Towards Use Cases in Sparse Architectural Data Exchange

A comparative study triggered by recent developments in healthcare interoperability

Gabriel Wurzer[1,2]

[1] *Vienna University of Technology, Austria*

[1] *http://www.iemar.tuwien.ac.at*

[1] *wurzer@iemar.tuwien.ac.at*

[2] *Systema Human Information Systems, Austria*

[2] *http://www.systema.info*

[2] *gabriel.wurzer@systema.info*

Abstract. Data exchange continues to be among the most challenging problems during planning projects. Recent tendencies strive to support a diversity of involved domains, by introducing semantically rich data models which promise a seamless transition of the data from one field to the other and back again. Often, this interchange is facilitated by using only a subset of an otherwise complex data structure (e.g. specified by IFC). A specification concerning "which data to include and which to omit", however, remains non-standardized, subject to internal agreements among modeling package vendors and may not be available for outside parties. We find that this is a pity, since exposing also the orchestration part would help improve the quality and reduce the time spent with the implementation of data exchange. In our work, we want to report on recent developments in the healthcare domain, which faces similar interoperability problems, and compare these to architectural data exchange. Our discussion will focus on the "Integrating the Healthcare Enterprise" (IHE) framework, which provides a way to structure data interchange using Use Cases. Our goal is to work towards the introduction of similar techniques in the architectural field.

Keywords*. Data Exchange; Use Cases; Standardization; Frameworks, Small Is Beautiful.*

1. Introduction

Data in the healthcare sector is segregated in between the involved fields: A Hospital Information System (HIS) stores data about patients and documents, a Radiological Information System (RIS) together with a Picture Archive System (PACS) is occupied with managing radiological imagery, Laboratory Information Systems

(LIS) are used to store data about lab samples and results, and all are orchestrated to give one common system – the “clinical system”. Recent developments, however, aim at a communication of clinical data on a wider level, in order to introduce Electronic Health Records (EHRs) on a national basis. In this context, the medical field is faced with a diversity of interchanging systems, which specify data in various formats. As a consequence of this problematic interchange, the Radiological Society of North America has introduced a framework named IHE (“Integrating the Healthcare Enterprise”), that specifies medical use cases and data interchange semantics with reference to already-existing data formats (RSNA 1998). The framework does not define new standards, it rather uses existing ones and prescribes when and in what order data messages are to be exchanged, as well as the contents of these messages.

Architecture has also seen a variety of standardized data formats, some of which are open and some of which are vendor-specific and closed. A common problem in all of these is the methodology of data exchange, which is accomplished by using potentially densely-populated files, which store all information needed. In the reality of architectural software, however, only subsets of these have been filled, with the effects of limited compatibility even when using standard formats. We wish to introduce several new concepts into the architectural domain, which we feel should be discussed by the community:

- The use of “challenge and response” instead of “one file says it all” (see page 278).
- An explicit definition of Use Cases arising from the needs of planners, rather than a data file to be applicable for every possible use (see page 278).
- An application landscape that consists of several communicating software components which may be exchanged or replaced during a project, rather than families of vendor-specific, monolithic software packages (see page 280).

Our thoughts are based on the movement towards IHE in the healthcare industry, which we will present in the first part of the paper (see "Data Exchange using IHE", page 273). The second part ("Implications for Architecture", page 277) is occupied with mapping the presented concepts to architecture. To conclude, we sum up our presented thoughts and give a discussion over possibilities and pitfalls that need to be avoided ("Conclusions", page 281).

2. Related Work

The need for a collaborative data exchange lies in the nature of architectural planning projects, in which actors across multiple disciplines hold data in a variety of representations. The key question of how to bring information from A to B has a long history in the Architecture-Engineering-Construction (AEC) Sector: Early product model formats such as IGES concentrated on the syntax of the exchanged entities, but failed to supply additional semantics. This restriction was lifted to a certain extent with STEP, which also considered functional aspects of a product model. Recent exchange formats such as IFC further introduce an object-oriented representation of entities, which is a basis in all Building Information Models (BIMs). The taxonomies found in these approaches are coherent but fixed, based on strict hierarchization and inheritance as known from object-oriented programming (OOP). Newer approaches (Ugwu et al 2005) work on lifting this predetermined structure by additionally representing and transmitting design knowledge (i.e. semantics of entities, inheritance relationships, etc.). Further work (Carrara et al. 2009) diversifies the interchanged knowledge into General Knowledge and Specialist Knowledge as ontologies tailored to the involved discipline. The additional representation of process knowledge was presented e.g. by Gero and Kannenglesser (2006). A step-wise data interchange triggered by Use Cases has, however, not been previously looked into - in all cases, the physical exchange is done via files. Other than in the Information and Computer Technology (ICT), especially in the field of web-based communication, Service Oriented Architectures (SOAs) have so far not found widespread use in the AET sector. What exists are variety a BIM Servers (Beetz 2009), which allow for the upload and shared exchange of building data. A truly component-based approach such as the one presented herein, is to date still missing.

