The New Navigators: from Professionals to Patients
R. Baud et al. (Eds.)
IOS Press, 2003

Specification Levels and Collaborative Definition for the Integration of Health Information Systems

Juha Mykkänen[a], Tomi Tikkanen[b], Juha Rannanheimo[c], Anne Eerola[b], Mikko Korpela[a]

[a]University of Kuopio, Computing Centre, HIS R & D Unit, Kuopio, Finland

[b]University of Kuopio, Dept. of Computer Science and Applied Mathematics, Kuopio, Finland

[c]Pohjois-Savo Polytechnic, School of Business and Administration, Kuopio, Finland

Abstract

In this paper, we extend our previous work for defining interoperability for health information systems into proposed documentation levels for specifying integration, and a model for defining various interoperability aspects in collaboration between integrators, health service providers and system providers. We are using these models in defining solutions for a set of different integration needs in PlugIT project in Finland. We propose collaborative definition and clear levels of integration specifications to promote adequacy, consistency and efficiency of integration solutions and software component specifications in health information systems.

Keywords:
health information systems, integration, interoperability, interfaces, collaborative design

1. Introduction and objectives

Integration has remained one of the major issues in health information systems development [1]. Message-based integration has been used successfully in several integration projects. Message-based integration, however, does not solve problems of data and functionality redundancy in health information systems and is not able to provide a shared record [2]. Repeated point-to-point connections and non-consistent interfaces have made integration efforts difficult and costly. Lately, interoperability based on component interfaces, web services and coordinated clinical context between applications have been emerging as integration solutions [3, 4, 5, 6]. These solutions provide a tighter functional integration and more interactive interoperability between systems and reduce redundancy of data and functionality. The methods for application integration have focused on functional integration and technology-based interface specification. Selecting appropriate approaches for integration needs is a complex task in integration projects, and systems integration in healthcare requires definition of more specific processes for the integration [7, 1].

In PlugIT project [8], the aim is to increase interoperability and decrease the introduction threshold of health information systems in Finland using open interfaces and component-based approach. We aim to provide open interfaces for identified interoperability needs and define interoperability aspects, including standards-based and implementation-specific features, on a sufficient level to improve the reuse of integration solutions and to enable benefits of component and interface-based approach. In this paper, we extend our previous work and provide an overview of a model we propose for specifying integration collaboratively in integration projects. We also identify several practical levels of documentation as outcomes of the integration process.

2. Methods: Collaborative integration definition

Several reference models have been introduced for various aspects of integration. ISO reference model for open distributed processing (RM-ODP) [9] identifies several viewpoints to the interoperability of systems. Seven-level interoperability model [4] defines several levels, which address technical, architectural, functional, semantic and system lifecycle aspects of interoperability. In our previous work [10] we have introduced a process for defining many of these aspects for health information systems, but have not specified the multi-professional collaboration, which is needed to ensure that all viewpoints and aspects are considered.

Integration can be specified using an approach, where functional collaboration models or transactions are identified separately from technical implementations or technology standards. Even several different specific integration implementation technologies can be selected for one functional integration need. This approach has been emerging in domain-independent Model Driven Architecture approach by the OMG [11], as well as in Integrating Healthcare Enterprise (IHE) initiative in the radiology domain [12]. We have identified the responsibilities and capabilities of different parties and specified the necessary steps for producing interoperability specifications in multi-professional collaboration (See Figure 1). The system providers and their clients (health service providers and system users) must create a common vision of the goals of the integration. To achieve a common agreement on interoperability among e.g. competing system vendors requires identification of win-win opportunities as well as client demand for compatible interfaces.

The objective of the integration should be initiated from the process improvement requirements of the health service providers. These needs create integration needs for the information systems in the health environment. For integration projects, health professionals offer expertise in relation to the systems and their context in health care, in addition to the identified integration needs. The IT departments of the health service providers offer technical expertise as well as knowledge of the existing systems and technologies in their organisation.

Health system providers and vendors offer readily-made applications, which need to interoperate in the health environment. There are often readily-made point-to-point integration solutions, which can be generalised to provide a more reusable interoperability model. The system products provide a pilot platform for the specifications that are produced during the integration process.

In PlugIT project, the project itself acts as a neutral "integrator" which connects the health professionals, IT experts of health service providers, and system providers (typically at least two for each set of requirements) in a given integration situation. The health service providers and system providers may have their own quality management systems. Our process aims to be flexible enough to adapt into different work methods and organisational cultures. The project has already identified a set of important integration needs.

