Semantic Integration in Healthcare Networks

Richard Lenz[a], Mario Beyer[a], Klaus A Kuhn[b]

[a]Philipps-Universität Marburg, Institut für Medizinische Informatik, Marburg, Germany
[b]Technische Universität München, Lehrstuhl für Medizinische Informatik, Munich, Germany

Abstract

A seamless support of information flow for increasingly distributed healthcare processes requires to integrate heterogeneous IT systems into a comprehensive distributed information system. Different standards contribute to ease this integration. In a research project focussing on the development of a reference architecture for inter-institutional health information systems, we identified and categorised concurring integration standards by distinguishing between technical and semantic integration on the one hand, and data and functional integration on the other hand. In addition, standards for semantic integration are roughly categorised according to their scope. By placing standards into a corresponding matrix a "semantic gap" is revealed, which cannot be covered by standards as it contains volatile medical concepts. As a conclusion, it is recommended to conceptually consider the necessity of system evolution in systems architectures and also in future integration standards.

Keywords:
Information Systems, Information Management, HIS

1. Introduction

Healthcare increasingly changes from isolated treatment episodes towards a continuous treatment process involving multiple healthcare professionals and various institutions. This change imposes new demanding requirements for IT. Thereby, IT applications should guide data acquisition in a way that data are ready for reuse in different contexts from the beginning, without the need to manually index or transform the data. To achieve this heterogeneous systems have to be integrated into a comprehensive distributed information system. Integrating autonomously developed applications, however, is a difficult task, as individual applications usually are not designed to cooperate and often based on differing conceptualisations. Powerful integration tools (e.g. application servers, object brokers, different kinds of message-oriented middleware, and workflow management systems [1]) are available to overcome technical and syntactical heterogeneity. Yet, *semantic heterogeneity* remains as a major barrier to seamless integration of autonomously developed software components (cf. [2]). Semantic heterogeneity occurs when there is disagreement about the meaning, interpretation or intended use of the same or related data [3]. It occurs in different contexts, like database schema integration, ontology mapping, or integration of different terminologies. The underlying problems are more or less the same, though they are often complex and still poorly understood. Stonebraker characterises disparate systems as "islands of information" and points out two major factors which aggravate systems integration [4]: (1) Each island (i.e. application) will have its own meaning of enterprise objects.

(2) Each island will have data that overlaps data in other islands. This partial redundancy generates a serious data integrity problem. Based on this statement, data integration can be led back to a mapping problem (semantic mapping of different conceptualisations) and a synchronisation problem (to ensure mutual consistency of redundant data in different databases under the control of autonomous applications). The mapping problem has been extensively discussed in the database literature under the term "database schema integration" (e.g. [5-8]). A major perception in this research has been that schema integration cannot be automated in general. Batini et al stated:*"The general problem of schema integration is undecidable."* [9]. Heiler states that *"understanding data and software can never be fully automated"* [10]. Consequently, the process of schema integration always needs a human integrator for semantic decisions. Colomb recognized that there are cases where no consistent interpretation of heterogeneous sources is possible ("*fundamental semantic heterogeneity*") [11]. In such cases one either has to modify software components or simply accept a low degree of data quality.

Standard ontologies are needed to reduce the effort for semantic integration. Moreover, as medicine is a rapidly evolving domain, concepts for system evolution are needed. Fortunately, far reaching standards for information interchange have already been developed in the medical domain. Yet, healthcare software is still far away from plug and play compatibility. In a research project focused on a reference architecture for comprehensive IS in healthcare networks [12], we have identified concurring and semantically overlapping standards. To get an overview of the standards' characteristics and interrelations, we have arranged them to a system of standards which we find to be helpful for architecture development.

2. Objectives

In this article we try to clarify how different standards contribute to systems integration by distinguishing different aspects and dimensions of integration. The objective of this approach is to identify and characterise the "semantic gap" not covered by standards. Our goal is to derive recommendations for future system architectures and standards development.

