RETHINKING IHE PROFILES IN THE HEALTHCARE SERVICES WORLD

Stefano Lotti

HL7 Italia Chair

Enterprise Architect at Invitalia – Government Agency for Inward Investment Promotion and Enterprise Development
• In this presentation we see SOA like a “scientific revolution” in the sense of Thomas Kuhn epistemological theory as presented in the book “Structure of scientific revolution” and some other essays

• Be sure, this is not a philosophic/academic presentation. The objective is the effective and practical working interoperability in large scale systems and this requires some review of our mindset to change our design routines.

• A Kuhn’s scientific revolution emerges when a traditional paradigm meets increasing problems and this results into a convoluted theory and behavior

• With this given situation a new theory can emerge. It can be a completely new theory or “only” a marginal practice that can move to the center (F. Flores)

• In any case, it’s relevant to understand that a Kuhn’s revolution is not a palingenesis. It Is primarily a perspective change.
• As an example the Newton’s physics remains valid in the Einstein’s relativity but the sense and the place of Newton’s theory is changed.

• Things in a scientific revolution are frequently the same but are viewed from a different perspective and, consequently, we change our understanding and our behavior in the world.

• So, in this context, SOA is at the same time simple and difficult:
  ▪ it’s “simple” because it resolves many practical and convoluted problems of IT and business
  ▪ it’s “difficult” because it requires a change in our IT and business preconceptions.

• The following considerations are based on practical experience in several Regional, National and European projects
Philosophy?

«The world is all that is the case»
Ludwig Wittgenstein
• In our tradition the central concept, is *the application* where business requirements and technical details are intimately intertwined.

• For us, it’s relevant to understand that every piece of software is specified and realized *ad hoc, in front of a specific problem/task*. Consequently applications are structurally vertical and orthogonal to the business processes.

• When, progressively, integration requirements increase (because organizations are systems!) there is, to cut a long story short, simply a big mess.

• The “*accidental architecture*” created by this approach must be integrated but we have used the same approach for integration.
• We have in any case:
  - **Identification of task/integration to be automated**
  - **Requirement definition**
  - **Realization of an application and/or an integration to automate a specific task or a process segment.**

• Consequently the **integration works on a point to point basis (the “*spaghetti integration*”)**. EAI and middleware approach are attempt of a technological solution but the design approach is an invariant.

• Clearly this is not a technology issue, but a design issue and we can identify two relevant aspects of the old venerable approach:
  - **The intertwined between technical details and business requirements**
  - **The design is, in the practical behavior, at application level. The enterprise system, as a whole, is not a part of the game**
• This kind of concern is, obviously, very relevant in Healthcare space for the intrinsic complexity of the business. We have many actors in many different cooperating organizations, information complexity, process complexity (sometime in an ad hoc structure) and plenty of legacy systems at work.

• When we go beyond a small organization, a pilot project, or a small geographical domain we have the necessity to design large scale integrated systems (e.g. a National or Regional EHR or large scale patient’s identification system). The traditional approach cannot manage such complexity for the following reasons:

Too many specific integrations, too much rigidity in applicative integrations, too much mix of business and technical details.
• SOA in this context emerges as a “revolutionary” approach (in the Khun’s meaning) that changes the structure of the design viewpoint.

• The main characteristics of a SOA can be individuated in:

  - The design is centered on the business system as a whole not on the applications components.
  - In a SOA we consider primarily interfaces design among components.
  - Interfaces and components are at different logical levels of the system.
• In a SOA the design of interfaces and applications *is detached* and there is a supremacy of a “contract first” approach based on *business needs*. This an essential key of SOA IT rethinking.

• **SOA, in its essence, breaks the wall between informatics and business:** SOA is, in one time, business organization and technology.