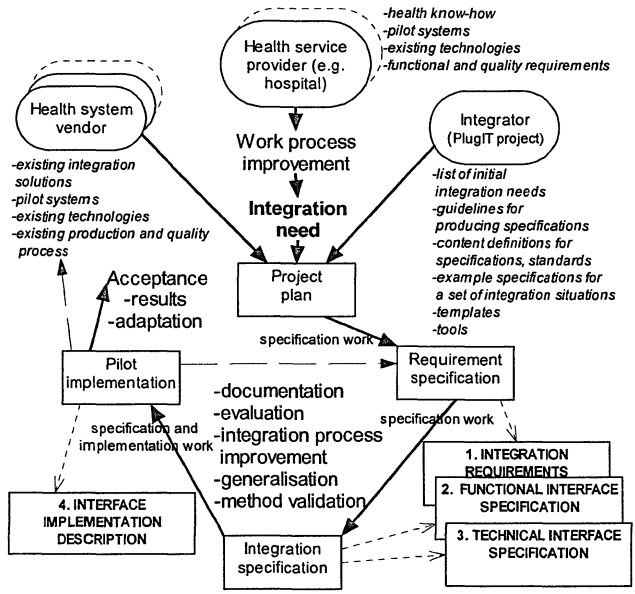

Figure 1 - Organisation of integration work in PlugIT project

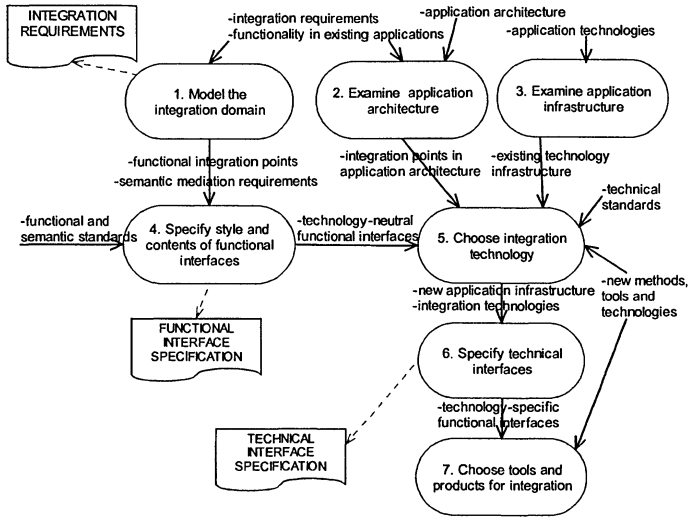

Figure 2 - Process for integration specification (Adapted from [10])

We aim to identify the interested parties in a given set of integration needs early in the integration process. The integrator contacts the participants and starts to outline and plan for the project. Project plan includes at least the purpose, functional outline, resources and time table, and relationships to other specifications and (e.g. system introduction) projects. In the specification work, draft specifications are produced, evaluated, revised if needed, and finally accepted as specifications collaboratively. (See Figure 1).

The project starts with the requirements specification phase, in which the functional requirements of the integration domain are elicited using use cases and process and data flow modelling etc. In the integration specification phase (See Figure 2), specifications for technology-neutral functional interfaces and collaboration models, and for technology-specific interfaces are produced. We have further developed the process specified in [10] to produce these specifications. Technical specifications are then implemented in pilot implementation and system introduction phases. Specifications can also become drafts for the new versions of specifications.

3. Results: Four levels of integration specifications

We are using a stepwise and incremental refinement approach in defining the integration. This reflects also in the documents produced. With this approach we are able to describe the sufficient level of detail in different levels, to provide easy start-up for integration projects, to offer clear and simple templates for specification documents, and to separate the interface implementation aspects from the interface specification aspects. The documents are shared between the participants.

We have defined some generic principles for all specification documents. The systems are identified as actors to provide a more generic approach, but system names are preferred in examples and analysis of current situation. The draft producers are encouraged to include the rationale for the design decisions in the documents. The documents may specify a minimum and additional conformance levels. The following integration specification documents are produced:

1. integration requirements,
2. functional interface design,
3. technical interface design, and
4. interface implementation description.

