electronic Journal of Health Informatics
http://www.ejhi.net
2010; Vol 5(1): e4

An Approach towards Semantic Interoperability using Domain Models

Georg Heidenreich[1], *Pantelis Angelidis*[2]
[1]*Siemens AG Healthcare, Germany*

[2]*Univ. of West Macedonia, Greece*

Abstract
This paper introduces so-called Domain models to support semantic interoperability of connected IT-systems. The focus of the paper is to motivate the combination of technical and informal specifications and to present a procedure of modelling independent IT-systems at the semantic level for the purpose of semantic interoperability. The resulting domain models may then be the foundation for systematic mappings between those IT-systems which preserve semantics of both messages and persistent records.

Keywords: Semantic Interoperability, Information Models, Electronic Health Records

1. Introduction

The importance of semantic interoperability has grown in recent years with the emergence of increasingly interrelated information systems. Clinical IT-systems like point-of-care devices, ICU systems, picture archive systems (PACS), radiology/lab information systems (RIS/LIS), hospital information systems (HIS) and financial systems for reimbursement, claims and controlling often have very limited interfacing capabilities. Patient record systems, remote services (e.g. radiology, lab), Health Information Exchanges (HIEs), insurance systems, quality assurance systems and public health applications are typical IT systems external to the hospital and should also be interoperable to each other as well as towards clinical IT systems.

Today patient-related data typically cannot be exchanged between those system and there is a need to create "meaning bridges" that translate content between the various systems and their components. Such exercises have been recently undertaken in large scale, supported by increasingly complex rule-based strategies and validated by resource-intensive consensus building in expert communities.

As categorized by the SemanticHEALTH report[1], there are "three layers of artifact to represent meaning"

"1. Generic reference models for representing clinical (EHR) data, e.g. ISO/EN 13606 Part 1[2], HL7 CDA Release 2[4], the openEHR Reference Model" which are just containers for clinical use and which have options and room for interpretation. There are already specific implementation guides (some with semantic content modules / archetypes) for the use in CDA Release 2 [4] defined by several countries (Germany, Finland, France, Italy, Spain) in the context of their national or regional projects, but no general european consensus exists how to use such containers.

"2. Agreed clinical data structure definitions, e.g. openEHR archetypes, ISO/EN 13606 Part 2 archetypes [3], HL7 templates, generic templates and data sets" and similar global detailed models which in practice still need a lot of explanation and constraining. For example, the collaboration among interested clinician-developers facilitated by the openEHR foundation on the use of ISO/EN 13606-2 archetypes will rapidly produce a large number of maximal data sets based on clinical data structure definitions that are assumed to be globally meaningful in a unique way[3]. This paper questions that detailed models can have a global and precise semantics at the same time.

"3. Clinical terminology systems, e.g., LOINC and SNOMED CT"[6] which again provide very detailed information but which can only have a meaning in a defined context. The various attempts to represent cause-symptom-relationships – as one example of establishing context - are an indicator for that principal defect.

We demonstrate here significant drawbacks of "global detailed models". Due to the "context effect" described in this paper, the "global

The *electronic Journal of Health Informatics* is an international journal committed to scholarly excellence and dedicated to the advancement of Health Informatics and information technology in healthcare. ISSN: 1446-4381
© Copyright of articles is retained by authors; originally published in the *electronic Journal of Health Informatics* (http://www.ejhi.net). This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License (http://creativecommons.org/licenses/by-nc-sa/2.5/au).

model" approach will still allow for different interpretations and therefore is unlikely to be successful. It rather seems appropriate to respond to requirements of a specific field or sector in order to succeed in writing valid and useful models and implementation guidelines.

One indicator is that existing generic, multipurpose (type 1 of the layers shown in SemanticHEALTH), as well as highly structured models (type 2) or classifications (type 3) are being constrained or even altered for their specific use in different local contexts, mostly in the form of national modification of the generic instrument. For Europe in particular, the issue has an additional strong political dimension as cross-border data interchange for the support of intra-union citizen mobility has been described in a number of recent key documents as a cornerstone towards European integration [5].

The language barrier is the first thing that comes to one's mind in such a setting. Today, multilingual representations of clinical semantics do not exist at least to the level required to achieve successful understanding in different settings and regions. Although a number of the semantics are probably already available in SNOMED [6], they have not, for example, yet been validated across a significant number of the languages of the EU.

More important, an identified systematic mechanism for ensuring the consensus required to preserve clinical understanding in different settings at the semantic level does not exist, nor are there any effective collaborative tools for testing multilingual, interoperable clinical semantics across application domains.

What is also missing is a European framework to support multicultural clinical practice in a cross-border environment envisaged by the political priorities, as it has also been recognized by the recent Recommendation of the Commission that calls for specific actions for the Member States in the area of semantic interoperability [7].

