Draft Implementation Guidelines - living document

Version: 0.3
Date: August 29, 2013
Author: Valerie Smothers
Author email: vsmothers@jhmi.edu

If you would like to contribute to this document, please contact Valerie Smothers for login credentials.
Version History

Version No.	Date	Changed By	Changes Made
0.1	13 Nov 2012	Valerie Smothers	Initial outline
0.2	7 Dec 2012	Valerie Smothers	Modified the outline
0.3	29 Aug 2013	Valerie Smothers	Added information from eVip and edited sections on Conformance, Case Exchange, and Integrating Assessment Questions

Acknowledgements

Overview - Valerie

Background...

The MedBiquitous Virtual Patient Standard provides a data structure that allows one to represent …. This structure then enables ….

This implementation guide provides general guidance for common implementations of the MedBiquitous Virtual Patient Standard version 1.0. Specific adaptations for your environment may be necessary.

A broad description of the purpose of the standard and who it is for (FAQ for all, rest more technical). Incorproate white paper form website?

Importance of MVP standard for sustainability (also covered in PPT but good to stress this)

For Educators

FAQ - The group will send in ideas and answers

Why bother adopting the spec?

How to write to the spec – i.e. Where to find the core information and greater details

How much extra work is it typically to do this for most implementers?

Where to find case studies on what others have done

How do you include multiple choice questions in a conformat virtual patient?

The best way to include multiple choice questions in a virtual patient is to use the extension capability of the VirtualPatientData file to include multiple choice question data represented using the IMS QTI specification. Different authoring tools handle multiple choice questions in different ways. If you plan to exchange cases with another organization, be sure to agree on the mechanism for representing multiple choice questions.

For Programmers

General Guidelines

- Central essential aspects of the spec which some software authors may find hard to include or address

The MVP Architecture

Explain Virtual Patient Data, Data Availability Model, Activity Model, and Media, along with how they interact with one another

concise but only semi-technical description (1 page max) of how the pieces fit together (the DAM etc) - this will be hard to do

Notes on conformance

1. What are the downsides of close adherence to the MVP format?
   - What do you lose if 100% conformant?
   - Some caveats about what you typically lose in porting cases from one system to another

Content Packages

To be a conformant instance of MedBiquitous Virtual Patient, a content package:

- Shall contain virtual patient data, activity model, and data availability model XML documents that validate against the MedBiquitous Virtual Patient (ANSI /MEDBIQ VP.10.1-200x) XML schemas available at http://ns.medbiq.org/virtualpatientdata/v1/virtualpatient_schemas.zip
- Shall be a conforming XML instance of a SCORM 2004 4th Edition Resource Content Package Application Profile or Content Aggregation Content Package Application Profile
- Shall conform to any additional requirements stated in this specification (see 5.2.3 VPDText, 6.1 Media Resource Foundations, 6.2.2 Organizations, 6.2.3 Resources, 6.2.3.1 Referencing within a Manifest, 7.2.3 DAMNodeItem, 8.2.6 Rule, 8.2.11 ActivityNode, 8.2.14 ConditionalRule, 8.2.16 ConditionType, 8.2.19 Link)
- May include elements not defined in this document only by using the XtensibleInfo container element defined in the section XtensibleInfo.

NOTE: the definition above comes from the spec. I think we need to stick with that and delete the grayed out text from evip below. I don't want multiple definitions of conformance.

The eViP application profile of the MedBiquitous Virtual Patient standard defines four conformance levels:

Level 1 - Package validation

The first and lowest level of conformance implies that the archive structure and content is conformant with the eViP profile specifications. This means that the correct directory structure and file names are used and that all required files are present. On this level, the content of the files is not checked.

Level 2 - XML/XSD validation

The second level of conformance requires that the XML files are well-formed and validated against their schemas. This includes validation of XML-id references inside and between XML files in the package, references to media resources, and XPath references in the MVP content files.

Level 3 - Import validation

The bottom-line for the third level of conformance is that the author has a clear profit from importing the package into the system. It implies that the target VP player or authoring system extracts and imports the content in a relatively meaningful way.