3. Data Exchange using IHE

IHE is a framework which describes Use Cases (e.g. admit patient) in several Domains (e.g. IT Infrastructure, Cardiology, Radiology). These Use Cases are formalized as Integration *Profiles*, which describe the necessary data transfer that happens in the background as set of *Transactions* between *Actors* (usually software components). A transaction specifies ways the actual transfer of the data with reference to standardized file formats, their content, and the transfer methods used. IHE does not invent new standards, it uses already existing ones. In fact, standardization organizations which are occupied with specifying file formats are involved in and work in close cooperation with the IHE.

Figure 1 brings an example of a typical Integration Profile, which will be now described in full detail. The Use Case under consideration is concerned with *Cross-Enterprise Document Sharing* (abbreviated as XDS), in which one clinic submits a document to a central storage (*Repository*). The Repository stores the actual content of the submitted data as set of files. Metadata (e.g. Title, Date) are furthermore forwarded and stored in a *Registry*, which is an index over all data that is used in querying the patient's data. For patient data, there is a separate index, which is again used for queries and search. This *Patient Index* has the goal to merge the different patient data replicas present in each hospital into one centrally known data set (each system has its own notion of patient data). It furthermore references a set of documents that are submitted for the patient, known as the patient's *Electronic Health Record* (EHR).

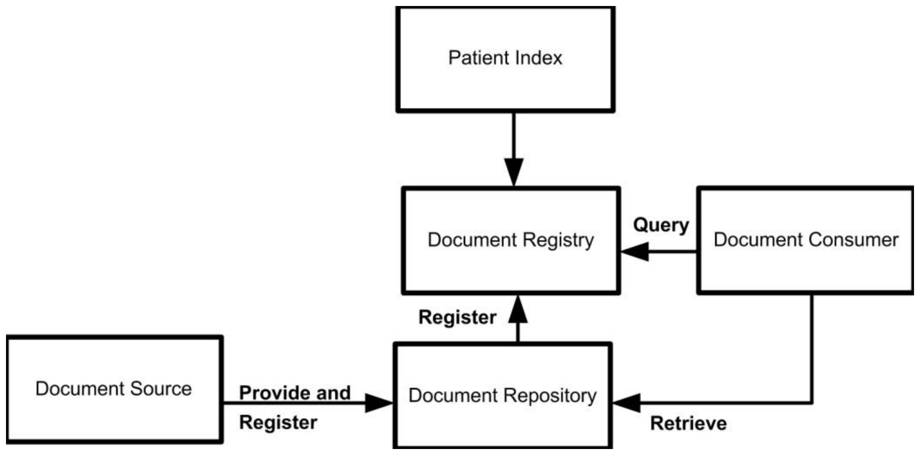

Figure 1
IHE Profile (Cross-Enterprise Document Sharing, abbreviated for better readability).

When retrieving a document, a requestor (*Consumer*) has to hand the patient's identification to the Registry, which will then return all metadata for the documents found, including links to the content in the repository. This metadata is usually presented in a tabular form - as a list of documents for that patient. When clicking upon a document entry, the consumer accesses the Repository using the link contained in the metadata (*Retrieve*). After having received the document, it is in the duty of the consumer to be able to depict a document, which can be in any document format (e.g. Word, PDF, XML). There is, however, a strong move towards the use of structured XML data (called Clinical Document Architecture, CDA), in order to supply consumers with document that consists of paragraphs with added semantics (e.g. medication paragraph according to a previously agreed-upon specification). This semantic interoperability with reference to paragraphs of data is still a future topic, since the field is currently in the transition from closed-format documents to XML. There have been a wide range of research approaches that try to map from one kind of paragraph (e.g. medication as defined in country 1) to another (medication as defined in country 2), a usual approach used to carry this out is to use ontologies. The relation to becomes important when looking at our related work in the architectural field, which is also very much focused on this aspect.

