

Where The Context Lies

By
Timothy W. Cook

This work is copyright Timothy W. Cook, 2008.

Use and distribution of this document is covered under the requirements of the
Creative Commons Share-Alike License.

Please see: <http://creativecommons.org/licenses/by-sa/3.0/> for details.

Release 2.0
(The first release was on 23 September, 2008)



Where The Context Lies

Timothy W. Cook

<http://www.linkedin.com/in/timothywaynecook>

16 March, 2009

Introduction to Release 2.0

There isn't really a lot of changes to the text for this release but the feedback I have been getting on the first version makes me realize that the major point I was trying to get across just didn't get there. So I'll be blunt here. Stop thinking 'data' model' and start thinking 'information model'. The persistence layer is completely implementation specific and has nothing to do with the actual information (as opposed to data). The key concept is being able to exchange "information" with semantic integrity.

Acknowledgments:

This work is made possible by the generous contributions of knowledge and experience of the openEHR Foundation and openEHR community at large. See <http://www.openehr.org> for more information.

Scope:

This point paper was developed from personal practical experience as well as the study of vast amounts of peer-reviewed and anecdotal literature regarding success and failures in healthcare information system implementation and integration projects. This paper does not reference specific literature as would be required of an academic study. Instead I have chosen to focus on the foundational problem of the semantic interoperability issue and challenge the reader to think differently about information system design.

"Insanity: doing the same thing over and over again and expecting different results."

--Albert Einstein

Let us stop the insanity now!

Where The Context Lies

Timothy W. Cook

<http://www.linkedin.com/in/timothywaynecook>

16 March, 2009

Problem:

Healthcare information is full of who, what, when, where and why **context**. The current design approach to healthcare information systems (HIS) doesn't provide a facility to transfer that context when the data is exchanged with other systems.

Background:

It has long been recognized that there is an inherent need for full semantic interoperability between information systems that are used to manage healthcare information. But what is a *healthcare information system* (HIS)?

In the broadest sense; a HIS is any information system that maintains and manages information affecting the health of a person or population. It is widely accepted that not only the medical condition of a person is significant to their health but a host of social, economic and environmental conditions place direct and indirect effects on personal and population health.

As we enter the discussion on the foundational concept of semantic interoperability we will focus on the clinical systems in the HIS world. But I ask the reader to keep the broader picture of these other information systems in their mind.

Discussion:

While the science of information design encompasses much more than computerized information systems. We will be discussing the computerization of healthcare (specifically clinical) information in this paper.

Expressed in its simplest terms; the current approach to developing an information system follows these steps:

- requirements gathering – using one or more of the many methodologies
- systems analysis – determine from the requirements the systems involved to meet the needs of the requirements
- data modeling – build a model of the data elements available from the information sources (systems) to meet the needs of the requirements
- implement the data model in some persistent storage facility
- implement the software required to add, edit and manage the storage and use of the data elements defined in the model
- maintenance of the system then entails modifying the data model and the software as the requirements change

Where The Context Lies

Timothy W. Cook

<http://www.linkedin.com/in/timothywaynecook>

16 March, 2009

This process has proven robust and reliable over many years and millions of software applications being deployed when sound software engineering principles are adhered to in the entire process.

Looking into how typical applications handle the information management aspect a bit more; we see how an application like this actually manages information. We see that the data elements are discretely persisted and the software and/or data management layer manages *the relationships* between these data elements.

Remembering that data elements are simply data elements. It is their relationship to each other and their relationship to the user, via input and output design in each application, that gives them their semantic context in order to create information from their existence. As an example of the data versus information conundrum; let us say that we have a data element of the integer 102. That integer can represent many things. Is it a systolic blood pressure measurement, a heart rate, a body temperature? It all depends upon the context that includes the units of measurement. The validity and clinical significance also depends upon patient position, point of measurement, device used, who the patient is, etc.

To be sure, there are certain elements of a database management system that lend to this context. But do not forget that this is a part of the software layer not the actual persistence of the data. While it is often conceived that table names and column names in SQL databases provide this functionality. I ask the reader to consider what happens when this data needs to be exchanged with an application that uses a hierarchical or object or XML database? There is no standard way to do this translation. In fact, even between applications based on SQL databases, the migration of data must be performed on a case-by-case basis where the table and column names are not the same and a custom translation must be built for every case.

Certainly the development of standardized HL7 messages was a big leap forward in information exchange in healthcare. Two big drawbacks to this approach are that the message formats had to be made very generic in order to accommodate systems that may or may not have all of the data elements in a specific message and this exchange process requires a manual mapping on both the sending and receiving ends for each message to their local database. An entire industry developed around providing these mapping engines and the expertise to setup the complex mapping routines. Even if the development of the message formats were defined by domain experts, the mapping is typically performed by software experts, not clinicians.

Where The Context Lies

Timothy W. Cook

<http://www.linkedin.com/in/timothywaynecook>

16 March, 2009

Healthcare information can be and should be used by many people. From the stream of healthcare providers involved in the care of one patient all the way up to epidemiologists and healthcare provisioning experts such as policy makers and economists. Scanning the literature the reader will find many many calls for various standardized data sets to be built into applications. These data set calls are almost entirely made based on the interests of the people making this call. Common sense and indeed experience shows that there is no way for every healthcare application to incorporate all the “minimum data sets” being requested by these competing interests. Even if there existed this magical cover all minimum data set, remember that the real semantic context of collected data that results in information is in the software of the original application not in the data itself.

