a common data format, however, does not resolve interoperability issues in inter-organizational integrations, since the document or message semantics need to be understood consistently. Therefore, standards are needed that guide how e.g. XML is used in conjunction with relevant code lists and identifiers in vertical domains and subdomains such as healthcare and its appointment booking. Horizontal SOA specifications for messaging, functional and data-oriented interfaces and choreography specifications are important enablers of vertical solutions and vertical standards.

HL7 version 3 Scheduling

HL7 version 3 (HL7v3) [8,14] is a set of specifications which aim to support semantic interoperability between health information systems. Originally based on object-oriented principles, HL7v3 utilizes a shared Reference Information Model (RIM), data type definitions, vocabularies and a model-driven method for producing integration specifications [7]. More than 30 different committees in HL7 have produced foundation and technology specifications, structure and semantic design specifications and sub-domain models for healthcare integration in HL7v3.

HL7v3 utilizes UML-based, harmonized RIM model which gives structure to more constrained domain and message information models [14]. Normative message models can be mapped and constrained to XML implementation technology (schemas) which are non-normative. Interaction modeling of HL7v3 includes storyboards, application roles and trigger events which are referenced in message types. In addition, vocabulary domains, state transitions and common message elements are specified. Each HL7v3 domain (e.g. scheduling, patient administration, structured documents or clinical genomics) includes a domain information model and refined message models. Majority of HL7v3 specifications include messages for information interchange, but structured documents domain has generic Clinical Document Architecture (CDA) specifications for exchanging healthcare documents.

HL7 v3 messaging infrastructure specifications include wrappers for message transmission (including delivery acknowledgements) and control acts for workflow and error handling. HL7 v3 transport specifications specify web services, ebXML and simple minimum level transports for messaging.

HL7v3 messaging specifications can be applied in various ways in each domain. More detailed implementation guidelines have been produced for local or national application, in addition to local definition of extensions or new domains using HL7 methodology and locally specified vocabularies and code sets.

CDA documents and HL7v3 messages and models have been applied in several SOA-oriented projects, including regional or national use. However, the mapping of various HL7v3 artifacts to SOA concepts and principles is not uniform and horizontal technologies supporting SOA development have differences and overlaps on modeling, technology and infrastructure level in comparison to HL7v3.

With HL7v3 standards, local implementation guidelines are produced based on generic standard on a national level, and further refined for regional integration of scheduling solutions, for example. Healthcare organizations also utilize ESB and platform products which support Web Service technologies and standards such as HL7. Their functionality includes support for process designs, mappings between schemas and adapters to popular packaged applications.

Interoperability conflict types

As mentioned in previous sections, main analysis elements of this study are 1) local or project-specific requirements, 2) central design principles, features and domain-neutral technologies used in SOA initiatives, and 3) healthcare-specific interoperability standard specifications. Local requirements as well as horizontal and vertical standards are important inputs for interoperability solutions [7] which are necessary in SOA initiatives. We can identify conflicts on several levels in SOA projects which apply horizontal and especially vertical standards to local requirements. The most notable conflict types between the main elements include:

- conflicts between horizontal standards such as SOA technologies and vertical standards (HV),
- conflicts between SOA principles and vertical standards (SV),
- conflicts between local requirements and vertical standards (RV).

In the following sections we discuss these different types of conflicts using a case example and the SOA evaluation framework outlined in this section.

Results

In this section, we present the results, including 1) approaches of the related projects, 2) identification of conflicts between the study elements and 3) results of analysis of SOA features in relation to the case standard. Concepts of the standards evaluation framework [12] and HL7 methodology [14] are used.

We have applied HL7v3 "Scheduling" domain to support regional and web-based healthcare scheduling services which communicate with health service provider systems (hospitals, health centres etc.). The back-end scheduling applications have wide control over the time slots, resources and reservation rules for health services which are opened for external booking through the shared service.