3. Methods

At a conceptual level, information systems are designed around three layers: presentation, application logic, and resource management [1]. According to this well known abstract m

odel of information systems, we distinguished different aspects of integration: data integration, functional integration and presentation integration:

- *Data integration:* When we characterized semantic heterogeneity as the main cause for high integration efforts, we focused on data integration, because it is the backbone and starting point of each successful integration project. Any process control always requires a meaningful exchange of data, too [13]. The goal of data integration is to create a unique semantic reference for commonly used data and to ensure data consistency. As a basic categorization for such a semantic reference we roughly distinguish three different facets: (1) The *instance level*, referring to the semantics of individual data objects, which corresponds to the meaning of entries in a database. (2) The *type level*, designating the semantic classification of data objects, which roughly corresponds to the database schema. (3) The *context*, which refers to the semantic relationships that associate an object with other objects.

  To illustrate the difference we may consider a concept "diagnosis" on the type level, and a particular instance, say "Encephalitis", and the context of this instance which is determined by the patient, the physician who made the diagnosis, and other objects that contribute to a particular statement (information).

- *Functional integration* refers to the meaningful cooperation of functions. Uncontrolled data redundancy is often the result of an insufficient functional integration. Autonomously developed systems often provide slightly differing but still overlapping functionality, which aggravates integration even if common ontologies are already used. Data integration is concerned with the consolidation of declarative knowledge, while functional integration is concerned with the consolidation of procedural knowledge on which applications are based. Both aspects have to be considered for *application integration*.
- *Desktop integration or presentation integration* refers to the user interface of a distributed system. Desktop integration is aimed at user transparency, meaning that the user would not know what application was being used or what database was being queried [14]. This requires more than a unified layout and uniform interaction mechanisms. Examples are "single sign-on" and "desktop synchronisation". Desktop synchronisation is needed when a user has multiple windows to different applications on her desktop that share a common context. Synchronisation is required when the context is changed in one of the interlinked applications.

Another orthogonal aspect of integration standards is their *scope*. We can distinguish between technical and semantic integration. By "technical integration" we refer to the technical infrastructure which supports application integration. "Semantic integration", in contrast, refers to the meaning of data and functions. By contrasting the scope with data and functional integration we receive a rough matrix that helps to characterise different integration standards. Table 1 shows how different standards can be positioned into this matrix.

Table 1 – A classification of integration standards

	Technical integration	*Semantic integration*
Data integration	Syntactic frameworks	Ontology and vocabulary
Functional integration	Middleware	Application frameworks

4. Results

XML and RDF are examples for *syntactic frameworks* supporting data integration. Standards for semantic integration in healthcare are increasingly *based on* XML in order to improve syntactical compatibility with commonly accepted data processing formats.

Middleware standards typically provide a common infrastructure for interconnecting distributed software components. Such standards are primarily intended to provide programming abstractions, which help a programmer to easily bridge different hardware, operating systems, and programming languages. Examples for standardisation efforts in this area are CORBA, .net, EJB, or Web Services.

Ontologies and vocabulary standards support semantic data integration, as they serve as a semantic reference for system programmers and users (cf. [15]). Considering the different facets of data integration we find that well accepted standards like HL7 V2 and DICOM are primarily concerned with organisational issues on a type level. Terminological control is only supported to a limited degree. Yet, numerous standards support terminological control for medical issues at an instance level. Upcoming standards like CDA [16] and DICOM SR cover the interchange of medical contents also on the type and context levels.

