Enabling Interoperability, Accessibility and Reusability of Virtual Patients across Europe – Design and Implementation

Nabil ZARY a,1, Inga HEGE b, Jörn HEID c, Luke WOODHAM d, Jeroen DONKERS e, Andrzej A KONONOWICZ f

a Virtual Patient Lab, Department of LIME, Karolinska Institutet, Stockholm, Sweden
b Medical Didactic Unit, University of Munich (LMU), Germany
c Centre for Virtual Patients, University Clinic of Heidelberg, Germany
d Centre for Medical and Healthcare Education, St George’s University of London, UK
e Department of Educational Development and Research, Maastricht University, The Netherlands
f Department of Bioinformatics and Telemedicine, Jagiellonian University, Kraków, Poland

Abstract. Virtual Patients (VPs) have successfully been integrated into medical and healthcare curricula for a number of years. Lack of time and resources is a frequently reported problem encountered when developing VPs for teaching and learning. Consequently there is a need for cross-institutional repositories of VPs. The aims of the study were two-fold: to enable interoperability between virtual patient systems and to investigate if (and how) an application profile is implemented in four different types of VP systems. This European collaborative implementation of a blend of several specifications (Medbiquitous VP XML, Medbiquitous Healthcare LOM, and SCORM) is innovative and the study has shown a variation in how the application profile could be implemented.

Keywords. standardization, virtual patients, simulation, interoperability

1. Introduction

Virtual Patients (VPs) have been applied successfully in medical and healthcare education for a number of years [1]. Lack of time and resources is a frequently reported problem encountered when developing VPs for teaching and learning [1]. A substantial amount of effort is also wasted in duplicative work. There is consequently a need for cross-institutional repositories of VPs to enable and enhance the sharing of content. The solution seems to be the creation of a large, varied and efficient bank of VPs with a common interoperability standard that facilitates the migration of data between diverse VP systems. In practice it is not always feasible to implement whole specifications. Many of the features provided by the standards may be outside the areas of interest for a particular target-user community. On the other hand, some aspects of a general purpose specification may be defined as being too broad and need to be constrained. To

1 Corresponding Author: Dr. Nabil Zary, Department of Learning, Informatics, Management and Ethics, Karolinska Institutet, 17177 Stockholm, Sweden; E-mail: Nabil.Zary@ki.se.
solve these problems and accelerate the implementation of standards the concept of Application Profiles has been introduced. An Application Profile is defined as an assemblage of “data elements drawn from one or more namespaces, combined together by implementers, and optimized for a particular local application” [2]. The aim of application profiles is not the declaration of new terms and definitions, but the selection and re-use of existing elements to tailor to the needs of a given group of users.

The aims of our study were:

- To design an application profile that enables interoperability between virtual patient systems.
- To investigate if (and how) the application profile is implemented in different types of VP system.

The intended outcomes were to contribute methods for enabling interoperability between VP systems and knowledge about how to increase accessibility and reusability of VPs across players and institutions.

2. Material and Methods

2.1. Participants

This study was conducted as part of a project co-funded by the European Union. The aim of the European Virtual Patient project (eViP) is to create a bank of repurposed and enriched multicultural virtual patient cases from across Europe [3]. Nine European universities using 4 different VP systems participated in this project.

2.2. Material

The four VP systems (CAMPUS, CASUS, OpenLabyrinth, Web-SP) included in the study have similarities and differences with regards to the features available as well as their didactical approach [4–7]. The navigation model enables the division of the systems into three major categories with different properties. VPs in CASUS follow a linear model which is comprised of linearly ordered cards. A card consists of a text field, may contain media resources (as images and videos), hyperlinks, and (optionally) a question of diverse types. The transition from one card to another is often conditional upon answering a question.

The semi-linear model, represented by CAMPUS and Web-SP, enables a choice to be made from long lists of options. The VPs are usually built from templates containing many standard interview answers, typical physical examination results or laboratory values. The default entries are modified to show symptoms of the given disease. The navigation between different stages of patient treatment is often linear.

Finally, the branched model in OpenLabyrinth can be represented by a graph consisting of nodes (similar to cards in the linear model) interconnected by edges representing the potential decisions of the learner. Completing a case requires choices to be made at key scenario points with the consequences of these choices affecting the final path through the case.
2.3. Method

First, the partners worked collaboratively toward the design of an application profile (eViP profile) that would allow interoperability between the four VP systems included in the study.

Then each VP system implemented the eViP profile. The data collected focused on the mapping of the Medbiquitous VP XML specification to the different modules of the VP player and on how the import/export procedure was integrated in the graphical user interface. It was agreed that the export should be managed automatically. The import might need some manual adaptations, depending on the models used by the source and target systems.

3. Results

3.1. The eViP Profile

The eViP profile was developed from a set of several pre-existing specifications. The eViP profile is a profile of the Medbiquitous Virtual Patient (version 0.48) including the IEEE 1484.12.1-2002 Standard for Learning Object Metadata (LOM) and Medbiquitous MEDBIQ LO.10.1-2008 Healthcare Learning Object Metadata Specification (Healthcare LOM). Finally SCORM 2004 3rd edition was chosen for packaging the virtual patient content.

