Annex C
to
eHealth-INTEROP Report
Semantic interoperability
in response to
eHealth Interoperability Standards Mandate

(SA/CEN/ENTR/000/2007-20 eHealth Mandate M/403 – Phase 1)

http://www.ehealth-interop.eu

Document History:

Annex C, Semantic interoperability, to SA/CEN/ENTR/000/2007-20 eHealth Mandate M/403 – Phase 1 Report

Document Location: *http://www.ehealth-interop.eu*
Validity: *From date of publication until approval by ESOs, EC and EFTA*.
File name: *ESO_eHealth-INTEROP_AnnexC_v1000.doc*

Change History:

Date	Version (n.Nrr)	Changes file name format: ESO_eHealth-INTEROP_C0Nnn_CCYYMMDD.doc
2008-11-26	0.600	Post comment editing started.
2008-12-22	0.900	Post-comment editing completed
2009-02-10	1.000	Final version approved by CEN, CENELEC and ETSI

Editorial project team:

Pantelis Evangelidis,
Georg Heidenreich,
Charles Parisot,
Melvin Reynolds (lead editor).

Please submit any comments on this document to:

Ms Mary van den Berg Standardization administrator	mary.vandenberg@nen.nl Tel.+ 31 152 690390 Fax + 31 152 690190	NEN-Healthcare Vlinderweg 6 NL-2623 AX Delft
Ms. Shirin Golyardi Standardization consultant	shirin.golyardi@nen.nl Tel.+ 31 152690313 Fax + 31 152690563	

Contents

Main Report, Summary of Report and Annexes A, B & D available from http://www.ehealth-interop.eu

C.1 Semantic interoperability considered

C.1.1 The Goal of semantic interoperability

C.1.1.1 Definitions

Semantic interoperability carries many different definitions. In Europe, it has been defined as:

- "to ensure that meaning of exchanged information is understandable by any other application that was not initially developed for this purpose" [European Interoperability Framework, http://ec.europa.eu/idabc/servlets/Doc?id=19529 , Brussels, 2004]
- "the major enabling factor for the safe and sensible communication of patient data." [SemanticHealth, IST-2005- 027328, Brussels, 2005]
- "the ability of two or more computer systems to exchange information and have the meaning of that information automatically interpreted by the receiving system accurately enough to produce useful results, as defined by the end users of both systems." [Wikipedia, 2006]
- "the ability of systems to 'understand' and process the information exchanged." ["A Roadmap for Interoperability of e-Health Systems in Support of COM 356 with Special Emphasis on Semantic Interoperability, RIDE, Brussels, 2007]
- being "achieved when the meaning of exchanged information is understood by applications and services. Achieving semantic interoperability, i.e. mastering semantic heterogeneity, today is seen as one of the biggest challenges for the integration of information systems. Basically, this is due to the fact that meaning changes by context and over time and different requirements in different domains result in different information models" [QUALIPSO, http://www.qualipso.org, Brussels, August 2008]

Fig. 1 describes the claim: At different end-points (interfaces) of IT-systems the meaning i.e. information about the real world (upper half) needs to be re-constructed to the original concepts. The terms used by IT-systems (lower half) do not have to be the same, but their interpretation should match in the real world.

