# Concept Note: Adopting a Formal Process for Defining and Testing OpenHIE Workflows

2018-09-27

## Background

The OpenHIE specification is described in terms of functional workflows. These workflows are expressed in terms of standards-based interactions between architectural actors. At a top level, OpenHIE’s architectural actors are illustrated in Figure 1.



Figure 1 – OpenHIE’s High-level Architecture

As its standards framework, OpenHIE has adopted **profiles** from the global standards development organization, **IHE** (Integrating the Healthcare Enterprise; [www.ihe.net](http://www.ihe.net)). IHE profiles may be characterized as Implementation Guides. Profiles do not define new standards; rather, they describe at a conformance-testable level how a portfolio of informatics standards are to be employed to operationalize specific use cases. Generally, profiles describe the behaviours of a transaction’s participating actors including initial conditions, information exchange protocols, post-conditions, mandatory exception handling and audit and security requirements.

Usefully, IHE profiles provide a level of encapsulation and re-usability that make it simpler to define OpenHIE workflows. OpenHIE’s workflows, when defined in terms of profiles, inherit the underlying mandatory conformance-testable behaviours of the participating actors. It is also important that these behaviours can be extended to express OpenHIE-specific requirements beyond what is defined in the internationally-balloted profile. Again, this simplifies the level of expression needed; the OpenHIE workflow need only describe (in conformance-testable detail) what are the extensions (the deltas)… not the entire functional capability embedded in the profile.

## Proposal

There is a formal grammar that may be leveraged to describe the behaviours of participating actors in an interoperable information exchange. The field of Behaviour-driven Development (BDD) has evolved significantly over recent years (<https://hiptest.com/blog/2016/04/28/getting-started-with-bdd-part-1/>) and a formal language, **gherkin**, has been broadly adopted to unambiguously express workflows in a way that is readily understandable by business owners and yet is rigorous enough to support both manual and automated conformance testing.

The OpenHIE community, by virtue of being an open source initiative targeted at the development of global public goods, enjoys an unlimited license to HipTest (<https://hiptest.com/>). HipTest is a platform for:

* Describing workflows using the gherkin grammar
* Developing manual and automated test assets that may be employed to conformance-test the defined workflows
* Providing real-time, always-up-to-date documentation regarding the workflows and the behaviours of participating actors in these workflows.

The proposed action is for the OpenHIE communities to formally adopt HipTest as the shared tool for documenting OpenHIE’s workflows. Further, the OpenHIE Architecture Community would provide an over-arching governance over the use of HipTest so that it was consistently adopted and so that a re-usable family of underlying “nouns and verbs” are employed across our workflow definitions. (Technically… these nouns and verbs are referred to in HipTest as “action words” – and it is at the action-word library level that test automation is operationalized).

## Action steps

A HipTest project has been set up (<https://app.hiptest.com/projects/63905>) to house OpenHIE’s workflows. The following are currently established as administrators on this HipTest project:

* Carl Fourie
* Jennifer Shivers
* Richard Stanley
* Luke Duncan

The recommended action steps are to:

1. Formally adopt this approach
2. Establish the Architecture Community’s governance and operationalize same
3. Set up community leaders as administrators on the HipTest project
4. Leverage the HipTest tool as a way to formally define OpenHIE’s mandatory actor behaviours and to support conformance testing of these actors.