This paper first introduces general requirements for semantic interoperability of connected yet independent IT-systems. The focus of the paper is to present a procedure of modeling independent systems at the semantic level for the purpose of semantic interoperability. Such domain models may then be the foundation for systematic mappings which preserve semantics.

2. Consensus

In a recent EU report [8], upon which this paper builds, five different definitions of semantic interoperability have been identified at European level. Many more exist if one moves to the international arena. In this paper, semantic interoperability means the ability of two computer applications A, B to preserve the meaning of information in the sense that information from system A - after communication, storage and/or processing - will be in system B and expressed through terms that are interpreted by humans to "mean the same" (also: "reflect the purpose").

"Common use and understanding of terms" is a practical definition of this view onto semantic interoperability. The word "common" refers to a group of humans who share a consensus on how to use and interpret terms in an IT-system. In this paper, the word "term" should not only include keyboard input and screen output, but also sensor readings, data messages as well as actions performed at interfaces.

Although semantics are inherent in every implementation, users (and in sometimes vendors as well) don't realize their importance until they have to face issues of misinterpretation or simply lack of information. In a cross-border environment that the EU is aiming to, it is easily understood that if information is to be readily available in an interoperable environment, this information needs to be represented at a given point of care in a way that is compatible to the local practice. A multilingual application for example does not simply mean support of more than one languages, it means (self)adaptation in a multicultural (clinical) environment.

Semantic interoperability, especially at the clinical information level, is critical to achieving full exchange, understanding and appropriate consequent action among linguistically and culturally disparate clinicians and other parties in a cross-border environment. In the following of the paper - to illustrate our modeling approach – using the Automatic Teller Machine (ATM) example where international semantic interoperability appears to have reached higher maturity than IT-interoperability within the health field.

Any ATM dispensing money is (considered) - in our words - an "interface" of a world-wide ATM network that produces a number of "terms" e.g. a screen to withdraw money- but also the output of money.

The shared consensus can only be described through a natural-language explanation of how to use and interpret the terms of each interface of a system. The shared consensus for ATMs is in a global understanding of the principles of entering a card and a PIN, selecting an amount and taking the cash. Vice versa, any attempt to explain a formal system of technical terms using yet another technical formalism would not obtain a consensus from a user community: The success of ATMs is NOT based on formal semantics specifications but on the fact that many people understand ATMs in the same way ! Similarly it is not the information model but the consensus among all clinical professionals that is critical to the success of some hospital IT system.

As a result, semantic interoperability is beyond the scope of technical systems, as there needs to be an understanding among human users, that terms "mean the same" as they are used by semantically interoperable systems. As humans think in informal concepts apart from all technical representations, it is important to express the same concepts but not necessarily the same representations.