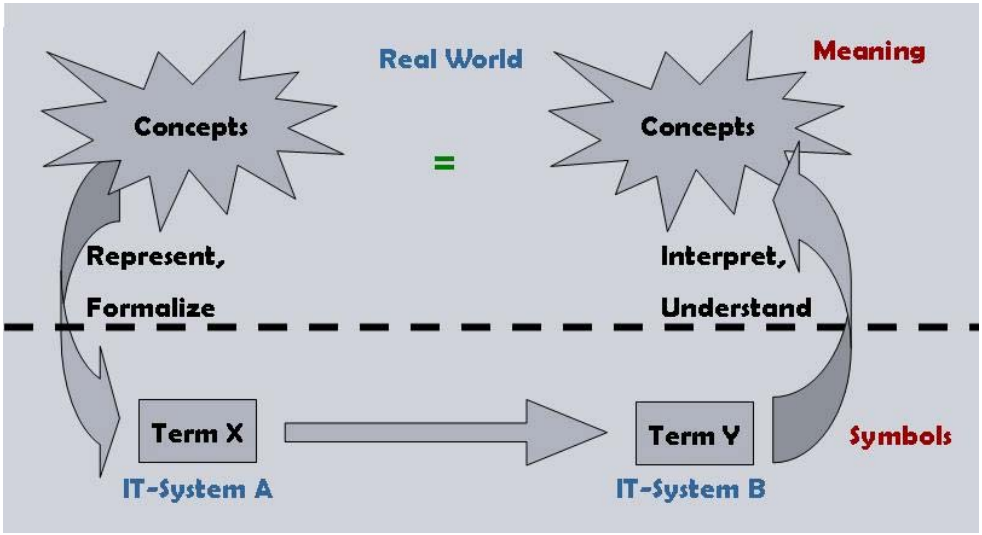

Fig. 1: The claim of Semantic Interoperability

What we are looking for, is beyond the scope of technical systems, as there needs to be an understanding among human users, that terms used by semantically interoperable systems "mean the same".

C.1.1.2 A working definition

"Common use and understanding of terms" is a pragmatic definition of 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 chapter, 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. So, the ATM dispensing money is - 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 on such terms can be either implicit or explicit, but in both ways it is the natural-language explanation of how to use and interpret the terms of an interface that makes up the foundation of semantic interoperability.

Semantic interoperability needs documents explaining the related technical interfaces to humans. Speaking more precisely: “Semantic interoperability requires an interpretation from technical terms of the interface into real world concepts”.

Banks typically include such explanations (interpretations) in their "Business Terms" - since the operation and behaviour of their ATMs is an important part of their business and their customer contracts. Semantic Interoperability of IT-systems interfacing towards ATM networks therefore require a sort of “Business Terms” – written in natural language – to define the meaning of terms and interactions of such interfaces.

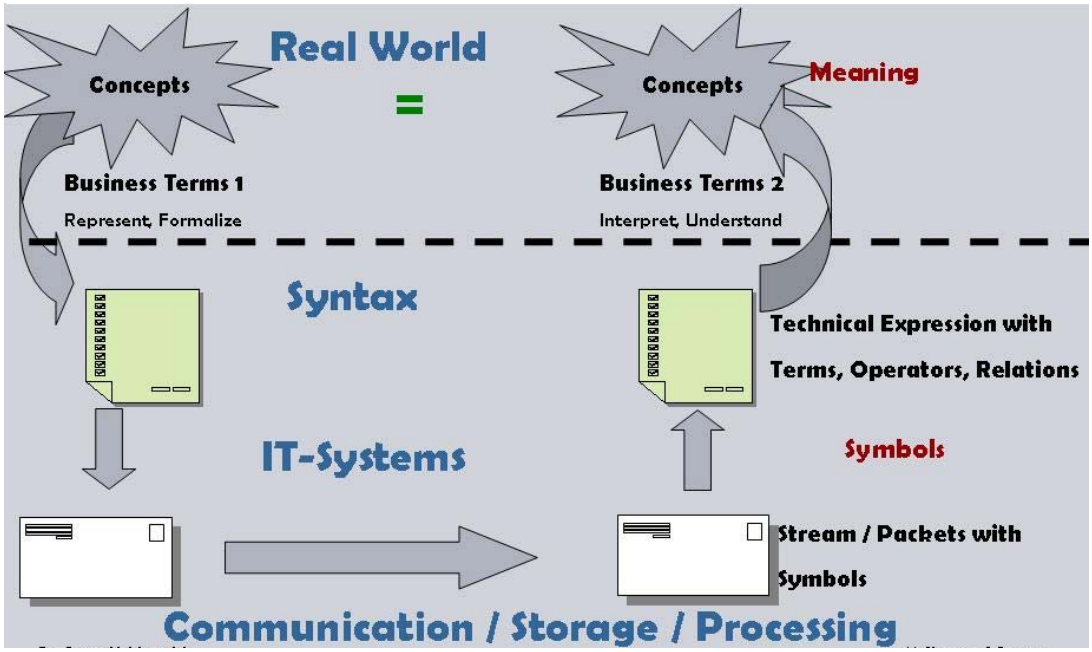

Fig. 2: Semantic Interoperability regarded as equality in the conceptual space.

In this report, semantic Interoperability means the ability of two (IT) systems A, B to preserve the meaning of information in the sense that information from system A - after communication, storage, processing - will be in system B expressed through terms that are interpreted by humans to “mean the same” (also: “reflect the purpose”).

- Semantic interoperability
  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”)

In practice, a natural-language “Business Terms” document (set) is a common way of explaining the usage and meaning of technical terms at the interfaces of IT-systems. In a more technical domain, “Reference Manual” would be a typical title for such an interpretation.