  “The system IS the Enterprise” (Zachman)

• The “SOA revolution” does not require us to remake from scratch all the systems in place. We have many business logic and data and they “only” need to be reconsidered from a systemic services oriented viewpoint.
IHE profiles, SOA and rethinking

«Welcome to the real world»
Matrix
• IHE Profiles play an essential role in eHealth field. With its enormous work IHE has “urbanized” the use of the eHealth standards in the real world.
• An IHE Profile is a specification document that constrains selected standards to support the working integration.
• Historically the process starts with a specific use case analysis and defines the detailed use of the standards in the real world.
• So the IHE profiling approach is based on a use case by use case process.
• The IHE commitment to the profiles’ development is important now and in the future. However we should consider the necessity of a method rethinking.
- The diagram represents an IHE profiles architecture fragment for resources (documents) sharing and Identification management in a service aware view.
- In IHE profiles we have 7 services contracts related to identification management (5 used here) and 4 services contracts for document sharing.
- In orange there are “services” that use WS standard and in yellow “services” with HL7v2.x messaging technology (MLLP).
• We found in IHE “services” inventory 3 very different vocabularies: ebXML RegistryServices (RS), HL7 v3, HL7 v2 (in two flavors 2.3.1 and 2.5)

• The documents sharing area uses ebXML RS and HL7 v2.5 datatype but the most used resource type uses HL7 v3 (CDA2) with very different datatype.

• Each profile is, in general, well specified. However questions arise at an architectural level when we use the current specification in a real world complex large scale system. We have:
  - too many options
  - no contracts standardization among “services”
  - different levels of services granularity (and low, especially in messaging)
  - It ‘s not possible to use modern WS technology with HL7 v2.x

• Every “brick” seems good in itself but we do not have a “building” at the end
An interesting aspect is the evidence that frequently profiles are binded to technical scenario and specific application architectural choices.

As an example XDS considers only a registry and repository architecture (differently from a RegistryRepository ebXML architecture characteristic)

XCA (cross gateway spec), on the other hand, is only for gateways scenario even if, from a business point of view the operations are semantically identical with XDS. In fact if a participant is a gateway this is not relevant for the consumer, it’s simply a technical or a permission related issue and should not create complexity and overlap.

In XCA some questions of consistency and semantics arise from the use of ebXML RegistryService (RS) like XML vocabulary. In fact a “gateway” per se make no assumption that behind it there is a registry and repository architecture.

Similar questions arise in another profile, XDR, for point to point scenario that, again, uses ebXML RS and in this case the existence of a Registry/Repository architecture is completely excluded.
• The operations are duplicated and bound to implementation architecture.

• In some profiles, conversely, there is an operation reuse like in XDR and, partially, in XCA but the vocabulary used (ebXML RS) is not consistent with the scenario and introduces complexity: plenty of languages used, message complications and paradox (e.g. a “registry error” element without a registry).

• The use of ebXML RS with HL7v2.5 Datatype also introduces vocabulary inconsistency with the Identification area, as we shall see later.
• The Identity area is more complex and has a mix of HL7 v2 (in different versions) messaging specs and HL7 v3 web service profiles

• The HL7v3/WS flavor does not have all the “operations” of HL7v2/MLLP (messaging) specs

• It is not possible to transport v2.x messages with webservices in a standardized way

• So, in this context, the operations are duplicated for another type of application architecture binding (messaging and WS) and frequently because they are fine grained in a message oriented approach
• It’s quite evident that this complexity must be reduced. Every profile, on its own, works properly but hasn’t coherence with the other profiles

• This is not an error of IHE Profiles it’s simply a legacy of the past. Some years ago, before the emergence of the necessity of real massive interoperable large scale architectures, these issues were at the periphery of our concerns

• The weakness is the absence of an effective architecture. So the great works of IHE profiles “bricks” must be rationalized for a meaningful use in a real eHealth architecture

• The knowledge incorporated in profiles is useful and can be reused but it is necessary to avoid the proliferation of specs, the inconsistency among specs and the technical architecture binding

- The main issues are:
  - Contracts standardization absence
  - Implementation binding
  - weak flexibility (e.g. HL7 v2 should be supported)
  - operation overlap generated by implementation architecture binding

• Ok folks, it’s the time for Services Orientation
A rationalized architecture

«There are more things in heaven and earth, Horatio,
Than are dreamt of in your philosophy.»
William Shakespeare, Hamlet
• HSSP artifacts seems to have the necessary characteristics:
  ▪ Are based on business scenario (via SFM analysis)
  ▪ Have a clear “contract first” approach and are independent by technical implementation
  ▪ They represent, programmatically, a true “service inventory” with a good level of coherence among services (this reduces language mix and inconsistency)

• It must be stressed that these services specification are independent from the underlying implementation
The diagram represents a Healthcare services based architecture fragment for resource sharing and Identification management.