Figure 1: The Goal of Preserving Concepts through Semantic Interoperability

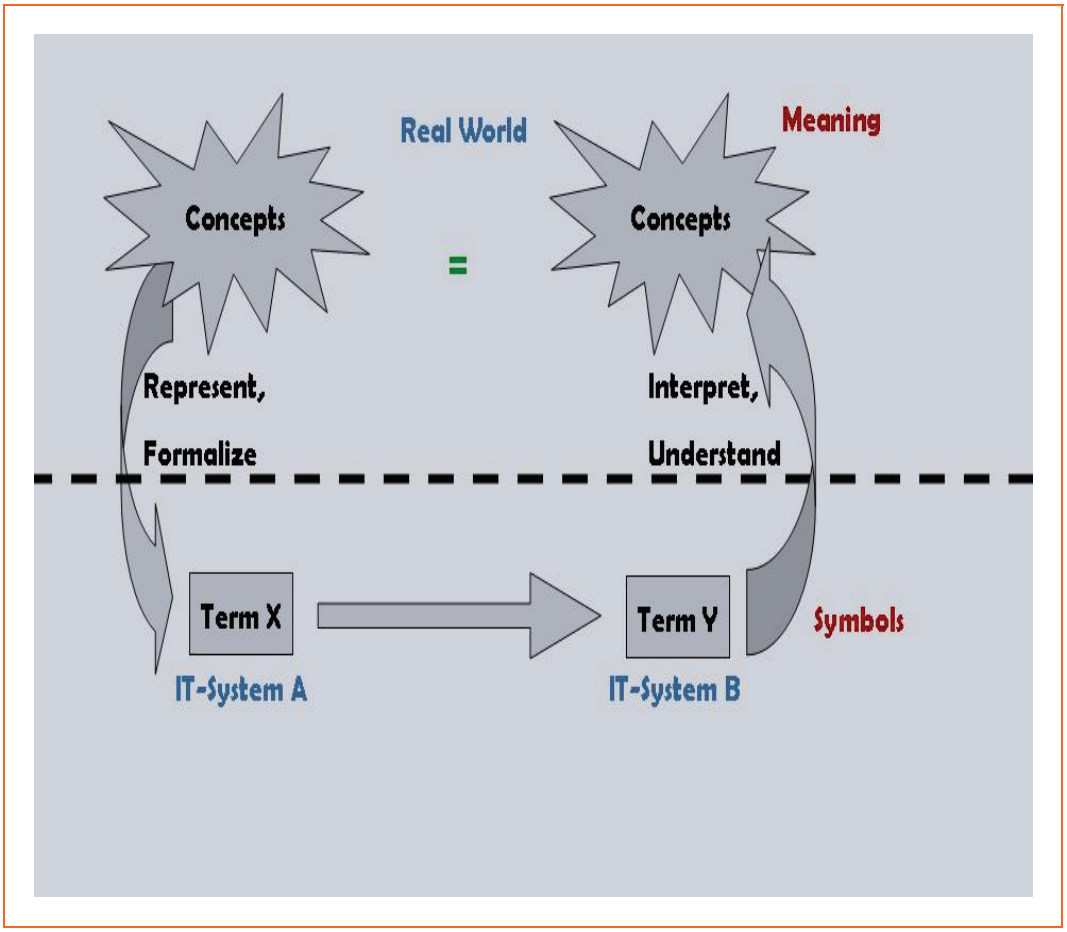

Figure 1: The Goal of Preserving Concepts through Semantic Interoperability

Thus, we can give a more focused definition: Semantic interoperability is the property of two systems, A,B to be able to map "meaning" represented as terms in system A - after storage, communication, processing from terms in system B into the original conceptual information ("meaning", see Fig. 1) [9].

3. The Context Effect

Any term within a model (archetype, reference model, terminology) has a meaning. However, the meaning of a term is rarely mono-sense. The context in which a term is used many times alters or even negates its meaning. As an example, the letter 'A' appended to a given ICD-10-Germany code is a modifier notifying that a given diagnosis can be "excluded". Not processing that letter may cause serious misunderstandings and safety risks. This indicates that the additional (surrounding) information may qualify the meaning of a term in a model or terminology. If "X" is a certain lab test, then an "order to perform" X does not mean the same like "confirmation to have done" X.

This is the "context effect": the meaning of any a term is not absolute, but depends on any related information, because it is the context environment which constrains the meaning of a term. Many other examples can be found, some of them [6] include "site" (of patient), "suspicion" (diagnosis), purpose such as "intention", "order", "promise", "completion", "forwarding", "refusal" (procedures) and status "received", "checked", "scheduled", "failed", "cancelled", "completed", "caused_by" (or, "symptom_of" that generally creates a whole network of terms for findings and diagnoses).

Note that at a time more than one such "context effect" may exist in parallel and the meaning of practically every (modeled) term is depending on it. Practice and experience (e.g. the ever growing messaging models (*R-MIM*s) of HL7 v3 and their claim to be global and precise at the same time) has shown that no model or terminology can foresee all possible influential real world information. There is always influential real world "context" that adds a factor affecting the meaning and interpretation of a term in any model. Although some

semantic model can always be extended to include a context factor, as already explained, even more context factors will turn out to be relevant, such that there will never be a "final" model and the whole exercise will result to an endless loop process.

This effect of the dependence of meaning on the specific application context ("context effect") makes it unlikely that a realistic "pan-european common eHealth information model" can be created, as has been claimed possible elsewhere [5, 6] and has been attempted by several European research projects [10, 11].

4. Contracts

In systems of some reasonable size, semantic interoperability needs informal user documents explaining the elements of related technical interfaces to humans [8]. Speaking more precisely, Semantic Interoperability requires an interpretation from technical terms of the interface into real world concepts – specific to care settings, organizations and legislations.

Semantic interoperability of IT-systems therefore requires a sort of agreement - written in natural language - to define the meaning of terms and interactions of such interfaces. In practice, a natural-language agreement document (or: a set of documents) is a common way of explaining the usage and meaning of technical terms at the interfaces of IT-systems.

In case that legally binding actions shall be supported, a "Contract" is the foundation of the common understanding of terms to be used by IT-systems with different contract parties (See Fig. 2). The concepts of e.g. mobile communication or cash dispensers share a broad consensus all over the world - which makes it relatively easy to design semantically interoperable applications.