All Profiles have been elaborated in a *community process*, in which industry partners and healthcare professionals team up and agree on actors, transactions and select underlying data formats and transfer methods. Going into more detail, this process consists of four different phases of elaboration:

- In the first step, clinical and technical experts define critical Use Cases for information sharing.
- Technical experts create detailed specifications for communication among systems to address these Use Cases, selecting and optimizing established standards. The result is one or more Profiles, which are handed out for trial

implementation.

- Industry partners implement the profiles according to the given specifications. In the course of implementation, changes are gathered and merged into the specification.
- As last step, IHE tests vendors' systems at carefully planned and supervised events called Connectathons.

A Connectathon is a huge developers gathering, in which all implementers must test whether their software can communicate with all other vendors implementing the same Profile. If this is the case, the IHE certifies the correct interoperability by handing out an Integration Statement ("The vendor has successfully proved his interoperability in the Profile under consideration"). This integration statement then forms a major competitive advantage over other proprietary software packages, making it "IHE conformant".

4. Implications for Architecture

While the relations from healthcare data interchange to architecture seem to be unobvious at first, a deeper look unveils points which are currently lacking or still need to be addressed in the field:

- Architects are using a variety of software packages by different vendors to do their planning work. These packages provide for data interchange in the form of file interchange. Both proprietary as well as open formats are used; however, one characteristic in the exchange is that the actual contained data differs in semantics, and is generally sparsely populated (i.e. information is not present in the file that is available to the modelling package).
- The way in which data formats have evolved was triggered by software vendors, which aim at the storage of their domain-specific information. Data transfer to other applications happens by mapping information from one domain model to the other, however, this process is costly because of the need to support a variety of versions of the data format and domain model on both sides (*N:M* mappings need to be supported).
- The usage of the data over the course of the design process has been implicit; the lack of clearly-stated definitions in the sense of expectations of what the end-user wants to achieve with the data has burdened data formats with a variety of data sections, most of which remain unutilized during concrete data exchange.

Healthcare has seen these developments in earlier phases, and has responded with a rationalization at the interface level of software packages. Two main arguments have led to the introduction of the IHE framework: The complications of maintaining the data exchange with all other vendors at the product supplier's side, and the protection of investments at the product buyer's side. Clearly, those points are also valid for architectural data exchange, in which software needs to be highly flexible and adaptable even in the short time-span of a project. The heart of our critique falls into three parts, which will now be presented together with proposals on how a paradigm shift could be reached.

6.1. "Challenge and Response" instead of "One File Says It All"

In single-file data exchange, information has to enter a data file in a highly structured way. With the responsibilities of exchange being put on a single medium ("*the file*"), high amounts of information that is irrelevant for the individual work step needs to be entered. If data exchange were instead built up on a "challenge and response" basis, in which the caller informs the information source about the type, purpose and context of the data exchange taking place, the succeeding negotiation among both parties could focus on the actual contents that need to be delivered. Furthermore, this concept is not limited to separate exchanging programs, but can be conducted also within the components of program itself (also see "Multiple Communicating Components", page 280).

Figure 2 gives an example of such a data exchange. An egress analysis and pedestrian flow software acts as *Geometry Consumer* and requests all "Area" and "Access" geometry in metric units (relative to a local origin). The query is performed against a CAD Application, which acts as *Geometry Provider*. The response of the latter program takes the form of a file (e.g. in format DXF, to be specified in the according use case!) which contains the requested data.

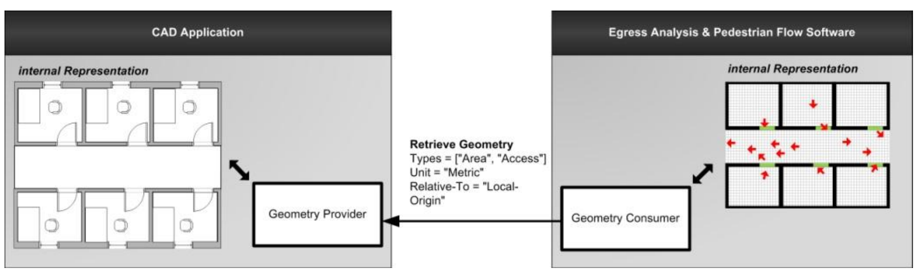

Figure 2
Example of "Challenge and Response" used in the exchange of architectural data

An implication of this approach is that both applications have to be online when the data exchange occurs, so that Geometry Provider and Geometry Consumer can negotiate. Since this is not always acceptable - i.e. one party might not own both programs, the programs are not started in the same instance, etc., there are two ways in which to counter this:

- The exchange might be carried using a server, which is always online and exposes the CAD drawing to requesting applications (BIM server implementing an actor).
- The data exchange might be asynchronous, in the sense that the request is received e.g. by mail and also returned using this medium.