Level 4 - Runtime validation

The fourth level is the most demanding level of conformance. It states that the imported, packaged virtual patient must run in an eViP-compliant system and is ready for use.

The process of testing for conformance levels 1 and 2 can be totally automated by one of the eViP conformance suites. The third and fourth conformance level can be tested in the target system and needs to be done manually by a learning technologist or subject matter expert. The protocol for eViP conformance testing is summarised by the block diagram in Fig 2.

Players

Conformant players must handle virtual patient data in the manner described in the document ANSI /MEDBIQ VP.10.1-2010 MedBiquitous Virtual Patient Player Specifications and Description Document.

Conformance testing

You can use a stepwise approach to testing conformance of content packages. the following is adapted from documentation created by the eViP project for developing a shared repository of virtual patients.

Two test suites are available to assist with validating the content package (are these test suites up to date with v 1.0 of the spec, or will they accept things from previous versions?)

http://code.google.com/p/mvptools/
http://code.google.com/p/mvp-evip-xslt-test-suite/

These tools go through several steps described below.

1 Package validation

Verify that the zip file contains the necessary files in the correct locations.

2 - XML/XSD Validation

Check the XML files for syntactical and structural errors. The files need to be well-formed (i.e. conform to the XML syntax rules) and valid (i.e. conform to semantic rules defined in the given XML schema files).

Examples of errors encountered on this stage involve problems with XML Data Types like date and time (e.g. different time notations) or numerical fields (e.g. using text comments instead of digits). Further problems encountered included typographic errors in text constants or XHTML tags disallowed by MVP being present in text VPDText elements.

The role of the conformance suite is merely to indicate errors, including errors caused by wrong identifiers or missing references. For more detailed analysis of XML conformance, XML development tools like Altova XML Spy or <oXygen/> XML are recommended.

References, either full XPaths or just identifiers, need to point to existing XML elements or assets outside the XML document. If this is not the case (either due to a typing error or a missing file in the package) an error message is displayed.

3 Import and Runtime validation

Tests on the import and runtime need to be verified in the context of the source and target virtual patient systems.

Potential problems may cause different encoding of special or national characters. In addition, systems may have different navigation models. When moving cases from a branching system to a non-branching system, the case editor will need to select the desired path through the content.

Context Specific Requirements

Before implementing the MedBiquitous Virtual Patient Standard, analyze the context in which it will be used and determine:

1. Your pedagogical goals for the system.
2. Which features your virtual patient will support, and what data elements and attributes correspond to those features.
3. If you plan to exchange Virtual Patient cases with other partners, are there specific features or elements that they require?

Use the rest of this document to help you make decisions about your system and implement the standard in ways that support your goals.

The MVP XML schemas may be modified to support context-specific requirements and restrictions. For more information, see "Adapting the Schema to Meet your Requirements."

Implementing MVP Features

Data Features

Types of structured data

Include demographics, medication lists, physical exam, diagnostic test results, interview question, differential diagnosis, interventions. Discuss benefits of using structured data and when that is useful.

Support for medical vocabularies

Vocabularies enhance the analysis that can be done on aggregate curriculum data by providing clearly delineated terms from which users can select the terms most appropriate to describe a given characteristic of the curriculum. When users choose non-vocabulary terms, it can be difficult to identify trends and determine commonalities and inconsistencies. Suppose Institution A classifies a course using the MeSH term Quality Improvement, and Institution B classifies a similar course using the non-MeSH term Performance Improvement. Anyone compiling data using the term Quality Improvement will have an incomplete dataset because data from Institution B's course will be omitted.

MedBiquitous recommends using agreed-upon vocabularies for the following elements:

Formatted text

Integration of Media resources

Organizing Virtual Patient Data

Feedback

Interactive interview questions

Support for differential diagnosis providing feedback on the likelihood fo the diagnosis and the diagnosis attributed by the case author

Learners may select interventions and receive feedback on the appropriateness and results of the intervention

Using the DAM to provide feedback

Data display options

The ability to show the learner data upon selection

The ability to show the learner data at a later point in the activity

The ability to show the learner data if they previously selected (ie ordered a test, prescribed a medication, etc).