Despite well accepted standards for data integration like HL7 V2 and DICOM, healthcare applications are still far from plug and play compatibility. One reason for this is that the existing standards do not address functional integration issues sufficiently. In order to avoid these difficulties common *application frameworks* are required which serve as a reference for programmers to create functionally compatible software components. Requirements for an application framework directed towards open systems in the healthcare domain are described in [17]. In general such a framework must provide clear specifications of interfaces and interaction protocols which are needed for embedding a software component into a system of cooperating components. The best example for such a standard in the healthcare domain is the IHE initiative ("Integrating the Healthcare Enterprise") [18]. IHE does not develop new standards for data interchange but specifies integration profiles on the basis of HL7 and DICOM. Thereby actors and transactions are defined independently from any specific software product. An integration profile specifies how different actors interact via IHE transactions in order to perform a special task. These integration profiles serve as a semantic reference for application programmers, so that they can build software products that can be functionally integrated into an IHE conformant application framework. HL7 V3 will also take a step into this direction, as conformance to HL7 V3 is specified in terms of "application roles" [19]. Like IHE actors, an application role is associated with some dedicated functionality (e.g. "lab order sender") – it comprises a set of trigger events, messages and data elements which are needed to integrate an IT component with this functionality. An IT component will typically fill many such application roles.

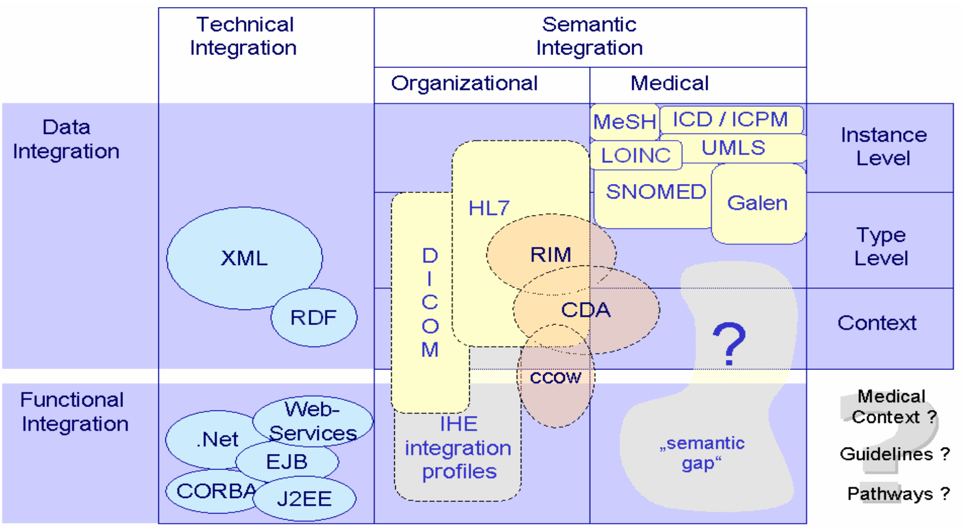

Figure 1 – Contribution of different standards to application integration

Figure 1 shows a rough characterisation of standards according to our classification matrix. The position of HL7 in this diagram refers to HL7 V2. Some improvements that come with HL7 V3 (e.g. RIM, CDA and CCOW [20]) are roughly indicated. The intention of the diagram is not to precisely and comprehensively classify the different standards but to get an idea which aspects of semantic integration are typically covered by such standards. It turns out that there is a gap in the lower right corner where standardised medical processes could have been expected. Medical pathways and guidelines fall into this category. This is essentially medical knowledge which has to be consented by medical experts and which evolves over time. Consented medical knowledge is necessary for cooperative patient treatment, but it is probably unsuitable as a subject of standardisation, as it rapidly evolves.

Despite of many standards for medical terminologies are in place. Yet, a unique and comprehensive ontology of the medical domain is not within sight, and, even worse, all given examples continuously evolve over time – necessarily. Thus, semantic integration of heterogeneous systems in healthcare will have to deal with volatile medical concepts.