In this fragment we have only 2 different services at all.
The Services architecture in the previous slide covers the same scenario of the IHE profiles but we have some relevant advantages:

- We have already noted the **reduction of service contracts used**: the contracts are now not related to a specific technical scenario.

- The **standardization of interfaces is good**. No more unnecessary mix of ebXML, HL7v2 and v3, WS and MLLP. The languages used are consistent. This standardization is related also to **services composition**. This is relevant for a true Business Process Management (BPM).

- Consequently **there isn’t an architectural solution binding**: in the previous diagram we can see only a possible architecture and the SOA interfaces can be reused in a different scenario. The services are effectively like “bricks” independent from implementation choices. This is also relevant to make easier the integration of different legacy infrastructures.

- **H7 v3 and V2 (or whatever) are supported** in an organized way by means of the **semantic signifier** mechanism that separates operations and contents in a structured way.

- We are not talking about “application” but **we stay in a services space**. This is the focus and it implies that we can integrate and govern the architecture avoiding unnecessary technology lock-in.
• We want to empathize the **better governance** related with a SOA architecture

• In the architecture that we have seen previously, **as an example**, we can start with a gateway based architecture and subsequently propagate the interfaces and, eventually, remove the gateway between **community** if it’s useful (it’s a business lead decision)

• The data contents of the services can be versioned. So it is possible to govern the evolution of the interfaces. **The services remain stable and the data content can evolve without breaking the architectural structure** (e.g. progressive removal of HL7 v2 content).

• This aspect is essential in large scale eHealth programme. It is not enough to have a piece of software or an integration working. The system as a whole must work and must be capable of an organized evolution.

  e.g. Initially this community can start with a preexisting legacy interface or IHE Profiles. **The same approach is valid for the other community**
Another relevant aspect is the **reuse**:

- **We have a content reuse**: the data content of IHE profiles remains valid (e.g. PCC, XD-LAB etc.)
- **We can have an application reuse**: an existing IHE infrastructure (or whatever similar infrastructure) can be wrapped

The diagram on the side represents some approaches for document sharing (the second is also presented in a IHE SOA white paper)

In an evolutionary logic every component can be replaced (e.g. with a RIMBAA generated Registry/Repository or whatever) without affecting the architecture.

So we’re not necessarily talking about new software but **only** about new and consistent interfaces.
Conclusions

«Incessant the falling Mind labour’d
Organizing itself, till the Vacuum
Became Element, pliant to rise,
Or to fall, or to swim, or to fly
With the ease searching the dire Vacuity»
William Blake, The Book of Loos
Usually we think that the changing is progressive and the question is only how fast it changes. This is not always true and in fact a jump has already taken place.

This is true in contemporary IT and particularly in eHealth. We stay in the middle of a Scientific Revolution (in the terms of Thomas Kuhn).

The paradigm of contemporary IT is changed: The focus is not the task automation but the organization itself and consequently the interfaces’ surfaces of the systems.

We speak about the same thing (we have people and software) but the concept of application is pushed back to the periphery of our concern.

As we have seen (I hope) it is not a question of aesthetics but a simple, practical and concrete question: how our complex healthcare systems can be realized and managed in a meaningful way.

Surely this is a new world compared to traditional IT. This new world requires appropriate methods and implies some new objects and the reconsideration of old objects in the new context.
«Strength is irrelevant. Resistance is futile. We wish to improve ourselves. We will add your biological and technological distinctiveness to our own. Your culture will adapt to service ours.»

(Star Trek, The Next Generation, episode: "The Best of Both Worlds" )