Integration requirements is a document, which sets the basis for the consecutive other documents and the implementation. The requirements document may cover a wide area of requirements, of which a smaller subset at a time is selected for further specification and implementation. This document is a shared description of the goals between the end users, the integration designers, and the implementers. This must be taken into account in writing the document, e.g. by avoiding

technical details and defining all central concepts. The most important part of the requirements is the outline of the functional needs for the integration; what is the part of the work and processes that needs to be developed. The functionality in the existing systems is an important input for the requirements document. It contains the identified actors or systems. UML use cases and activity diagrams can be used in requirements and process modelling. Also openness, transferability and maintainability requirements for the solution should be defined, as well as performance, security and other quality requirements, but they must be expressed realistically. The most important dependencies of different standards and the results from other projects should also be identified. In a given integration situation, the process can be stopped after the requirements phase, if there is no interest or resources in carrying the work further, or if other requirements have higher priorities.

Functional interface specification is based on the integration requirements. It addresses one clearly delimited functional integration need (scope), specifying a subset of integration requirements further. Thus there may be several functional interface specifications for one set of integration requirements. This document identifies the functionality (e.g. operations) and data contents (e.g. messages, parameters) for the integration situation. It also addresses what sort of integration is used (coordination between applications, data transfer or operation calls, invoking the user interface of another application etc.) It describes whether the integration is based on the network or installing the applications locally on the same network node or workstation, is there a user interface in addition to an API, and what the functional responsibilities between the participating systems are. The architecture and functionality in participating applications usually affect the functional interface specification. The aim is, however, to be able to support one functional interface specification using different technologies or application architectures. The functional interface specification is a common communication tool for the clients and the integration providers, and it should be clear and understandable for non-technical participants as well. Functionally the most important aspect of this specification is the description of the functionality and data for the integration. The specification may refer to functional and data standards (e.g. HL7 information models, EPR content definitions, CEN and OMG healthcare domain standards, IHE integration profiles etc.) Also the utilised code sets and their versions should be identified to provide the semantics for the data values (e.g. ICD classifications, national and provider-specific code sets). The existing specifications from the participants and feasible standards should be utilized.

Technical interface specification refers one functional interface specification document. There can be several technical interface specifications for one functional interface specification, if the same functional interfaces are implemented using different interface technologies. The technical interface specification is a design document and a technical contract between the implementers of the integration, including the developers who provide and use the interfaces. It contains the technical responsibilities of the systems, and has references to the selected integration technologies. The required technical infrastructure to support the *interface* technologies is described. The functional interfaces are specified using the selected technologies. The technology-specific documentation for APIs can be used. Semantics and format of each operation, data element, and return value must be specified. Strategies concerning empty data values, maximum data lengths, versioning of interfaces, specific order of invoking operations, and error handling are also included in the technical interface specification as needed. It is useful to utilize the same technical parts and standards in different integration situations. Selection of technologies depends on the selected integration model, and it is possible to select several technologies for one integration situation. In our project, we have identified some integration models for the integration situations. These integration models include:

workstation-level APIs (using e.g. Dynamic Link Libraries, COM components or simple web server interfaces), which may contain also user interfaces,

clinical context integration (coordinated user-oriented synchronisation of applications on the user workstation or web server), which can be based on CCOW [6] standard from HL7,

server-level APIs (using e.g. Web services, EJB, CORBA or COM+ components, or simple web server interfaces such as ASP, JSP, CGI and XML pages), without user interface,

use of common data storages or (e.g. CDA-encoded) repositories, and

message-based HL7 or XML interfaces with messaging platforms.

In PlugIT project, first three of these models are studied and solutions, templates and tools are developed for them, albeit we also make use of common markup languages (XML) and healthcare-specific content definitions and standards (HL7, OMG Healthcare specifications, CCOW).

Interface implementation description facilitates the evaluation, installation and maintenance of products implementing a given technical interface. There can be several product-specific implementation descriptions for one technical interface specification. The implementation description does not contain the specifics of the internal implementation aspects of the interface. It is more freely formatted than other specifications. It specifies the technical infrastructure required by the *implementation* of the interface (in addition to the interface infrastructure described in the technical interface specification). It also contains documentation of product- or component-specific configuration (e.g. service addresses or file names and parameterisation). Furthermore, it describes how the implementation responds to the requirements in higher-level specifications. The implementation description also documents additional restrictions brought by the implementation, and additions made to the higher-level specifications (e.g. additional encryption solution for network communication). It describes the portability and deployment of the implementation in different environments. It should be accompanied with examples on how to utilise the implementation, for example how to build client software for the provided service.

4. Discussion and Conclusions

The functional improvement needs in work practices of health care facilities are the starting point for the integration process. The commitment of both healthcare providers and information system providers is crucial for the process, and a neutral party (integrator, open project) is needed to facilitate and control the integration efforts. The process and the detailed documentation must be supported by guidelines, examples and templates because different parties of the process do not usually have all the expertise to consider every aspect of the interoperability. When our approach will be validated in practice, we also aim to provide example implementations and integration tools.