Navigation

Section based navigation

Global navigation

Creating a content path

Branching logic

Probabilistic navigation

Rule-based navigation

Using Timers

Scoring Learners

Some features the MVP standard supports include:

Transferring Data in Languages other than English

Many implementers may wish to use the MVP Standard to exchange virtual patient activities in languages other than English. The standard is designed to support the exchange of activities in languages other than English.

The following elements have a required vocabulary in English:

The Virtual Patient Working group recommends specifying:

- Which languages may be used for data exchange
- What vocabularies will be used for the following elements with recommended vocabularies in English: ….

Organizations may decide to translate the recommended vocabularies referenced in the standard into other languages. See the section "Using Vocabularies" for more information on the specifics of using translated vocabularies.

Finding and Sharing

Metadata

What metadata is important to include to maximize discoverability of cases when stored in repositories or just generally online

What metadata helps with portability of cases between systems

Case Exchange

NOTE: This section is adapted from the eViP Best Practice Guidelines for the eViP application profile and associated conformance metrics.

Virtual patient systems have varied underlying models and distinct features.

eViP recommendations:

- **Allow the user to override an automatic export if the content is not foreseen by the MVP specification.**
  For important content for some systems that are not supported by the MVP specification because of the complexity (e.g. QTI assessment items [19]), it may be helpful to let the user decide whether the questions will be exported in the XtensibleInfo and potentially ignored by the target systems or imported directly as VPD Text for a non-interactive display that need to be manually converted in the target system. Such an option is available in, for example, the CASUS system.
- **The sender and receiver VP systems should use the same version of the standard.**
  Details change between versions, so agreing on the version to use is important.

Recommendations for branched systems

- **A straightforward import of a branched into linear or semi-linear model is difficult.**
  All the solutions investigated by the consortium during the implementation process have led to the loss of information or have required human intervention. However, as was demonstrated by eViP partners, achieving some level of data exchange is feasible.
- **Consider pre-processing of the activity node.**
  One method of importing branched content into a (semi-)linear model is to take all ActivityNodes and transform them using XSLT to one card. **(What is a card?)** This includes also all media resources grouped by DAM nodes referenced from the ActivityNodes. Next it is the task of the author to redesign the content into a (semi-)linear structure.
- **Consider transforming the activity model into a directed acyclic graph** (I don't follow this section - could someone from evip help? Is this tool specific?)
  The other possibility for converting a branched model into a (semi-)linear one is to transform the activity model into a directed acyclic graph (DAG). This model enables multiple functionalities:

- Request of the start node
- Request of the end node(s) (list)
- Request of all previous nodes of a node (list or null if start node)
- Request of all next nodes of a node (list or null if end node)

If available in the original system weightings can be included in the graph. The algorithm starts by importing a start node. The list of potential next nodes is displayed in each imported card. The author can then decide about the order of cards. Unused nodes can be imported as additional comments to the model or as feedback in questions provided that the QTI feature is implemented.

- **Assign weights to the edge between nodes prior to exporting a branched VP**
  If possible, it is recommended for the branched system to include in the export function the weights (credit) assigned to the edges between nodes of the activity model. This feature gives the possibility for importing systems to use algorithms for finding the critical pathway through the graph that could aid the import process. However there is no guarantee that the optimal path will reflect the intentions of the original case, because a great deal of potentially interesting information that lies outside the optimum path may be omitted.
- **Provide a graphical model of the VP**
  If possible, it is recommended for the branched system to export a graphical model of the case to support the orientation while repurposing the case in the target system with a different system model.

Recommendations for (semi-)linear systems

- For systems with a (semi-)linear structure and no specialisation of cards in the data model it seems to be a more pragmatic solution to export all content as VPDText nodes rather than manually assign the cards to particular classes of VPD data (like Diagnosis, Physical Exam or Intervention). However, if there is a way to map the data elements to a specialised tag this method is encouraged.