C.1.1.3 Syntactic Interoperability

(Syntactic) "Interoperability means the ability of information and communication technology (ICT) systems, as well as, of the business processes they support in order to exchange data and enable the sharing of information and knowledge" [IDABC, http://europa.eu.int/idabc/, EC, Brussels, 2005]

Syntactic interoperability, which also deals with applications A, B – such that application B is able to fully process the output (structure) of A (and possibly vice versa). Syntactic interoperability is a prerequisite to semantic interoperability, but does not have a defined interpretation (like e.g "Business Terms") into real-world concepts. Semantic interoperability is more - in that B is required to map all output of A to internal terms which have the same meaning – based on the "Business Terms" of A and B.

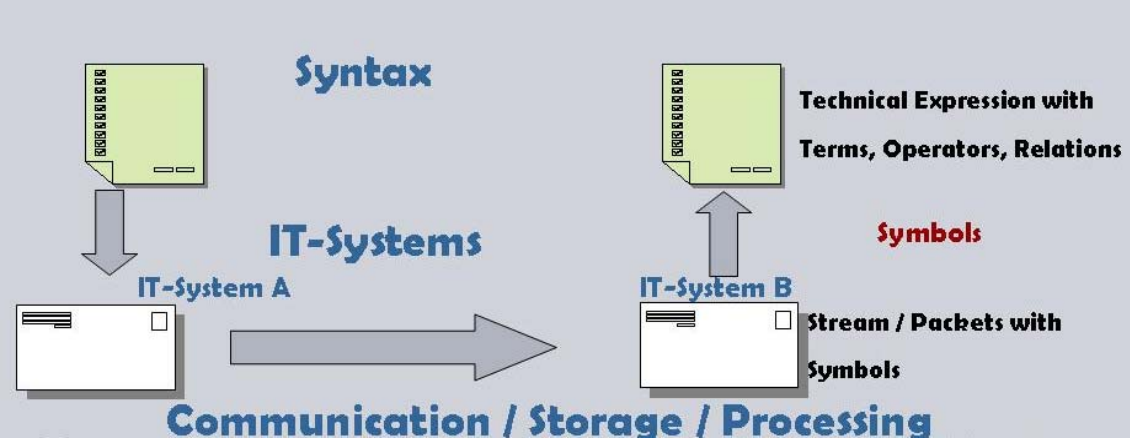

Fig. 3: Syntactic Interoperability

Experience from integration projects in large hospitals shows that the classes and attributes of formal models which do NOT emphasise an additional informal explanation (which is the case with syntactic interoperability), are often misused as "containers for something" which just needs to be communicated in a given situation. As integration is continued, the use of classes and attributes becomes more and more disconnected from their original intention which gives rise to an IT-Hydra that creates more problems whenever a bug is fixed. We claim that the lack of good and comprehensive explanations is one main cause for what might make IT systems integration unmanageable.

C.1.1.4 Implicit consensus on meaning

The concepts of banking applications and also mobile communication share a broad consensus all over the world - which makes it relatively easy to develop semantically interoperable applications. The networks of cash dispensers (ATM) is a simple example of a technical system that is understood by a very large number of people, though it bridges organisations, 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 to be not so certain).

We consider this consensus the main difference between health care IT and banking IT, because for most users, the semantics for banking may be left implicit. In the majority of cases, no explicit representations of obvious terms and their concepts are needed.

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 informal "Business Terms*" for all interfaces of the respective application. The implicit understanding in a large group of users may contribute to the acceptance and success, but whenever there are misunderstandings and disputes, people will check the "Business Terms" 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. Though being simple with respect to usage and operation, achieving this common understanding world-wide is by no means a simple task from a technical point of view. Behind the scenes, experts in the banking field, did a lot of work to exactly define the meaning and use of those terms - via lengthy and detailed natural language explanations which now make up the "Business Terms" for ATM users.

Fig. 4 gives an example of a domain model (grey boxes in the middle) plus messages of two applications (left: ATM in USA, right: semantically equivalent message in the bank related to the account shown. Note that there is no explicit interpretation ("Reference Manual") but most readers guess the meaning from an implicit understanding of the terms. Still there are some conversions, not only the format conversion for the date string, but also the currency conversion. which may include a hidden fee.