For the above purpose, a model-driven approach for specifying integration solutions [7] was first applied as part of a multilateral project (SerAPI) which included several workshops, requirements documentation and harmonization and use case and architecture specifications. For integration technology standard selection, various alternatives were evaluated and HL7v3 [8] was selected. The standard was studied and then extended with required features according to the HL7v3 methodology. The project produced both a generic HL7v3 specification for scheduling use in Finland in general, and more specific implementation guidelines for regional scheduling projects [4]. The specifications have been harmonized and refined nationally and implemented in several health information system products to date.

Requirements conflicts (RV) in our case were first encountered in phase between the requirements and their mapping to the models of HL7 Scheduling domain. It was soon found out that the standard model and interactions missed several required features for national use in Finland and the regional scheduling. These included query interactions, referral identification information and identification of care pathways required by emerging national interoperability requirements. However, the base HL7v3 standard provides mechanisms to extend or adapt the base models, and several of them (constrained models, vocabularies, localization using HL7v3 methodology to produce extensions and new interactions) were used. In addition, reverse requirements conflicts were evident: several classes of the HL7v3 models included options which were unnecessary or too loosely defined for our case. In HL7 models, there were various options to put a given information element in, but these were narrowed down using implementation guidelines. In addition to many different types of model-level conflicts, also conflicts between technology and control conventions of the participating applications and the suggested use of web service technologies in the standard were observed. These included root elements in SOAP messages for special operations, and indirectly specified asynchronous communication model in transport layer which was incompatible with the architectural and synchronous requirements of the participants. These conflicts lead to three national harmonization cycles before the technical specifications were finally accepted.

Horizontal / vertical conflicts (HV) were encountered especially with messaging and transport layer. The project had been previously working with many web service interfaces which suggested us to use the web services transport profile also for HL7v3 messaging. However, the convention using which "standard HL7 XML messages" are wrapped to WSDL interfaces and SOA messages had significant differences to our previous experience of using tool-supported web service models. In practice, instead of code generation, a more XML-oriented development model was introduced.

In further work, we used the defined HL7v3 and regional scheduling service specifications as one basis for national guidelines for scheduling solutions with various maturity levels [15,5]. SOA approach was extended to promote different implementation pathways and to support modular transition towards higher maturity levels, aiming to support different regional strategies.

The summary of the analysis of scheduling standards and solutions in relation to central features and principles of SOA is presented in Tables 1 and 2 and the following paragraphs. The following notation is used in tables:

(++) very good support for SOA principles and features

(+) matching concepts can be identified or defined

(-) it is difficult to identify matching feature or justify the fulfillment of the SOA principle

(--) conflicts with SOA principles or features

Table 1. SOA basic features observed in relation to HL7v3 Scheduling.

SOA basic features	HL7v3, Scheduling domain correspondence
Service (business level)	+: Application role
Interface (set of related functions)	-: Application role, receiver responsibilities
Operation (functional capability)	-: Sequence of related messages, application role, message type
Message (payload semantics)	+: Message definitions, R-MIM, textual descriptions
Interface technologies	++: XML Implementation Technology Specification (serialization of HL7 models into XML schema) --: HL7 WS transport profile mapping to WSDL
Messaging technologies	+: HL7 Web services transport profile SOAP, WS-I --: HL7v3 transmission wrapper
Required or implied supporting technical infrastructure / ESB	+: Interface engine, messaging platform (implied)
Semantic specifications	+: Shared Reference Information Model, constrained domain and message models
Separation of business level functionality from technology (including message ids, acknowledgements, etc.)	-: Composite HL7 message includes payload, control act (work-flow), transmission (messaging): transmission and control layers do not have clear separation of business level functionality from interaction patterns or message identification, for example
Separation of functional capabilities from informational payload semantics	--: Functional responsibilities often implicit or using switches in message instances
Separation of service provider and consumer	-: Application roles are not explicitly providers or consumers, and include both
Separation of business process definition from service interfaces and implementa-tions	--: Composite HL7 messages include control act wrapper intended also for workflow specifications, but intermingled with the messaging

Table 2. SOA principles observed in relation to HL7 v3 Scheduling.