Recommendations for terminology-based systems

- In terminology-based systems (e.g. CAMPUS), it is impossible to import unstructured text (like that from VPDText) into a highly structured database model. For those systems, often the only possibility for an import is to store the data externally and develop a new MVP/eViP player (as it was done in the case of CAMPUS) that is integrated with the legacy system.
- While exporting data from terminology-based systems, their implementers may face the problem of losing some parts of case related data due to the less structured Virtual Patient Data model in the MVP specification. In those cases, either exporting the content as VPDText or the use of the XtensibleInfo elements is recommended, containing any data that is too specific for general use. For the second option it is recommended to define a separate namespace and a corresponding schema file. External systems may take advantage of this data if they implement this particular kind of extension. The remaining tools will automatically ignore the XtensibleInfo content.

Communicating with would-be users

1. remind implementers to specify in broad terms that a non-techie can grasp to some extent what is needed
   - Hardware, operating system, additional modules, server side software
   - What the end user needs on their system: browser only, Java app, etc
   - Is the service run from a central service, the institution's own servers, a public or cloud based server, or a hybrid of these

XML Tips - Valerie

Schema Locations edit

In order to validate Curriculum Inventory XML documents, you may wish to store all of the associated schemas on a local server and reference those local copies for validation. To use local copies, the schema locations of the following schemas must be changed within the curriculuminventory.xsd schema document.

- competencyframework.xsd

- competencyobject.xsd
- healthcarelom.xsd
- healthcaremetadata.xsd
- healthcarevocabularies.xsd
- member.xsd

Change the schemaLocation attribute of the import element to change the location used for validation. The following example shows import statements that have been changed to use local versions of the schemas. In this example, the xsd files are all in the same directory as the curriculuminventory.xsd file. The schemaLocation attribute may use relative referencing, so the example schemaLocation references the file name since the file is in the same directory.

<xsd:import namespace="http://ns.medbiq.org/member/v1/" schemaLocation="member.xsd"/>

<xsd:import namespace="http://ns.medbiq.org/competencyobject/v1/" schemaLocation="competencyobject.xsd"/>

<xsd:import namespace="http://ns.medbiq.org/lom/extend/v1/" schemaLocation="healthcaremetadata.xsd"/>

<xsd:import namespace="http://ltsc.ieee.org/xsd/LOM" schemaLocation="healthcarelom.xsd"/>

<xsd:import namespace="http://ns.medbiq.org/competencyframework/v1/" schemaLocation="competencyframework.xsd"/>

<xsd:import namespace="http://ns.medbiq.org/lom/vocab/v1/" schemaLocation="healthcarevocabularies.xsd"/>

Curriculum Inventory instance documents may then reference the local copy of the curriculuminventory.xsd schema in the schemaLocation attribute of the root element as in the example below. In this example, the curriculuminventory.xsd schema is in the same directory as the instance document.

<CurriculumInventory xsi:schemaLocation="http://ns.medbiq.org/curriculuminventory/v1/ curriculuminventory.xsd" xmlns="http://ns.medbiq.org/curriculuminventory/v1/" xmlns:lom="http://ltsc.ieee.org/xsd/LOM" xmlns:a="http://ns.medbiq.org/address/v1/" xmlns:cf="http://ns.medbiq.org/competencyframework/v1/" xmlns:co="http://ns.medbiq.org/competencyobject/v1/" xmlns:hx="http://ns.medbiq.org/lom/extend/v1/" xmlns:m="http://ns.medbiq.org/member/v1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

Please note that changing the location of the schemas used for validation does not affect the conformance status of Professional Profile instance document.

Declaring Imported Schema edit

Describe the various VPD schemas and imported schemas. what schemas must be declared and where.

provide example XML showing those declarations and assigned prefixes.

Adapting the Schema to Meet Your Requirements edit

Organizations implementing the MedBiquitous Virtual Patient may wish to further restrict the scope of data considered valid or add new data not addressed in the standard. The schemas are designed to support either of these scenarios.

Extending the MVP EDIT

Add info about Alternative Path in DAM

1. How to sustainably extend the spec

EDIT The Curriculum Inventory schema allows for elements from other namespaces to be included under the root element. Use the steps that follow to extend the Curriculum Inventory schema to incorporate new data.