In April 2002 IMIA WG10 Working Conference, a need for common description for components in health information systems was identified. We propose the aspects in our functional interface specification, technical interface specification, and interface implementation description to be included in the generic description of healthcare software components.

The identified basic services can be expected to become part of the basic application infrastructure. This healthcare-specific middleware, or set of interfaces required in the application infrastructure [2,5], will be gradually extended to include more and more services for the health information systems. This enables vendors to focus in the core competence and advanced features in their products, and reduces overlapping clinical and administrative work as well as redundant data and functionality on intra-organizational (e.g. hospital) and inter-organizational (e.g. regional) levels.

In this paper, we neither presented the identified integration needs nor described the exact steps and decisions in the process, but provided an overview of the process and the documentation. We also did not describe the project-specific acceptance process, which is also needed to gain approval and credibility for different specifications and their versions. We are using the models presented above in defining solutions for health information system integration needs in PlugIT project [8]. These needs involve heterogeneous functional, architectural and technical needs and solutions, which tests the feasibility of our approach in different situations. Some specifications initiate from modelling

and re-engineering the work practices in health organizations (top-down), others from opening up and generalising the point-to-point integration solutions already found in the products (bottom-up). It is reasonable to reuse the existing integration know-how, infrastructure and technical parts in the systems and develop these models further based on the practical experiences.

5. Acknowledgements

This work is part of the PlugIT project, which is funded by the National Technology Agency of Finland TEKES together with a consortium of software companies and hospitals.

6. References

[1] Kuhn KA, Giuse DA. From Hospital Information Systems to Health Information Systems: Problems, Challenges, Perspectives. *Methods of Information in Medicine* 2001; 4:275-287.

[2] Katehakis DG, Kostomanolakis S, Tsiknakis M, Orphanoudakis C. An Open, Component-based Information Infrastructure to Support Integrated Regional Healthcare Networks. In: Patel V, Rogers R, Haux R, editors. MEDINFO 2001. Proceedings of the 10th World Congress on Medical Informatics; 2001 Sep 2-5; London, UK. Amsterdam: IOS Press; 2001. p. 18-22.

[3] Blobel B. Application of the component paradigm for analysis and design of advanced health system architectures. *International journal of Medical Informatics* 2000; 60:281-301.

[4] Herzum P, Sims O. Business Component Factory. New York: Wiley Computer Publishing; 2000.

[5] OMG Healthcare Domain Task Force. CORBAmed Roadmap, Version 2.0 (draft). OMG Document CORBAmed/2000-05-01; 2000.

[6] The Clinical Context Object Workgroup. The Clinical Context Object WorkGroup: It's Standards and Methods. Document CCOW98-02-16; 1998.

[7] Knaup P, Mludek V, Wiedemann T, Haux R, Bauer J, Kim L, Selle B, Schilling F. Integrating specialized application systems into hospital information systems - obstacles and factors for success. In: Hasman A, *et al.*, editors. Medical Infobahn for Europe. Proceedings of MIE2000 and GMDS2000. Amsterdam: IOS Press; 2000. p. 890-894.

[8] PlugIT project. PlugIT project home page. Available from: URL: http://www.uku.fi/atkk/plugit/

[9] ISO/IEC. Reference Model of Open Distributed Processing, Part 1. Overview. ODP Reference Model ITU-T X.901 | ISO/IEC 10746-1: 1995 DIS (E); 1995.

[10] Mykkänen J, Porrasmaa J, Korpela M. A process for specifying integration for multi-tier applications in healthcare. In: Surján G, Engelbrecht R, McNair P, editors. MIE 2002. Proceedings of the 17th International Congress of the European Federation of Medical Informatics; 2002 Aug 25-29; Budapest, Hungary. Amsterdam: IOS Press; 2002. p. 691-696.

[11] Object Management Group. Model Driven Architecture - A Technical Perspective. OMG Document ormsc/2001-07-01; 2001.

[12] HIMSS, RSNA. Integrating the Healthcare Enterprise - IHE Technical Framework Volume I - Integration Profiles, Revision 5.3. HIMSS/RSNA; 2002.

7. Address for correspondence

Juha Mykkänen, University of Kuopio, HIS R & D Unit, P.O.B. 1627, Fin-70211 Kuopio, Finland, Juha.Mykkanen@uku.fi