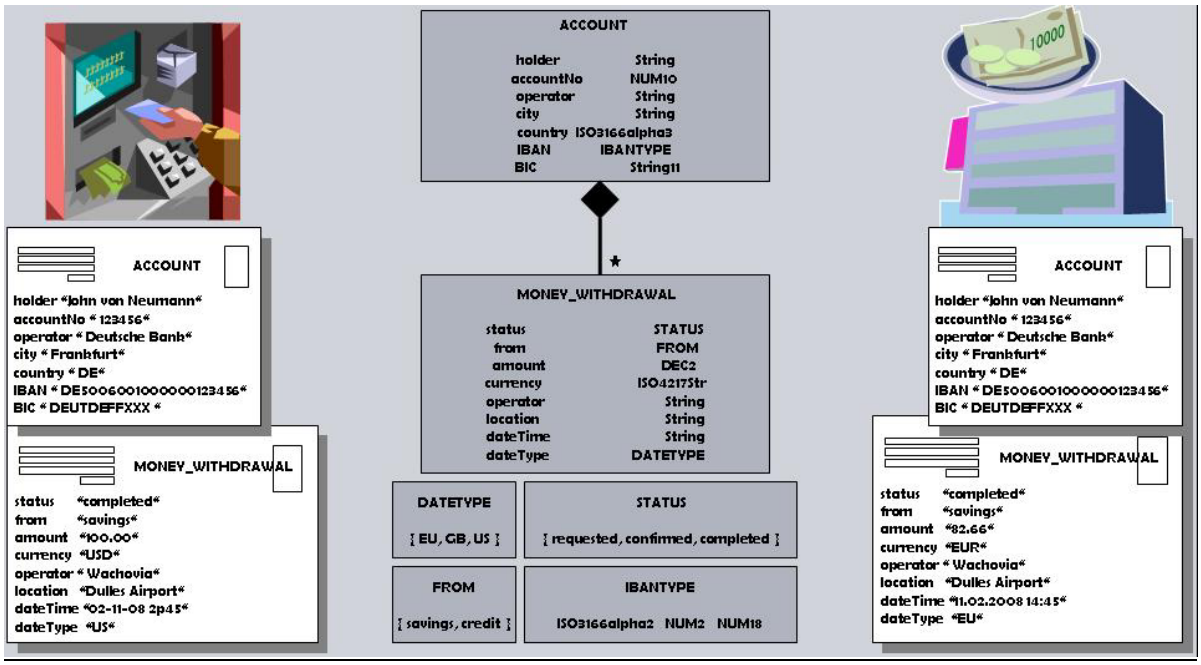

Fig. 4: A domain model with implicit interpretation

C.1.2 Clinical terminologies

One example of semantic interoperability is the use of clinical terminologies within a data structure. Each code ("term") from the terminology represents a clinical concept. The interpretation from terms to concepts is the natural language explanation of the terminology, which in the case of e.g. ICD-10 is independent of IT-applications. Therefore, two IT-systems only need to preserve the term, as there is one common explanation of each term.

There are some criteria for the quality of terminologies, such as completeness, no remainder classes ("other"), a unique and invariant interpretations from terms to concepts and the inclusion of more general concepts.

In so-called post-coordinated terminologies, modifiers allow to alter the basic interpretation of a term [e.g. modifiers R, L, B for locations left, right, both as a suffix to ICD-10 codes]. This allows a terminology express a variety of meanings with few terms only. Other terminologies, which have every concept explicitly defined as a term, are called "pre-coordinated" terminologies".

Large clinical terminologies define each term as the combination of several "independent" attributes, such as the main SNOMED-dimensions "topography, morphology, aetiology, function, disease, job, procedure". Each of these dimensions has its own generalisation hierarchy. [www.IHTSDO.org, Copenhagen].

There is a community of experience for terminology development as a collaborative process. There are even tools available to support such collaborative development. Since information models could be regarded as "terminologies with structure", the development of models, can be performed by similar processes, maybe with support from the same tools.

It should be said that, a pragmatic implementation of a terminology always assumes a perspective or purpose of the respective user. Therefore, while terminologies can be successfully used in limited sub-domains, e.g. laboratory codes or mammography findings, it does not seem realistic to dictate the use of an all-encompassing “final” code set or terminology for the whole world of healthcare. This justifies the proposed method of looking at the use-case first an then deriving such technical measures as a clinical terminology.