1. **Write a new XML schema for new data elements and declare a targetNamespace.**
   Develop a new XSD schema that defines the data elements that are missing from the Curriculum Inventory. All new elements must be associated with a namespace. This can be achieved by using the XSD targetNamespace attribute. The following example defines an element called CurriculumDean that can be used to identify the Dean in charge of curriculum. The schema defines http://ns.myurl.com/curriculumdean/ as the targetNamespace, so the CurriculumDean element is associated with that namespace.

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema targetNamespace="http://ns.myurl.com/curriculumdean/" xmlns="http://ns.myurl.com/curriculumdean/" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:element name="CurriculumDean" type="xs:string"/>

</xs:schema>

1. **Place new namespace qualified elements below the root at the end of the XML instance document.**
   Declare the namespace of the schema with new data elements in the instance document. Usually this is done by declaring the namespace in the

root element and assigning a prefix to the namespace. Then the prefix can be used when referencing the new elements. You may also declare a default namespace for an element and its subelements by declaring the namespace in the uppermost element belonging to that namespace. Then include the new element(s) just before the closing CurriculumInventory tag.

In the example below, the prefix cd is declared for the http://ns.myurl.com/curriculumdean/ namespace within the CurriculumInventory root element. The cd prefix is then used to label the CurriculumDean element before the closing CurriculumInventory tag.

Example

Integrating Assessment Questions using Other Standards

The following section is adapted from the eViP profile.

Virtual patient cases are used in both formative and summative assessments. A wide range of different question types is available in contemporary VP systems. In some cases information included in questions and their feedback might be vital for the case resolution. To demonstrate this, an exemplary case from the eViP inventory has been evaluated (evip:vp:1000132). Approximately 48% of the case's original text content has been lost after discarding questions. For that reason inclusion of assessment items into exported VP packages seems to be an important issue to cover. Representation of questions is at the present moment out of the scope of the core MVP specification.

QTI

The Question and Test Interoperability specification (QTI) [19] [20] [21] is the leading format for the exchange of assessment content and results produced by the IMS Global Learning Consortium. The specification defines the QTI information model and its XML data binding. Different question types can be encoded including multiple choice questions, sorting, filling gaps or hotspot. QTI is supported by many virtual learning environments including Moodle, Sakai or Ilias. The latest version available at the moment of writing was 2.1.

The first step is to add the appropriate schema definition file (in the case of QTI 2.1 it is imsqti_v2p1.xsd) to the exported package as it is shown (Fig. 15). (Get code example)

Next, the QTI's namespace is to be declared on top of the virtualpatientdata.xml: (Get code example)

Finally, the QTI conformant XML fragment needs to be inserted as child of the XtensibleInfo node.

Since the XtensibleInfo is by default ignored by MVP conformant VP systems, no negative consequences are to be expected in players that do not support QTI. The content of XtensibleInfo can be either discarded or a warning may be displayed about the fact that the content is not supported.

However, if QTI support is available in the target system, the question is detected and imported.

Other specs

There are many other external specifications beyond QTI that could be potentially useful in virtual patient packages. A notable example is the W3C Timed Text format for encoding of multilingual subtitles in videos (covered in BPG 4). Some VP system builders may also wish to encode in the XtensibleInfo their systems' own distinctive features that are unlikely to be useful in other environments. As an example, part of patient description that is characteristic only for the CAMPUS system is presented in Fig. 19. A similar feature is also implemented in the Web-SP system.

Persistance

The specification correctly makes no assumption about the underlying storage mechanism. However all the systems I know of seem to use SQL Databases. Translating the data from the specified xml structures into a working relational structure and back again is something of a challenge. The advent of NoSQL databases doesn't really help this, because traversal between the structures is complex. The original assumption seems to have been that implementers would work directly from the xml files, but traversal is still complex.

References

Healthcare LOM
ANSI/MEDBIQ LO.10.1-2008, Healthcare Learning Object Metadata (Healthcare LOM). MedBiquitous Website. http://www.medbiq.org/std_specs /standards/index.html#HCLOM. Accessed June 1, 2011.

Glossary

To ensure clarity and consistency, we provide working definitions of the terminology we use in these guidelines.