Completely controlled environments like the U.S. Veterans Administration and Kaiser-Permanente have demonstrated that a centrally controlled approach to healthcare information systems development and management using current approaches does work and does reduce adverse events as well as improve access to patient information at the point of care. The evidence exists that interoperable healthcare information systems are a benefit to those organizations as well as to their patient population. However, even in countries with nationalized healthcare services it is improbable if not impossible to mandate a common data set for all healthcare information systems that meets the needs of all healthcare information users.

What is needed is a new way of thinking about complex information systems. I'd like to begin with an analogy. Thinking about the very popular children's (and adult's) toys, Lego® blocks. A box of these blocks contains various sizes and shapes of components designed to work together to form objects. The kit comes with instructions so that a certain sub-group of the blocks can be used to build a truck. Some of those same blocks can be used maybe in combination with additional ones, to build a helicopter according to a different set of instructions. There are other instructions to build various other objects all based on the same set of common blocks all according to the restrictions described in the instructions.

What if we had a common set of healthcare blocks that were implemented in software such that given instructions about any clinical concept they could represent that concept as an information instance? We can call this set of healthcare blocks a common reference model (CRM). We can also call the instructions for each concept a Clinical Knowledge Model (CKM).

Where The Context Lies

Timothy W. Cook

<http://www.linkedin.com/in/timothywaynecook>

16 March, 2009

If the CRM is abstract enough then it could be implemented in any object-oriented language on any hardware platform. It therefore provides the freedom of use and creativity to meet the needs and desires of application developers. The CKM units (instructions) can be described by the domain experts for each concept because they exist as constraint descriptions of the very broad based CRM. They do not exist in software themselves. They simply tell the software what they are representing. In fact they describe themselves completely with all of their semantic context. Since they exist as data instances and not lines of software code; they can be exchanged with other applications based on the CRM without any loss of semantics. They can be persisted in any type of data management system. They can be queried with a standardized query language so that even the queries only have to be built one time because of the standardized data structures. Best of all, when the ever changing science of healthcare comes to call. A new version of a specific CKM can be adapted from the existing CKM and there is no need to change the software nor is there any loss to the semantic integrity of the existing data.

By now I hope that the reader is asking; when and how do we get started on building this comprehensive, future-proof model for healthcare applications? Just think of the implications. A blood pressure instance collected in a simple disease registry can be transferred to a patient's electronic health record and/or to a public health policy research application without translation and without loss of semantics. Of course real world applications would include larger extracts than this and may include components that describe the geographic area of the patient and the location and setting of the measurement as well as who performed the measurement. All of the context of this information exists in data instances and not in software defined relationships.

But to answer the question above. The model is already designed. It is not only designed but it has been proven in multiple programming languages using multiple persistence layers on multiple hardware platforms. Of course the reference model and knowledge model are not sufficient to build complete applications. There are also definitions for service and support layers.

Healthcare information has many complex temporal and spatial components and then mix in the fact that medical and healthcare knowledge changes rapidly. This is the crux of why healthcare information systems are so complex and typically 'purpose-built' without any real interoperability built into them.

Where The Context Lies

Timothy W. Cook

<http://www.linkedin.com/in/timothywaynecook>

16 March, 2009

There are health information experts that have looked at these specifications and walked away because they believe that they are too complex. The fact is that the specifications may be a little difficult to understand at first because the approach is different. The reality is that health information management is complex. Better said by a man that knew a little about complexity though:

"Everything should be made as simple as possible, but not simpler."

--Albert Einstein

Conclusion:

This new approach requires the reader to adopt a new way of thinking about information system design. It does not require letting go of sound software engineering principles. In fact this approach enforces the sound principles of separation of software and data.

In the words of the great satirist Samuel Clemens;

"Don't let schooling interfere with your education."

--Mark Twain

It is this author's opinion that continuing with the insanity of doing the same thing and expecting different results as if by magic, borders on being criminal. This repetition is taking valuable resources from the global healthcare funds pool that could be better spent delivering much needed services everywhere.

Of course high quality, rapidly delivered information is needed by decision makers in order to provision those services. But we have proven that our current approach leads to wasted time and effort and even poor quality results due to a lack of interoperability across applications causing misinterpretation of existing data. We must begin the change now.

The **context currently lies in the software** where we cannot exchange it. We need to put it into the data where it belongs.

Recommendation:

You should begin with an educational process in order to understand this approach. It is based on almost two decades of research, development and real implementations of the principles. It will take time and effort to understand and begin your own transitional process. It will take careful planning so that you do not abandoned your existing information systems and the data that they contain.

Where The Context Lies

Timothy W. Cook

<http://www.linkedin.com/in/timothywaynecook>

16 March, 2009

I suggest that you contact myself or one of the other experts in this area to begin your planning process. More information about these **openly available specifications**, the community of people and the general principles of the concepts are available at <http://www.openehr.org>

Timothy W. Cook

<http://timothywayne.cook.googlepages.com/>

Email: timothywayne.cook@gmail.com