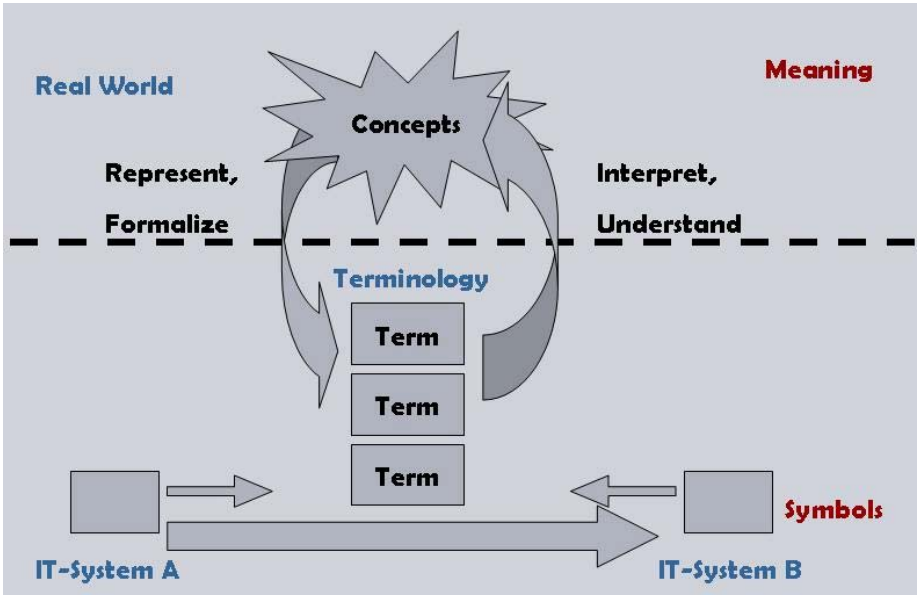

Fig. 5: Using a Terminology

C.1.3 Formal models

Maintaining and extending software in semantically interoperable applications becomes much more systematic, once a formalised model representing the rules and meanings is being used for each interface. The “Business Terms” will then be described by the model, in a formal way. In turn, there need to be “Model Terms”, i.e. informal explanations of the modeling elements in addition to the specific domain model. Furthermore, the interpretation of a single model element (via the “Model Terms”) is less specific than its actual use in a given model. Vice versa, an element within a specific model has more meaning than that model element type in the related model legend. As a result, there has to be an additional specific interpretation and usage definition of the model, another “Business Terms” document, mostly called “Implementation Guideline”.

In other words, a model is necessary for systematically developing and maintaining interoperable applications; nevertheless a model will always require an additional level of explanation of the meaning assigned to concepts. Generic models can be defined in base standards. Although explanations can be done directly at the implementation level (but then interoperability is lost), it only makes sense to provide them in an intermediate level where the common space within an application domain is identified. This intermediate level may be referred to by many different names, profiling being the one used in this document. Identifying the common space (many application domains) is by no means an easy task, but wide consensus is now recognising that ithis effort has to start from requirements. These can only be user requirements turned into application requirements (“Business Terms”). – known as the process of use-case definition.

Using only formal means - without any informal text - a given concept (a “thought”) itself can never be defined (captured by a formal model) such that it is used and understood in an identical way at all times and by everyone. Opposed to the claim (“our model captures semantics”) which designers of

information models are trying to make: Any attempt to link a “concept” to formalised semantics – e.g. by defining a formal class “Patient” within a domain model - will result in introducing yet another symbolic language which potentially may be associated to different concepts - such that it may have multiple different meanings depending on the person using that language. So there always has to be an informal text (an explanation note) which has to address related experience known to a user community in order to describe the usage and interpretation of terms.

It is therefore clear, that the model itself is not part of the “real world of meaning” – the conceptual space - but it remains in a symbolic space. Again, an informal explanation (Implementation Guideline) of the model to interface users is required because the model terms do NOT express “meaning and purpose” but remain in the symbolic (technical) space.

- A formally defined (or, measurable) model that captures concepts through formal terms and expressions, does not belong to the conceptual space: It is yet another language of symbols making up an additional syntactical layer.
- A model can prove „equality“ in the symbolic space: one easy way is to check whether two different systems can map the same piece of information to the same model element (or, expression). (see Fig. 6)