5. Discussion and conclusions

Both reference ontologies and application frameworks are needed to support semantic integration. Yet, IT standards should not try to comprehensively map an application domain into a single model, as the domain continuously evolves. Instead, IT systems should be based on generic models and be capable of incorporating the results of ongoing consensus processes among healthcare professionals. The evolution of information systems should be a demand-driven process under the control of healthcare professionals. *Process integration* is concerned with the alignment of IT systems to actual business processes in a concrete setting. This is not addressed by standards, but by appropriate models for demand-driven software development (e.g. [21]). Desiderata for such a demand-driven process are:

- Tools and techniques for *rapid application development* (RAD). These tools should allow reuse of existing data and IT services.
- An IT infrastructure for a healthcare network should provide a *robust and stable basis* for application development. Thus, the framework should be based on generic but stable domain models instead of comprehensive but volatile domain models.
- Modeling of *domain concepts should be separated from IT system implementation*. IT systems should be implemented by IT experts and medical knowledge should be modeled and maintained by domain experts. Yet, this separation is not easy, because algorithms (e.g. reminder systems) typically refer to medical knowledge to fulfill their task. One attempt to support such a separation of concerns, is the "archetype" approach developed in the context of the GEHR project [22].
- To bring application development as close to the end user as possible, a *multi-layered software engineering approach* is proposed. An idealised abstract model for such a multi-level approach for software engineering is shown in Figure 2.
- *Layered ontologies* may serve within this layered software engineering process as semantic references on different levels of software development. The layered approach of the CDA, and the generic HL7 V3 Reference Information Model (RIM) are emerging standards which are already built on this fundamental principle.

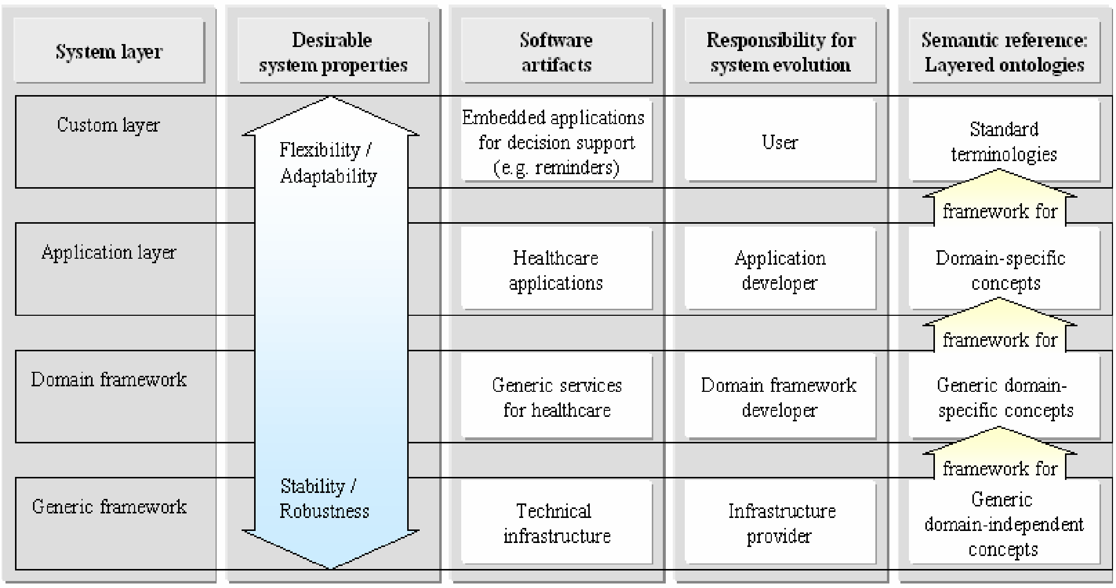

Figure 2 – A layered approach for system evolution

Layered approaches have proven to be a successful technique for separating concerns and reducing system complexity. Transferring this principle to the development and continuous improvement of information systems in complex application domains is aimed at allowing application developers and end users to build well integrated healthcare applications without the need to do low level coding and debugging. Appropriate tool support is needed at each level of abstraction in order to effectively make use of the lower system layers.

A layered approach, as sketched above, fosters a system evolution process that follows the principle of "deferred systems design" [23]: Volatile concepts are not pre-modelled and hard-coded in software, instead knowledge can be added or modified on demand and at runtime, as the domain evolves.

Our layered model can be used as an abstract reference model for evolutionary information systems. An example for an adaptation of this model to a real world hospital information system on the basis of commercially available system components is given in [21].