6.2. Use Cases in Architectural Data Exchange

Use Cases are subject to formulation by professionals. An example is given in the "Flow Computation Workflow" Profile in Figure 3, in which a pedestrian flow is to be calculated in the context of fire safety checking. The process starts with a CAD application wishing to perform a flow calculation and thus calling up an actor called *Flow Computation Provider*, giving the utilizations of each area in the office building under consideration as input in the form of a Comma-Separated Value (CSV) spreadsheet file with the following semantics:

- Row 1 (Header): "Area name";"Occupation"; "Time"
- further lines (Data): <name of area>; number of occupants; time of day in seconds since midnight

The *Flow Computation Provider* Actor is inside the already mentioned egress analysis and pedestrian flow software and communicates with the aforementioned *Geometry Consumer*, which is present in the same program. The data flow continues to retrieve the area and access geometry, then performs the pedestrian flow computation and returns it into the CAD application, where it is being visualized.

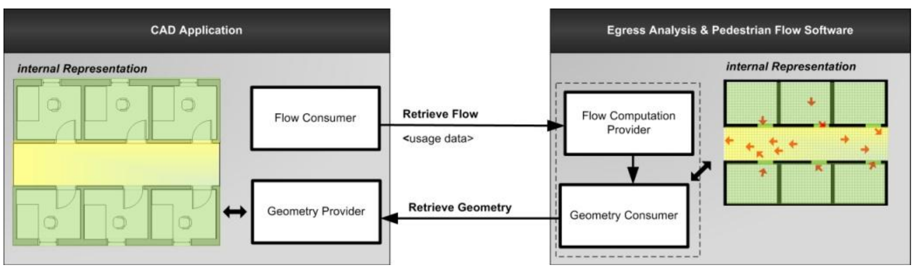

Figure 3

Example Use Case for computing the pedestrian flow of an office level through multiple actors performing a mutual exchange of data

As can be seen, the Use Case describes not only the typical context ("fire safety"), but also the underlying data exchange and semantics used. The definition of the DXF file was left out intentionally, since this refers to the presented problem of *sparse data*, and thus receives more attention in the following elaboration.

A DXF file is an openly documented data exchange format for geometry, which is contained as set of drawing entities (e.g. Polyline, Arc, etc.). In the context of the presented "Area" query, only a subset of the numerous capabilities of the DXF format, as is given in (AutoCAD 2011), would be used:

- The HEADER section of a DXF file stores semantic content, for example whether or not angles are expressed in a clockwise manner. However, semantics are defined in the used profile rather than in the file itself; therefore, the contents of this section are neither of importance, nor do they need to be interpreted on the receiver's side. The same is true for all sections that define additional preferences (e.g. viewport settings, user co-ordinate system, print settings).
- The ENTITIES section stores the actual geometry. In the case of the "Area" query, it is required that a list of Polylines, each being closed and having an arbitrary number of intermediate, three-dimensional points specified in

meters within a world coordinate system, are present. Areas are defined to be all walkable spaces of a building. In addition to the actual geometry, each Polyline has to have a unique name. Furthermore, the supplier has to specify the type of the area (e.g. room, hallway) according to a table (see Table 1). Should an area have more than one type, commas are used as separators.

The given specification is sparse: Only relevant pieces of the interchange format need to be filled and interpreted by the receiver. The semantics of the physical data exchange are defined in a minimal, non-ambiguous manner that is tailored to the Use Case under consideration. As a matter of fact, it is easy for software vendors to implement functionality in an iterative manner, with reference to the workflow in which it will be used.

Area Type	**Usage**
UNKNOWN	area type not known
ROOM	office spaces, living spaces
HALLWAY	circulative spaces

Table 1
Standardized table used for the specification of area type in the presented "Area" query.

6.3. Multiple Communicating Components

Having a software architecture made out of a set of cooperating components has a number of advantages when it comes to data interchange:

- In contrast to the now-common implementation of data exchange as a set of additional Plug-Ins (one for each program to import or export data from), an implementation can use one component set per problem domain. For instance, a CAD application could use the previously presented pedestrian data interchange to communicate with all partner programs implementing the required set of actors. As a result, development effort is invested for additional functionality, not for the support of additional file formats.
- In contrast to the current situation, the proper functioning of data interchange can be certified in a Connectathon to be held among the industry partners.
- With each component being a self-contained piece of software, there is no need to solely include it in the (off-line) installation. An Actor could also be made into a server residing on the network, thereby allowing partners to collaboratively exchange data even over company borders. A shift of functionality to the Internet Cloud could enable new business models, in which the cooperating partners pay for the usage of a Use Case *as service*, rather than having to invest into a whole application package.
- A standardized data exchange made out of communicating components also has the benefit that programs can be replaced during a project, without losing the ability to transfer data. For an architectural office, this translates into a protection of investments and allows the freedom to choose among

all providers of a specified Use Case.