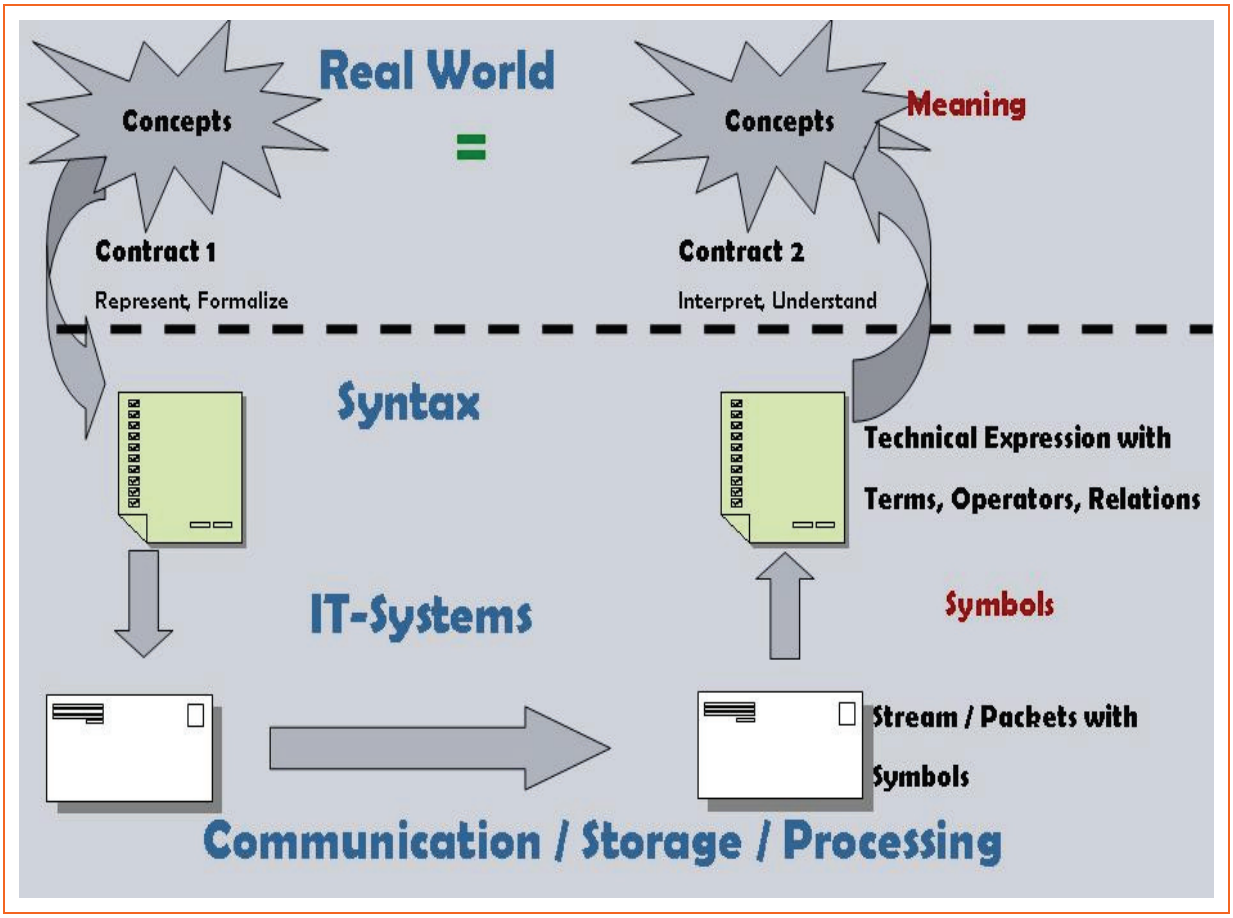

Figure 2: Semantic Interoperability with IT-Systems

The network of cash dispensers (ATM) is an example of a technical system that is understood by a very large number of people, though it bridges organizations, nations and languages. Among ATM users, there is a huge - yet informal - consensus, what the terms ACCOUNT, WITHDRAWAL, PIN and CASH mean in the real world (while other statements used in banking turned out not to be so certain). Independent of that broad consensus (which accounts for their success), the ability of successful IT-applications (ATM networks, say) to semantically interoperate in fact depends on comprehensive, detailed, yet natural-language "Contracts" for all interfaces of the respective application.

The understanding of terms in a large group of users may contribute to the acceptance and success, but whenever there are disputes, people will refer to the "Contract" to clarify a situation.

It is a difficult and time-consuming process to find (and document) the required consensus and agreement between (international) experts to identify necessary meaningful concepts, define terms and, the interpretations between both when creating the required “contract”. Though being simple with respect to usage and operation, achieving this common understanding and expressing it as a “contract” is by no means a simple task.

5. Domain Models

Information models serve as a reference from which terms, relations and expressions of IT applications can be explained via the required natural-language contract. Types of technical models are static models (class model, component model) as well as dynamic models (sequences, state transitions).

The main benefit of using information models is that implementations can be derived systematically - and often automatically - from detailed information models. This may improve reliability and safety of the implementation as well as reduce application development efforts [12].

In cases where a model class can have one of list of values, a terminology may be used to “import” semantics that have been defined externally and independent of a specific application. Such “controlled vocabularies” are like reusable building blocks of semantic interoperability. The respective vocabulary publishers should ideally be a recognized user group or at least involve users and domain experts.