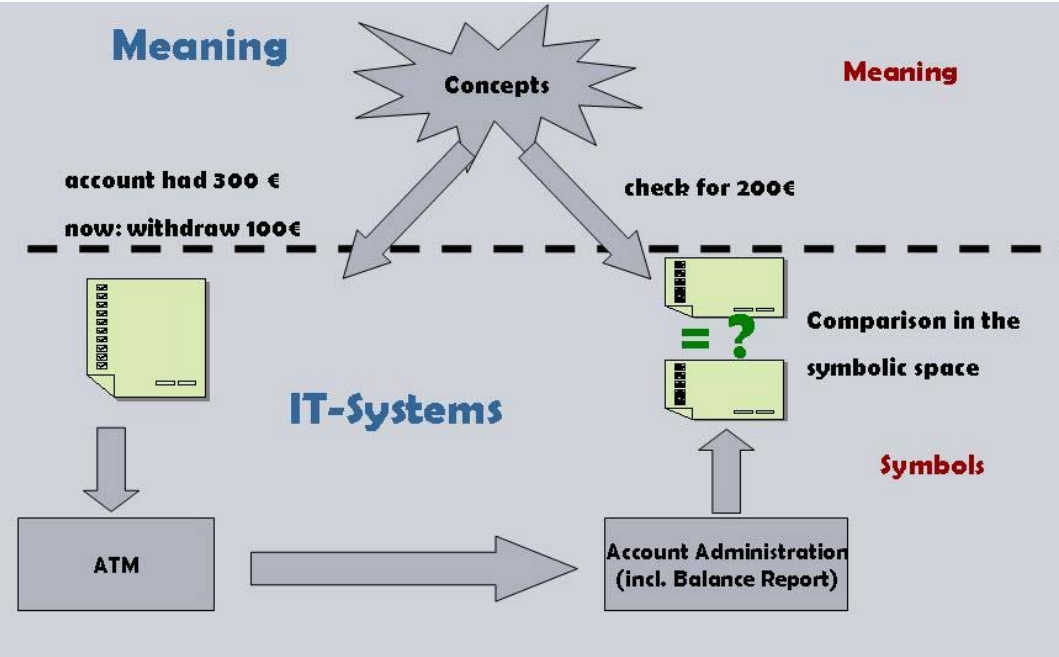

Fig. 6: Checking interoperability in the symbol space

All users (more precisely: programmers towards interfaces) will have to get familiar with the modeling technique (“Model Terms”) prior to using the model. This typically is fine with engineers, but in many cases chases the users (being the intended audience) away. But as long as users agree on whatever set of definitions for each of the model’s terms, relations and procedures, all IT systems consistently derived from that model will be semantically interoperable.

C.1.4 Domain models

A common formalised information model defining interfaces and internal states (incl. storage) of an application domain via relations, attributes and meaning for classes of concepts 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 operational semantics, any DIM needs to be extended by a set of plain text explanations of all its model elements (“Domain Business Terms” as in Fig. 7). Like with other formal models , these could be “Model terms” - for generic “alphabet” model elements- together with “Business Terms”. 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.

Note that applications building functions on top of a DIM must find a way to technically represent the classes, attributes and relation of the model – as high-level models like those of UN/CEFACT ebXML "Core Components" do not describe a technical implementation.

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 recognised user group or at least involve users and domain experts.

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) can be put.