5. Conclusions

The presented paper has argued for the introduction of sparse architectural data exchange based on Use Cases, in contrast to the now-common move towards all-embracing file formats such as IFC. The key problem we have addressed is the fact that large parts of a data file are negligible in the context of a specific work step in which data is to be exchanged. Therefore, it would seem beneficial to focus only on the specific information required, and transfer these using whatever data format fits the purpose best. Such an approach has already been implemented in the healthcare sector, in the form of an open framework called "Integrating the Healthcare Enterprise" (IHE). The novelty of our work therefore lies in the transfer of the concepts found in IHE into architecture, giving a discussion as to how this could help data interchange as we go along. We have identified three strong arguments that support such a type of data exchange: First, architectural software packages could use a sequence of "Challenge and Response" transfers in order to reduce the effort needed to perform an exchange. Second, such a "Challenge and Response"-could also take the context of the data exchange into account, thereby arriving at a set of Use Cases where data exchange needs to be conducted. Such a workflow-based approach can help to focus on a problem domain rather on the general case, further reducing the effort of implementation. Third, the defined Use Cases can be implemented as set of communicating components which are certified in an integration test among all implementing industry partners. This helps to ensure that exchanging parties can interoperate as expected.

The introduction of the presented data exchange requires a rigorous elaboration of Use Cases in working groups (i.e. a community process). In IHE, these "domain committees" consist of field specialists who are responsible for structuring, prioritizing and reviewing the work packages that lead to the formulation of Use Cases. These then undergo a draft phase (public commenting), trial implementation phase (technical refining) and are then published as finished Profiles. A similar approach would have to be done in architecture: Work groups lead by architects and industry partners should formulate Use Cases, which are then published for trial implementation among architectural software vendors. Once a Profile is finished, the exchange functionality expressed therein has to be tested from the next Connectathon on.

There is one pitfall connected to the certification process that is to be avoided when taking these concepts to architecture: Certification tests all software packages that are delivered at the time of the Connectathon. This, however, does not mean that the same software can communicate with certified software from a previous Connectathon, since Profiles can have changed (although this is not often the case). What is missing to date is an explicit versioning, in which components agree on using the same version of a Profile before starting to communicate. This approach is similar to secure communication, in which the communicating partners determine which cipher algorithm they have in common.

Acknowledgements

Our paper is a direct follow-up that addresses earlier work done by the group of Gianfranco Carrara at the Department of Architecture and Urban Planning of Sapienza University Rome. The comparisons to healthcare data interchange using IHE are furthermore based on actual work in this sector, in which the input of co-workers (Harald Bartl, Reinhard Egelkraut, Peter Divjak, Karl Holzer and Alexander Gottermeier) has been of major influence.

References

Beetz, J 2009 *Facilitating distributed collaboration in the AEC/FM sector using Semantic Web Technologies*, PhD Thesis, TU Eindhoven.

Carrara G, Fioravanti A and Nanni U 2009, 'Knowledge-based Collaborative Architectural Design', International Journal of Design Sciences and Technology, **16**(1): 1-16.

Chen PH, Cui L, Wan C, Yang Q, Ting SK, Tiong RLK 2004 'Implementation of IFC-based web server for collaborative building design between architects and structural engineers', Automation in Construction, **14**(1), 115-128.

Gero JS, Kannenglesser U 2006 'A function-behaviour-structure ontology of processes', in Gero JS (ed.), *in Design Computing and Cognition '06*, Springer, The Netherlands, pp. 407-422.

Kvan T 2000 'Collaborative design: what is it?', Martens, B (guest ed.), Special Issue eCAADe '97, *Automation in Construction*, **9**(4): 409-415.

Ugwu OO, Anuba CJ and Thorpe A 2005 'Ontological foundation for agent support in constructability assessment of steel structure – a case study', *Automation in Construction*, **14**(1), 99-114.

RSNA 1998 'Integrating the Healthcare Enterprise', [online]. Available from: <www.ihe.net>, [Accessed on 31th January 2011].