SOA basic principles	HL7v3, Scheduling domain
Service contract (operation, inputs, out-puts)	-: Not on operation level but on model and message levels
Loose coupling (independence from other services, consumers)	++: Self-contained messages and loose coupling (message oriented model)
Service abstraction (from specific software resources, omission of non-essential information)	++: Well abstracted from specific software -: lots of detailed optional information included in messages
Service reusability (reuse of service logic , including new contexts)	-: Service logic often not explicit +: messages and information models can be reused
Service autonomy (minimal dependencies to other services)	+: Dependencies mainly on information model level, little assumptions about runtime environment
Service statelessness	++: Messages tend to contain all the necessary state, no need for session management or stateful services
Service discoverability (metadata for discovery and interpretation)	-: Not much meaningful service metadata but reliance on standard documentation and implementation guidelines
Service composability (use for composite services)	-: Not actually discussed in standard +: but supported by shared information models

Service contracts in this case contain interfaces and their supporting documentation including regional guidelines. Interaction-specific XML schemas and SOAP examples serve as a main model for implementers, but most notably WSDL interfaces are of little use, as they mostly repeat aspects defined in other places and are unusable for code generation in practice. Significant uniformity in the interface is introduced by the extended standard and the implementation guideline. However, there are still rather many optional elements in the messages.

HL7v3 message-based solution provides loose coupling between participating systems. Similar or even more loosely coupled model could be achieved using also more document-oriented HL7 CDA approach. Technology neutrality and hidden software implementation is evident on technology level, but on model level, there were various information elements first omitted as internal to the organizations which were eventually included in the interfaces, as most scenarios required this control information. The specifications as such do not have much influence on the architectural patterns of finding the service or the back-end systems, but these are left to the communication and messaging infrastructure or configuration.

The service is abstracted on technology level and implementations could be rather easily replaced. The reusability of messages is excellent in healthcare settings, but has many distinct features which would be unnecessary burden for domain-neutral booking services. As the functionality is mainly implied by interaction types, no real function reuse can be evaluated. The service model is stateless, but acknowledgements are also used, and there are two types of setups for communicating the available time slots to the scheduling service.

Service autonomy could be hindered in a mediated architecture model with queried available time slots (one possible implementation setup). Here the scheduling service could potentially need to make synchronous queries to many back-end systems to retrieve available time slots. This could lead to performance problems upon unavailability of the back-end systems.

Service composability was one of the goals of the design. Indeed, the same interface which is used by the scheduling service for communication to the back-end systems, can be provided to the users of the service, providing centralized access point to service calendars of multiple organizations. In addition, the scheduling service seems to provide a central building block for SOA-based citizen services architecture in future.

The HL7 integration example highlights several conflicts and design choices related to the utilization of vertical standards. In relation to local SOA application, it also raises questions about the accuracy vs. generalizability of the solutions: project-specific requirements need to be fulfilled even if the generalized service or standard does not address all the needed features.

Discussion and conclusions

As we can see from the results, not all SOA features and principles can be easily supported as vertical standards are applied. The vertical standards often predate the idea of service-oriented computing and thus do not always perfectly match SOA ideals. Especially concerning these issues, the local adjustment is needed in the implementations. There are many issues which require good planning while locally implementing the standards. There is also some specification overlap between vertical standards and later horizontal SOA technologies. Many similar observations were made when analyzing RosettaNet specifications. To fulfil local requirements and to complement them for any real-world execution environment, additional details are needed on top of vertical standards such as HL7. Generic standards offer useful restriction mechanisms for implementation guidelines.

Conflicts between horizontal and vertical standards often arise from domain-specific extensions of extensible SOA base technologies which are inconsistent with other domains or invalidate the tool automation based on "clean" use of web services, for example. This seems to be typical for various document-oriented integration and SOA approaches.

Conflicts between SOA principles and vertical standards (SV) are evident due to long history of many vertical interoperability standards: SOA mechanisms have not been available and corresponding messaging layers have been inevitable in vertical standards. The modular design of standards and separation of various aspects of interoperability, however, greatly ease the introduction of new or alternative technologies once they emerge. Vertical standards do not have as clear a separation of concerns as SOA. On the other hand, SOA may presume a server-oriented mindset which is not necessarily optimal for real-world workflows and interaction patterns. Vertical standards do often have influence on many technical aspects in addition to their main scope in vertical space. Architectural or technical consequences may also be implied or hidden.