Each VP Package contains:
- Virtual patient data (VPD) files, provides the personal and clinical data that is relevant
- Media files
- Virtual patient data XML schema
- Activity model (AM) file, encoding what the learner can do and how he may interact with the VP
- Activity model XML schema
- Data availability model (DAM) file, specifies the aggregation of the VPD and media elements for exposure through the AM
- Data availability model XML schema
- Manifest document that details the files contained in the package and how the activity can be launched within a learning management system
- Manifest XML schemas
- Metadata files and Metadata XML schema
- Virtual patient player and an Index.html file (with javascript files) to launch the activity from a LMS
- General XML schema, DTD documents and XHTML schemas customized for virtual patients
3.2. Implementation of the eViP Profile

3.2.1. CAMPUS and Web-SP

For these two systems, the data can be mapped quite easily since the VPs use all the different Medbiqutious VP Virtual Patient elements:

- **PatientDemographics** for patient information
- **VPDText** for summary of medical history, summary of physical examinations, progression, knowledge questions, links, kinds of treatments, epicrisis and prognosis
- **InterviewItem** for medical history
- **PhysicalExam** for physical and technical exams **DiagnosticTest** for laboratory tests
- **Intervention** for therapies and patient management plans
- **Diagnosis** for diagnosis

In the `virtualpatientdata.xml` specific data is stored in the `XtensibleInfo` region

3.2.2. CASUS

<table>
<thead>
<tr>
<th>CLASSES</th>
<th>AM</th>
<th>DAM</th>
<th>VPD</th>
<th>Manifest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter</td>
<td>NodeSection</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Subchapter</td>
<td>NodeSection</td>
<td>DAMNode</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Card</td>
<td>ActivityNode</td>
<td>DAMNodeItem</td>
<td>VPDText</td>
<td></td>
</tr>
<tr>
<td>Card-Info</td>
<td>-</td>
<td>DAMNodeItem</td>
<td>VPDText</td>
<td></td>
</tr>
<tr>
<td>Card-Question</td>
<td>-</td>
<td>DAMNodeItem</td>
<td>VPDText</td>
<td></td>
</tr>
<tr>
<td>Card-Expertcomment</td>
<td>-</td>
<td>DAMNodeItem</td>
<td>VPDText</td>
<td></td>
</tr>
<tr>
<td>Card-Answer(s)</td>
<td>-</td>
<td>DAMNodeItem</td>
<td>XtensibleInfo (QTI) or VPDText (without QTI)</td>
<td></td>
</tr>
<tr>
<td>Card-Answercomment</td>
<td>-</td>
<td>DAMNode (referenced through ItemComment of answer)</td>
<td>VPDText</td>
<td></td>
</tr>
<tr>
<td>Card-Hyperlinks (internal)</td>
<td>-</td>
<td>DAMNodeItem</td>
<td>VPDText</td>
<td></td>
</tr>
<tr>
<td>Card-Multimedia</td>
<td>-</td>
<td>DAMNodeItem</td>
<td>Resource</td>
<td></td>
</tr>
<tr>
<td>Card-Multimedia-comment</td>
<td>-</td>
<td>DAMNodeItem</td>
<td>VPDText</td>
<td></td>
</tr>
</tbody>
</table>

3.2.3. OpenLabyrinth

The Import/Export Module system was developed as a 3-Tier System and compiled in C#.Net technology. Each of the relevant VP xml attributes, elements and subelements of MedBiqutious standard were designed and implemented as serializable/deserializable class objects. The business abstract class (called **ContractBase**) defines the contract of serialization/deserialization usage of the base class. Each VP xml class, node class, chiildnode class must inherit from this **ContractBase** base class to satisfy serialization/deserialization capability of its constituent nodes/childnodes.
3.2.4. The Import/Export User Interface

The import/export procedure was implemented either as part of the existing authoring environment (CAMPUS, Web-SP) or as separate functionality (CASUS, OpenLabyrinth). It is offered as the only means exporting VPs by OpenLabyrinth and Web-SP, or as one of several export formats by CASUS and CAMPUS.

4. Discussion and Conclusions

All four VP systems successfully implemented the eViP profile. The collaborative implementation of a blend of several specifications (Medbiquitous VP xml, Medbiquitous Healthcare LOM, SCORM) was innovative and the study has demonstrated a variety of ways in which the eViP profile could be implemented. However, further studies are necessary to investigate how this finding could impact on the reusability of the VPs when transferred between different VP systems.

Early conclusions may be drawn based on the results obtained while importing VPs in target systems. VP systems with semi-linear models such as CAMPUS and Web-SP use many specialized fields like “BodyPart” or “InterviewItem”, with relatively few general purpose text elements like VPDText in the Virtual Patient Data description. Systems with linear models (like Casus) are characterized by the frequent use of text-heavy VPDText elements. A distinctive feature of branched VPs is the relatively high number of Link and Rule elements in the Activity Model. Three (CAMPUS, Casus, Web-SP) of the four systems used the XtensibleInfo element to extend the specification. These extensions include QTI elements, Timed Text (TT) Authoring Format 1.0 – Distribution Format Exchange Profile (DFXP) movie subtitles and elements characteristic of individual systems – like an extensive vocabulary-based patient’s profile in case of the CAMPUS system. Some MVP elements in the eViP profile were not used. Examples include conditional rules, counter, timers, specialized body location indicators (e.g., ProximalOrDistal, InferiorOrSuperior) and high level classification fields like species or breed.

Finally, the eViP profile includes components that may be useful for sharing content with the broader virtual patient community, including those institutions that do not have a virtual patient specific system but do have a SCORM-conformant LMS.

References