A single technical information model defining interfaces and internal states (including storage) - within a healthcare domain and specific to a region - would be useful. Such an information model defining relations, attributes and meaning for classes of concepts in a whole domain is called a Domain Information Model (DIM).

DIMs help in formally constructing and analyzing statements of an application domain by explicitly describing the concepts of that domain via the formal classes, attributes and relations. In order to actually create specific and precise semantics, any DIM needs to be extended by a set of plain text explanations of all its model elements (“Domain Contracts” as in Fig. 3).

Like with other formal models, these could be a “Model Contract” - for generic “alphabet” model elements- together with an “Application Contract” for the specific model elements. It is important to note the vital role of that "plain text" in between the formal model and the human understanding of concepts. Also note that applications implementing some functions on top of a DIM must find a way to technically represent the classes, attributes and relation of the model – because high-level models like those of UN/CEFACT ebXML “Core Components”(or the HL7 v3 R-MIMs as such) do not describe a technical implementation [13].

Using a domain model also makes establishing the interface-specific “Contract” more systematic, in that an overarching “General Domain Contract” (better: “Implementation Guideline”) document may be created – but still there has to be the informal part in it – with explanations referring to concepts of the real-world around the system.

Adding a “semantically interoperable” application to a system’s interface just requires to read and implement the explicit informal domain model explanation (“Domain Contract”) – from the new application’s perspective. A domain information model - together with its informal explanations - gives a guideline to representing new concepts technically and also helps the users of these concepts to have a common understanding of information by providing interpretation templates into the terms (used by the model).

Complexity in healthcare requires implementers to publish and standardize explicit, written natural-language explanations, which can easily be written as “Implementation Guidelines” and which constrain available standardized formal domain models. In an optimal case, an application may directly use a given DIM without modifications, while in most cases further constraints and a modified contract will be required.

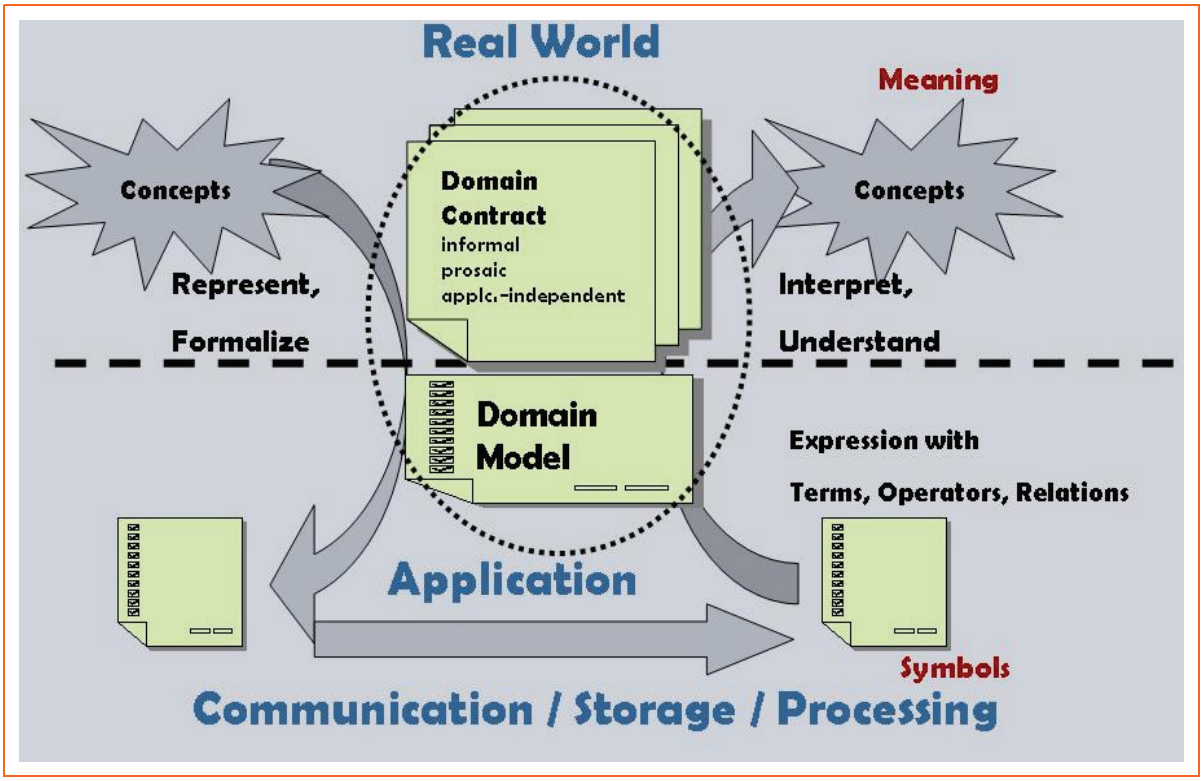

Figure 3: Using a Domain Information Model (DIM)

In the more general case, a specialized DIM had to be established for one or multiple specific applications. The Reference Information Model ("RIM") and the design methods of HL7 version 3 are examples for both foundation classes plus a design and refinement method which together provide a very generic and powerful approach to versatile DIMs.[3] The RIM's expressive power can be used without using of the published domain models of HL7 v3 which lack the specific contract and therefore fail to define precise meaning.