Conflicts between local requirements and vertical standards (RV) are manifold: missing parts and elements, partially matching semantics, and legacy-oriented constraints are common. Local requirements may conflict with SOA or standards on various different levels from business processes down to technology details.

Despite these conflicts and challenges, the use of vertical standards provided our case projects with lots of readily thought-of requirements and solution features, as well as existing models and infrastructures to draw from. Many SOA elements could be mapped to the elements of the standard. Similar conflicts and proposals to alleviate them have been identified and discussed between IHE standard profiles and SOA [9], and in further development of HL7 standards development methodology [10]. Recent approaches such as HL7 Services-Aware Interoperability Framework especially focus on increased support of SOA principles and improved reuse of base models across different integration paradigms. There is always a balance, however, between very strict guidelines and schemas and reusability of the solutions.

Acknowledgments

The authors thank the participants of the SOLEA, SerAPI and eKat projects and Paavo Kotinurmi and Timo Itälä who co-authored original and unpublished paper including comparison with RosettaNet standards which has been used as a basis of this article.

References

[1] Papazoglou MP, van den Heuvel WJ. Service oriented architectures: approaches, technologies and research issues. The VLDB Journal 2007;16(3):389-415.

[2] Lewis GA, Morris E, Simanta S, Wrage L. Common Misconceptions about Service-Oriented Architecture. In: Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems; 2007 Feb 26 - Mar 2. IEEE Computer Society; 2007. p.123-130.

[3] Ross J. Creating a strategic IT architecture competency: learning in stages. MIS Quarterly Executive 2003;2(1):31-43.

[4] Tuomainen M, Mykkänen J, Sormunen M, Luostarinen H, Pöyhölä A, Porrasmaa J. Ajanvarausrajapinnat - Tekninen liittymämäärittely. Versio 1.4. HL7 Finland ry; 2007.

[5] Mykkänen J, Tuomainen M, Kortekangas P, Niska A. Terveyspalvelujen ajanvarauksen valtakunnallisen arkkitehtuurin suuntaviivat. Oulu: eKat-koordinaatiohanke; 2008.

[6] Erl T. SOA Principles of Service Design. 1 edition. Prentice Hall; 2007. 608 p.

[7] Mykkänen J, Porrasmaa J, Rannanheimo J, Korpela M. A process for specifying integration for multi-tier sapplications in healthcare. Int J Med Inform 2003:70(2-3):173-182.

[8] HL7 version 3 Ballot: Domain: Scheduling. Health Level Seven Inc, January 2006.

[9] Painter J, Kirnak A, Moehrke J. A Service-oriented Architecture (SOA) View of IHE Profiles. IHE IT Infrastructure White paper, Public Comment version, 2009.

[10] Honey A, Koisch J. Services and the HL7 V3 Messaging Dynamic Model. White paper, HL7 Service Oriented Architecture Special Interest Group, 2007.

[11] Mead CN. Data interchange standards in healthcare IT - computable semantic interoperability: Now possible but still difficult, do we really need a better mousetrap? JHIM 2006;1:71-78.

[12] Mykkänen JA, Tuomainen MP. An evaluation and selection framework for interoperability standards. Inform Software Tech 2008:50(3):176-197.

[13] Nelson ML, Shaw MJ. The adoption and diffusion of interorganizational system standards and process innovations. In: MISQ Special Issue Workshop: Standard Making: A Critical Research Frontier for Information Systems; Seattle, Washington, USA. 2003. p. 258-301.

[14] Beeler GW. HL7 Version 3 - An object-oriented methodology for collaborative standards development. Int J Med Inform 1998;48(1-3):151-161.

[15] Mykkänen J, Tuomainen M, Kortekangas P, Niska A. Task analysis and application services for cross-organizational scheduling and citizen eBooking. In: Adlassnig K-P, Blobel B, Mantas J, Masic I, editors. Medical Informatics in a United and Healthy Europe - Proceedings of MIE 2009. Amsterdam: IOS Press, 2009. Studies in Health Technology and Informatics 150. p. 332-336.