



# Standards Implementation Tools: Maximizing eHealth Human Capacity

Gavin Tong, MBA

Standards and Interoperability Editor, is a Director with Gordon Point Informatics Ltd. in Toronto, Ontario

*"Efficiency is doing better what is already being done."*

Peter F. Drucker

Let's face it, health information standards are complex. Unfortunately, the complexity has a direct impact on the human resource capacity in the eHealth industry. The good news is that tools that decrease the complexity of standards can help address two dimensions of the capacity challenge, namely the quantity and capability of people to develop standards compliant applications.

Imagine the following job posting "Java developer wanted. Must have a degree in computer science, Microsoft .NET expertise, and HL7 V3 RIM certification." The pool of qualified candidates in Canada would be incredibly small, and even smaller for any particular city. Drop the requirement for HL7 V3 expertise and the number of potential candidates would increase exponentially.

In order to drop the requirement for HL7 V3 expertise, we need tools to help us reduce the complexity of the standards in several areas:

1. Standards reference other standards, which creates barriers to finding answers to pressing questions leading to more time spent "chasing the goose than preparing the meal."
2. The same standard needs to be described in different formats (e.g. word documents, data models, XML schemas, code, etc.) to meet various

human and machine requirements. Unfortunately, inconsistencies and contradictions are often introduced into the various formats due to the lack of tools to automate the translation process between the different formats.

3. Healthcare is a complex business and the standards are merely a reflection of healthcare business needs. The use of multi-code system value sets, dynamic value sets, and hierarchical code systems is a developer's nightmare but a necessary evil if our applications are going to meet the broad terminology requirements of our clinician end users.

Tools such as Infoway's HL7 Message Builder aim to reduce the complexity of standards and decrease the HL7 V3 learning curve. The HL7 Message Builder Tool allows developers to map their data to and from an object model that exposes pan-Canadian EHR messaging standards content in a business friendly way. This tool should diminish many of the additional challenges of HL7 V3 implementation such as message transport, parsing, and validation.

A tool that helps build well formed messages is important, but developers also require a testing environment to quickly assess their applications' ability to communicate with other systems in the EHR. Luckily, the HL7 Message Builder Tool comes with connectors to both Mohawk's EHR Reference Implementation (MARC HI) and Intellware's Test Level 7 (TL7) allowing

developers to simulate data exchange with EHR systems and receive meaningful error reports to help debug problems on the fly.

To be fair, there are many other tools required to simplify the challenges of implementing standards, too many to cover in this article. The vast array of tools could itself add a layer of complexity for developers. However, one way to resolve that challenge is to follow the adage of "teach a person to fish and they eat forever." By providing the standards in computable formats commonly used in the IT sector developers can create their own tools or re-use existing tools that could not otherwise be used for standards implementations.

Standards are the key to interoperability, but as BC Holmes, Senior Architect with Intelliware, points out, "The success of Canada's interoperability efforts hinges on our ability to get the tooling right." Ideally, tools such as the ones previously described will lower the learning curve and increase the number of people qualified to implement standards. In addition, these tools are expected to decrease development times and costs, ultimately allowing implementers to focus efforts on innovating user interfaces, application work flow, and business intelligence functions.

The author would like thank everyone who contributed their thoughts on this topic: Monica Harry, BC Holmes, Joginder Madra, Lloyd McKenzie, Andy Stechishin, and Ken Stevens.