6. Establishing Semantic Interoperability

There are various factors which make establishing a domain model and writing the required Implementation Guideline a challenging task:

- different use of established clinical terms
- the fragmentation of medical specialties
- the different health care stakeholders (payers, authorities, different kinds of providers)
- the context of different societies and languages
- different legislations
- political decisions at EU level

As already explained, the "context effect" prohibits the "global model" strategy of being successful. Therefore, it seems wise to respond to requirements of a specific field or sector in order to succeed in writing valid and useful models and implementation guidelines. In order to establish an eHealth domain model together with its informal explanations, the domain community (with the input of clinical users) needs to agree on "contract-like" definitions for data structures for exchange, each of their terms, relations and procedures, so that all IT systems consistently derived from that model are semantically interoperable. This is a huge endeavor, and would need organizations and policies beyond a specific region or organization to define such model-based data and corresponding coded values. Even within the same eHealth network, such a task is difficult to achieve because it needs to involve all stakeholders.

As the usage of application specific context information is essential to selecting and constraining domain models, it is clear that use-cases have to be described prior to establishing domain models. Therefore, one basic task is to define a limited set of related use-cases in order to precisely describe the application perspective, another task is in identifying and motivating the right users to help clarifying the terms in these use-cases.

Therefore, a formalized information model (regardless whether it describes just one interface or a whole application domain) shall go along with a related natural language "contract" - explaining the formal terms of that model. Note that the informal model "contract" of course shall be

precise – as informal does not mean sloppy. Semantic interoperability will then be possible among the consensus community respecting (or complying with) that contract document.

From an application developer's perspective, both the model as well as its contract document are equally important and should of course be made available to interested parties.

7. Deriving a Domain Model

We outline a new way of modeling existing applications in a systematic way. The idea behind our approach is that identified "mirror" objects – reflecting persistent objects managed by the various connected IT-systems - serve as the glue between systems as they carry all secondary keys to identify "the same object" in a different connected system. Mirror objects will be required for everything that has an identity (or, a primary key) and which plays a role in some message.

Whenever a message needs to be translated from one system to another, such mirror objects (as specified in the Domain Model) will be created and their attributes will be filled. Despite (secondary) key attributes the mirror instances will only have to carry changed values for modified attributes (also: new values for all attributes in newly created objects). Then all translations towards other connected IT-systems will be derived from these mirror objects and the remaining transient information.

The procedure basically consists of identifying persistent candidate classes and then unifying the "same" candidate classes of all the connected IT-systems into new Domain Model. Note that in this chapter, the "hyphen" notation refers to an instance of a subclass, i.e. "Object" means: instance of a subclass of the persistent class Object.

7.1. Starting a Domain Model

At the start, each relevant message of one selected IT-system A is analyzed and reflected as a new "Function" candidate class in the model. Messages reference or copy none, one or multiple persistent objects. So, from each message, all attributes (including those that may act as a key) of persistent "Objects" are carved out and defined as a part of that message's new candidate "Function". There are two ways to do that: If data in that "Function" may be semantically mapped ("the same concept") the attributes need to be mapped to existing or new attributes in that existing "same" object. If the persistent "object" found does not match an existing concept, then a new subclass of Object has to be created and all related attributes will be part of that new "Object". The key attribute that may act as a primary key in A for that new "Object" has to be identified. Each new "Function" will have one relationship towards each "Object" that it identifies. Identifiers of existing "Object" in some other "Object" are converted into a (directed) relationship in the Domain Model. The resulting class model will have one persistent part consisting of Object subclasses and their relations. The transient part consists of one "Function" per message containing all transient attributes that are part of the message but not stored in a persistent "Object". There will be no relations between "Functions", but of course between "Functions" and the required "Objects".

For each message, the whole mapping of message parts onto attributes of classes in the Domain Model has to be documented.

7.2. Merging more Applications

The persistent part of an existing Domain Model of application A defines the concepts for repeated application of the above procedure, when some application B is being merged: Concepts in "Object" candidate subclasses that are semantically "the same" as existing concepts in domain model A have to be unified. That "sameness" can only be determined using the respective "Contract" (or, "User Manual") coming with applications A and B - interpreted in the context of the intended way of interoperation of A and B.