References

[1] Alonso G, Casati F, Kuno H, Machiraju V. Web Services - Concepts, Architectures and Applications. Berlin: Springer, 2003.

[2] Pollock JT. The Web Services Scandal - How Data Semantics Have Been Overlooked in Integration Solutions. EAI Journal 2002; 2002(August): 20-23.

[3] Sheth A, Larsen J. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys 1990; 22(3): 183-235.

[4] Stonebraker M. Integrating islands of information. EAI Journal 1999;(September/October): 1-5.

[5] Elmagarmid A, Rusinkiewicz M, Sheth A (Hrsg.). Management of heterogeneous and autonomous database systems. San Francisco, California: Morgan Kaufmann Publishers, 1999.

[6] Conrad S. Schemaintegration - Integrationskonflikte, Lösungsansätze, aktuelle Herausforderungen. Informatik Forschung und Entwicklung 2002; 2002(17): 101-111.

[7] Bouguettaya A, Benatallah B, Elmagarmid A. Interconnecting Heterogeneous Information Systems. Boston: Kluwer Academic Publishers, 1998.

[8] Rahm E, Bernstein PA. A survey of approaches to automatic schema matching. The VLDB Journal 2001; 2001(10): 334-350.

[9] Batini C, Lenzerini M, Navathe SB. A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 1986; 18(4): 323-364.

[10] Heiler S. Semantic Interoperability. ACM Computing Surveys 1995; 27(2): 271-273.

[11] Colomb RM. Impact of Semantic Heterogeneity on Federating Databases. The Computer Journal 1997; 40(5): 235-244.

[12] Lenz R, Beyer M, Meiler C, Jablonski S, Kuhn KA. Informationsintegration in Gesundheitsversorgungsnetzen - Herausforderungen an die Informatik. Informatik Spektrum 2005; 28(2): 105-119.

[13] Bange C. Von ETL zur Datenintegration. IT Fokus 2004; 3(4): 12-16.

[14] Pille BT, Antczak RK. Application Integration. In: Ball MJ, Douglas JV (Hrsg.). Performance Improvement Through Information Management. New York: Springer, 1999: 144-152.

[15] Lenz R, Kuhn KA. Intranet meets hospital information systems: the solution to the integration problem? Methods Inf Med 2001; 40(2): 99-105.

[16] Dolin RH, Alschuler L, Beebe C, Biron PV, Boyer SL, Essin D et al. The HL7 clinical document architecture. J Am Med Inform Assoc 2001; 8(6): 552-569.

[17] Lenz R, Huff S, Geissbühler A. Report of conference track 2: pathways to open architectures. Int J Med Inf 2003; 69(2-3): 297-299.

[18] Vegoda P. Introducing the IHE (Integrating the Healthcare Enterprise) concept. J Healthc Inf Manag 2002; 16(1): 22-24.

[19] Health Level Seven I. HL7 Version 3 Statement of Principles. Health Level Seven, Inc. URL: http://www.hl7.org/Library/data-model/SOP_980123_final.zip

[20] Seliger R. Overview of HL7's CCOW Standard. Health Level Seven, Inc. URL: http://www.hl7.org/library/committees/sigvi/ccow_overview_2001.doc

[21] Lenz R, Kuhn KA. Towards a continuous evolution and adaptation of information systems in healthcare. Int J Med Inf 2004; 73(1): 75-89.

[22] Beale T. Archetypes: Constraint-based Domain Models for Future-proof Information Systems. In: OOPSLA 2002 workshop on behavioural semantics; 2002

[23] Patel N. Adaptive Evolutionary Information Systems. London: Idea Group Publishing, 2003.

Address for correspondence

Dr. Richard Lenz, Institut für Medizinische Informatik,
Klinikum der Philipps-Universität Marburg
35033 Marburg/Lahn, GERMANY
Tel. +49–6421–28–66298, Fax. +49–6421–28–63599, Email: lenzr@staff.uni-marburg.de