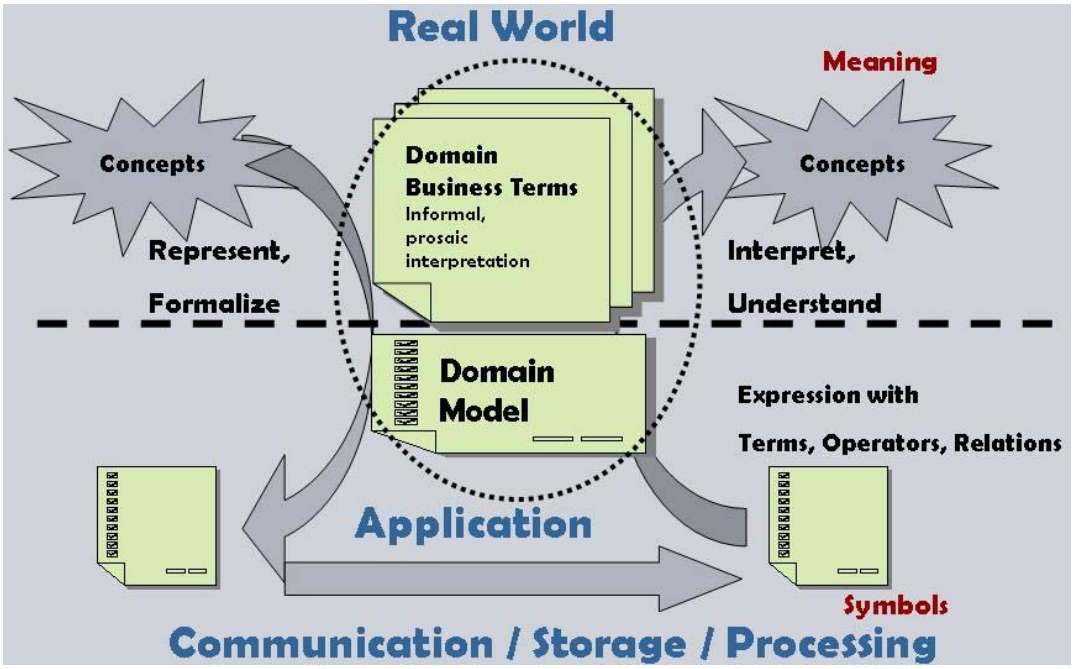

Fig. 7: Semantic interoperability based on a domain model with informal explanations

Complexity in health care requires implementers to publish and standardise *explicit*, written informal explanations, which can easily be written as "Implementation Guidelines" which constrain available standardised formal domain models.

Having a domain model also makes the interface-specific "Business Terms" more systematic, in that an overarching "General Domain Business Terms" (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 Business Terms") – from the new application's perspective.

C.1.5 The context effect

Some models and terminologies claim to be comprehensive for a domain. Any additional information that may qualify the meaning of a term in a model or terminology, however has to be taken into account. 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": any related information can qualify a term and therefore alter and even negate 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.

Other examples include “site” (of patient), “suspicion” (diagnosis), purpose such as “intention”, “order”, “promise”, “completion”, “forwarding”, “refusal” (procedures) and status “received”, “checked”, “scheduled”, “failed”, “cancelled”, “completed”.

Furthermore, the “caused_by” (or, “symptom_of”) relation generally creates a whole network of terms for findings and diagnoses, qualifying a little bit each of the terms in that net. Here is one example based on SNOMED: A patient was brought as an emergency case "P00300" into hospital. He complained about fever "F03003", shivering/ague "F03260" and diarrhea "F62400". Doctors first found an acute inflammation "M41000" of the stomach "T63000" and the duodenum "T64300", later the cause Salmonella cholerae-suis "E16010" was found so that the final diagnosis “gastroenteritis paratyphosa” "D01550" could be documented. In that specific case, all of these terms are related to each other, implicitly giving a more precise meaning to each of them.

Note that many such “context effects” exist and the meaning of practically every formalised term can be altered by some related context. No model or terminology can foresee all possible influential context information.

- There is always an additional influential real world “context” factor which is not yet in the domain model (terminology) – but which will affect the meaning and interpretation of a term in the model (terminology).

As a result of SI4, a model can always be extended by some influential context factor, but then even more context factors will turn out to be relevant, such that there will never be a “final” model. Also from this perspective, the restriction of a domain model to one or some use-case(s) seems to be important.

The dependence of models and terminologies on some specific application context ("context effect") makes it unlikely that a realistic “pan-european common ehealth information model” can be created. Though several large european research projects suggest to have a kind of permanent, mandatory, common eHealth information model, our Report suggests to focus on one or some specific use-cases.

C.2 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 (payors, authorities, different kinds of providers)
- the context of different societies and languages
- different legislations

Due to the “context effect” the “global model” approach is unlikely to be 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 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 endeavour, and would need organisations and policies beyond a specific region or organisation 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, and because it affects each enterprises’ business and IT environment as well.

As the perspective and intention of using application data is essential to selecting and constraining domain models, it is clear that use-cases have to be described prior to establishing domain models and terminologies. 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 formalised information model (regardless whether it describes just one interface or a whole application domain) shall go along with a related natural language reference manual - explaining the formal terms of that model. Note that the informal model of course shall be precise - as informal does not mean sloppy. Semantic interoperability will then be possible among the consensus community acting (or, implementing) according to that reference document.

From an application developer’s perspective, both formalised model as well as its reference manual are like an ICT standard and should of course be made available through public channels (e.g. internet) and without any restrictions.

C.2.1 Scoping

The complexity and the number of medical concepts may be addressed in practical projects by composing specialised interpretations by "scoping" of existing - more general – terms, possibly taken from a standardised, general model. Similar to post-coordination in terminologies, composing instances of models can create a more specialised interpretation. With this way of “scoping”, the "outer" terms make up the context for a more specialised interpretation of the more "inner" terms.