As a result, the model will contain more and more attributes including conversion rules for deriving dependent attribute values. In the "Objects" it will also have to manage secondary keys for the use as identifiers in each connected IT-system. In many cases, the values for such keys can only be obtained from some other connected IT-system such that there is a required sequence of connections to the different IT-systems in order to systematically obtain all necessary keys. Note that the attributes of mirror objects must be capable of distinguishing "no value" from zero.

7.2.1. Gateways

One way of connecting IT-systems is by making one IT-system A "lead" other system who have to process the messages initiated by A. The gateway-type of interoperability would consider one "leader" IT-systems which always triggers the processing of messages towards "follower"-systems. Our Domain Model then can be interpreted as a collection of "Functions" which each describe the translation from an incoming "leader Function" message into the model, and the creation of semantically equivalent messages for the same "Function" towards each connected "follower" system.

7.2.2. Portals

A different way of connecting IT-systems is by creating a "portal" which accepts messages from a human user or remote IT-systems with all connected systems "behind the portal" having to consistently process the messages sent to the portal. The portal-type of interoperability considers an external service interface and an external operator as the trigger for service-"Functions" - with the Domain Model describing the creation of messages towards all connected IT-systems. The list of Function subclasses in such a portal

system could be seen as the set of high-level functions offered as services via the portal.

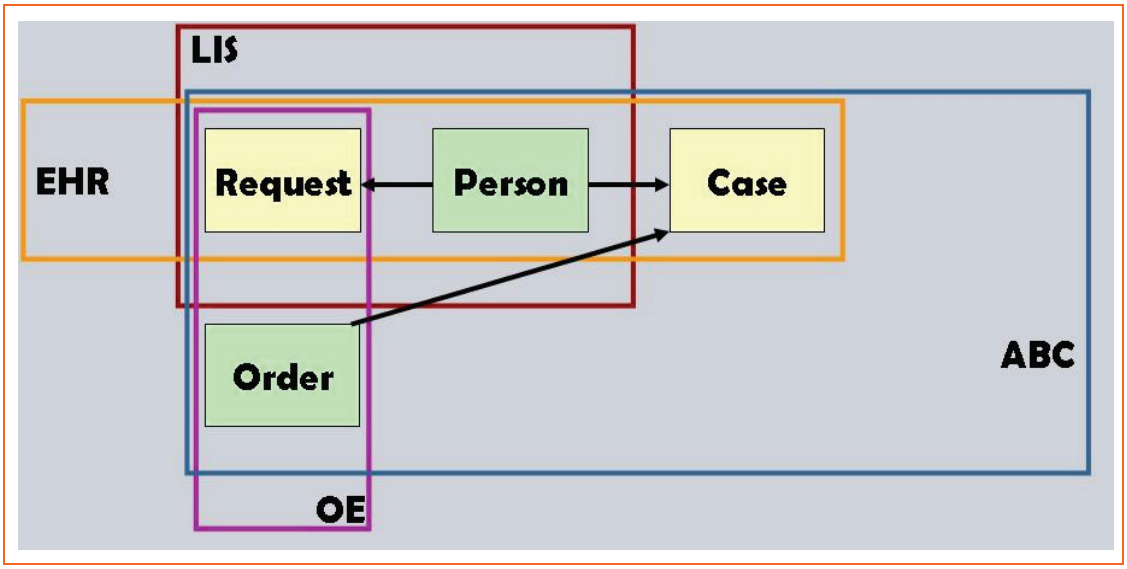

Figure 4: An Example Domain Model

7.3. An Example Contract

The text of this section is a simple contract for interoperability among four connected IT-systems (See Fig. 4):

A secured web-server accepts remote order entry (OE) from external systems connected to a local electronic health record (EHR), and also to a lab information system (LIS), and also connected to a accounting/billing/controlling system (ABC). One message of the OE server would mention the concept of a "Person" with some combination of attributes acting as a primary key – as the identifier provided externally may be from various independent remote systems. The LIS also knows the concept of "Person". So messages to/from LIS have to mapped to the Person class. LIS may only process messages with patient identifiers that are known to EHR, but that EHR system will provide such identifiers only as a response, i.e. in a subsequent step. ABC will not accept any external identifiers but will always create its own "case" identifier (unless a "case" is specified) and expects that case identifier to be used for one single clinical encounter. The "Person" will therefore first be identified by auxiliary attributes and later be updated with a useful patient identifier attribute value. The "Case" identifies a different object, that – in the Domain Model - is related to "Person" in a many-to-one relationship. In practice, the EHR system would decide whether to use an existing case for a known person or to specify a new case (without case identifier). Requests taken by OE but not assigned to a case have to be modeled apart from Orders that are assigned to a case by EHR Each message from/to ABC will have to link exactly one case instance from a given person instance (Instance constraint of the model).

8. Conclusion

A modeling technique for analyzing application interfaces has been explained in order to support subsequent semantic interoperability. Systematic mappings from messages to the proposed domain models can be established based on the above analysis procedure. The inverse mapping from the Domain Model to messages then maps each "Function" to a technical message, giving a systematic way to compose messages out of data possibly delivered by some other application. On top of some required technical integration, an implementation of such forward/backward mappings will then provide interoperability at the semantic level, with such mappings practically being implemented by integration engines.

Three observations from our example:

- The exam Request and the exam Order are conceptually close but have to be different classes only because they have different relationships within the Domain Model. At one time during message processing, an Order has to be created by copying values from the Request, yet using totally different identifiers. Does the original Request have to be recorded persistently ?
- The decision when to use a "new case" and when to re-use an existing case can not always be automated. In our example, EHR (and its operators) have the capability to specify a new or an existing case.
- The choice of auxiliary key attributes for Person determines

his/her fate: If they are too fine-grained (like e.g. timestamp of incoming Request) multiple Person instances for the same individual would be in the system – hiding access to earlier exam results which technically belong to a different Person. If they are too coarse-grained, different individuals would be merged in the system, possibly exposing wrong diagnoses in a person's record. As merges are way easier than un-merges of person's identities, EHR systems typically work with fine-grained identifiers and manage "soft-links" between records or between Case and record.

The essential foundation for meaningful and effective models however are the informal "Contracts" for the use of the IT-systems to be interoperating. This paper demonstrated that both "legacy" as well as brand-new applications may interoperate at a semantic level using a common approach based on technical domain modelling together with natural-language "contracts" which explain elements of each respective technical model.

It is clear, that the strength of the modelling technique and a good foundation of predefined, reusable elements (e.g. the set of HL7 v3 domains) supports the creation of domain models for interoperable systems. In cases where the predefined meaning of modelling elements (together with their prosaic interpretation !) suffices in defining an application, constructing semantic interoperability will be straightforward and systematic.

Acknowledgements

We thank Bernd Blobel, Christoph Dickmann, Anand Kumar, Kees Molenaar, Charles Parisot, Melvin Reynolds, Rene Spronk and Sylvia Thun for helpful discussions on the topics of this paper.

References

1. EC Directorate INFSO. SemanticHEALTH Report, Brussels, 2009, ISBN 978-92-79-11139-6 see http: // ec.europa.eu/information_society/activities/health/docs/publications/2009/2009semantic-health-report.pdf

2. ISO. ISO 13606-1 Health Informatics - EHRCOM part 1: Reference Model, Geneva, Switzerland, 2007

3. ISO. ISO 13606-2 Health Informatics - EHRCOM part 2: Archetypes, Geneva, Switzerland, 2007

4. Health Level Seven Inc.. HL7 v3 RIM. Ann Arbor, MI, USA, 2006http://www.hl7.org/library/data-model/index.cfm, see also: ISO 21731

5. EC Directorate INFSO. epSOS FactSheet. European Commission, Brussels, 2007. http://ec.europa.eu/information_society/newsroom/cf/item-longdetail .cfm?item_id=4113

6. IHTSDO. SNOMED-CT. Copenhagen, 2008. see http://www.ihtsdo.org/snomed-ct

7. EC Directorate INFSO. COMMISSION RECOMMENDATION of July 2nd 2008 on cross-border interoperability of electronic health record systems, Brussels, COM(2008)3282 final

8. Molenaar G C. eHealth-INTEROP Report, EU Mandate/403-2007: Phase 1, Annex C: Semantic interoperability. NEN, The Netherlands, Dec. 2008, http://www.ehealth-interop.nen.nl/

9. Heidenreich G, Dickmann C. Semantic Interoperability in eHealth, In: Blobel, Pharow, Zvarova, Lopez, CEHR 2007 Int. Conf. Proc, IOS Press, Amsterdam, 2007 pp. 89, 978-1-58603-834-2

10. Rector A . Barriers, approaches and research priorities for integrating biomedical ontologies: EC Directorate INFSO. SemanticHealth D6.1, Brussels, 2008

11. Smith B. Referent Tracking: Towards Semantic Interoperability and Knowledge Sharing. In Proc. eHealth Conference 2007. GVG, Cologne, 2008. ISBN 978-3-9811491-3-5

12. Object Management Group. Unified Modeling Language. Needham, MA, USA, http://www.uml.org/

13. UNECE. UN/CEFACT ebXML Core Components V2.01, 2007. http://www.unece.org/cefact/

Correspondence

Dr. Georg Heidenreich
Siemens AG Healthcare / H QT S&T 2
Henkestrasse 127
D 91052 Erlangen, Germany

Phone: +49 (0)9131 84 6498
Fax: +49 (0)9131 84 6555

georg.heidenreich@siemens.com