One example is the “Clinical Document Architecture” (CDA) structure with CDA-Header and CDA-Body: Formal terms in the CDA-Header make up the context for everything that is in the CDA-Body. In the scope of the CDA-Header, formal terms in the CDA-Body have a more special meaning than they would have without any context.

As a principle, if it is clear which terms make up the scope of which other terms, this technique can be used to express more specialised concepts than are explicitly available in a single standardised model.

For example, there could be a multi-level system of “scope-and-content”, like nested elements in XML.

C.2.2 Labelling

Another practical technique involves preserving unstructured (natural language) information while there are no matching formal terms available. The solution is to use formal terms (from available, standardised models or terminologies) in order to markup some embedded "un-parsed un-processed" natural language string. The markup terms will then be a more general description of the quite specific meaning expressed by the natural language string, so the natural language text is still more specific and carries more meaning than the related markup. As an example, XML elements may be used to contain natural language text plus structured markup. CDA supports such kind of embedding natural language.

Note that “labelling plain text” is in contrast to a "scoping context" which modifies the interpretation of embedded information. Instead, the formal markup captures all semantics that can be extracted and expressed via a given model / terminology, but mostly does not express everything that is said through the plain text.

Scoping can be an additional technique to create useful (i.e. very specialised) markup.

C.2.3 Relation to the core processes

It must have become clear by now, where the main challenge in achieving semantic interoperability lies: defining common terms that describe meaning in a systematic way. It must also be clear that this is a process that has to start from the real world (requirements), enter an (abstract) space of modeling information and then return to the real world (implementation). The model space is necessary and critical to achieve interoperability between different implementations (that reality makes exist), but it does not do the job on its own. Four steps (processes) are required, backed-up by a fifth (horizontal one): dissemination and knowledge sharing. These four processes analytically explained in Chapter 4 are (re)described here from the semantic interoperability perspective set in this chapter.

C.2.3.1 Business Use-case Definition

Use-cases are stated in terms of records, states, actions and messages - i.e. the use-cases refer to some set of terms with a meaning in an application domain. Therefore, the use-case inception phase serves as the native source of such domain terms: During inception, users (or application experts) explain things of their real world: facts, event, procedures - using some domain "slang". When defining business use-cases, the terms need to be extracted from that domain language and will serve as the anchor points of a prospective information model. The usage of terms in the respective real-world domain shall be explained as a part of the "Reference Manual" which comes along with that new model. Relations between classes of this model and - if possible - some attributes may be extracted at this early stage as well. However, at this time, it is most important to capture the user's view via the model rather than focussing on features of the model.

The focus of eHealth implementations in the daily practical work typically is on clarifying use-cases and defining / updating the natural-language interpretations for models and terminologies.

C.2.3.2 Profile Development and Maintenance

In order to establish the more precise terms used by technical use-cases (business terms), a more detailed information model shall be part of the use-case definition (a recursive procedure). Rather than strictly deciding in a chicken-egg-problem that goes "the model is first" versus "the storyboard is first", both the information model and the storyboard shall be refined in a stepwise fashion. So, one task in refining the use-cases is to create/extend information models reflecting application terms, relations and attributes.

- One task along with this is to find out those "new" terms which are related to or equal to the established terms in existing models.
- More often however, careful scrutiny reveals the opposite: the same term used by different use-cases typically has slightly different interpretations. Only additional informative text in the manual can clarify this.

- In order to reuse existing work, appropriate base Standards may be referenced for classes in the technical information model. Controlled vocabularies are an example.
- Some model terms have to be generic in that they will be refined or constrained at the level of Interoperability Specifications. Frequent examples are wildcard classes, to be filled with controlled vocabularies. The profile's model class then just expresses the intent and purpose, but no instances. An Interoperability Specification will later refine the model in defining to use an appropriate vocabulary for that given class.

C.2.3.3 Base Standards

In many cases during defining a Profile, existing standards can help to express some terms of the related specific, technical information model. Therefore, classes or aggregations of classes in prospective information models may be taken from or derived from existing standards. Standardised vocabularies are a typical example. Messaging models are another example. Note that because the meaning of terms in a Profile context often is more specific than without context (as in a standard's text describing a generic model), such terms need to be redefined (specialised) in the informal manual related to that profile.

On the other hand, common parts of a model that are likely to be reused by other Profiles, may be shared as a standard.

C.2.3.4 Profile QA

The semantic aspects of interoperability can be validated via test cases. For automated testing a facility for expressing test data instances ("expressions") and comparing them at the model-level with computed output is required.