

Grant Agreement Number ECP-2006-EDU-410030



www.virtualpatients.eu

Best Practice Guidelines for the eViP application profile and associated conformance metrics

| | |
|--------------------------------|---------------------------------------|
| Deliverable number/name | <i>D2.3b</i> |
| Dissemination level | <i>Public</i> |
| Delivery date | <i>09/01/09</i> |
| Status | <i>Draft/Final</i> |
| Author(s) | <i>eViP Technical Reference Group</i> |



eContentplus

This project is funded under the *eContentplus* programme¹,
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

¹ OJ L 79, 24.3.2005, p. 1.

| | | |
|----------|--|-----------|
| 1 | INTRODUCTION | 3 |
| 2 | BPG 1 - EXCHANGING VPS BETWEEN PLAYERS | 3 |
| 2.1 | GENERAL RECOMMENDATIONS..... | 3 |
| 2.2 | RECOMMENDATIONS FOR BRANCHED SYSTEMS..... | 4 |
| 2.3 | RECOMMENDATIONS FOR (SEMI-)LINEAR SYSTEMS..... | 5 |
| 2.4 | RECOMMENDATIONS FOR TERMINOLOGY-BASED SYSTEMS..... | 5 |
| 3 | BPG2 – HOW TO TEST FOR CONFORMANCE | 6 |
| 3.1 | EViP CONFORMANCE TESTING - EXAMPLE..... | 8 |
| 3.1.1 | <i>Level 1 – Package validation</i> | 8 |
| 3.1.2 | <i>Level 2 – XML/XSD validation</i> | 8 |
| 3.1.3 | <i>Level 3 – Import validation</i> | 10 |
| 3.1.4 | <i>Level 4 – Runtime validation</i> | 13 |
| 4 | BPG 3 – HOW TO HANDLE EXTERNAL SPECIFICATIONS | 13 |
| 4.1 | INTRODUCTION..... | 13 |
| 4.2 | POSSIBILITIES FOR EXTENSION OF THE MVP SPECIFICATION..... | 13 |
| 4.3 | ASSESSMENT ITEMS IN VIRTUAL PATIENTS..... | 14 |
| 4.4 | THE IMS QTI SPECIFICATION..... | 14 |
| 4.5 | IMPLEMENTATION..... | 14 |
| 4.5.1 | <i>Other external specifications</i> | 17 |
| 5 | BPG4 – INTERNATIONALISATION OF MULTIMEDIA CONTENT USING SUBTITLES | 18 |
| 5.1 | INTRODUCTION..... | 18 |
| 5.1.1 | <i>The W3C (TT) Authoring Format 1.0</i> | 18 |
| 5.1.2 | <i>The eViP profile of W3C TT AF DFXP</i> | 19 |
| 6 | SUMMARY | 23 |
| 7 | REFERENCES | 23 |

1 Introduction

The four best practice guidelines (BPG 1-4) presented in this document are based on the experience of the eViP Technical Reference Group gained while implementing the eViP profile in four VP systems (CAMPUS, CASUS, OpenLabyrinth and Web-SP). The first guideline (BPG 1) addressed technical issues related to the exchange of VPs between different systems. Both general and system specific recommendations are provided. The second guideline (BPG 2) proposes to implementers a conformance testing process. The third guideline (BPG 3) suggests recommendations for handling external specifications that are not covered by the MVP. The last guideline (BPG 4) offers a solution to the problem of internationalisation of media resources. Finally, in the last chapter an outlook is given of upcoming candidate guidelines that may address further aspects of interest to future implementers.

2 BPG 1 - Exchanging VPs between players

The four systems implementing the eViP profile represent different models of virtual patient systems. We distinguish between:

- Linear systems (CASUS, CAMPUS Key Feature)
- Semi-Linear systems (Web-SP, CAMPUS VP)
- Branched systems (OpenLabyrinth)

In addition to this classification, some of the systems have distinct features – e.g. CAMPUS is terminology-based.

This guideline is the first attempt, using several VP systems, to investigate technical issues of importance when exchanging VPs. The inclusion of additional systems could have derived additional conclusions and recommendations. Implementers are therefore encouraged to critically evaluate if the recommendations are applicable to their VP system.

Two types of recommendations are presented: general recommendations and recommendations relevant to a specific type of VP system.

2.1 *General 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 VPDText 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.

- For optimal compatibility, both the sender and receiver VP systems should use the same version of the MVP specification.

In the MVP 0.48 [10], the use of Media elements directly in VPD items like DiagnosticTest was allowed but this notation was deprecated in the MVP versions higher than 0.48.

2.2 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 the partners, achieving the third compliance level 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. 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

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. Fig. 1 presents an exemplary map created with the VUE tool [26] for a VP (case evip:vp:1000007) from the OpenLabyrinth system.

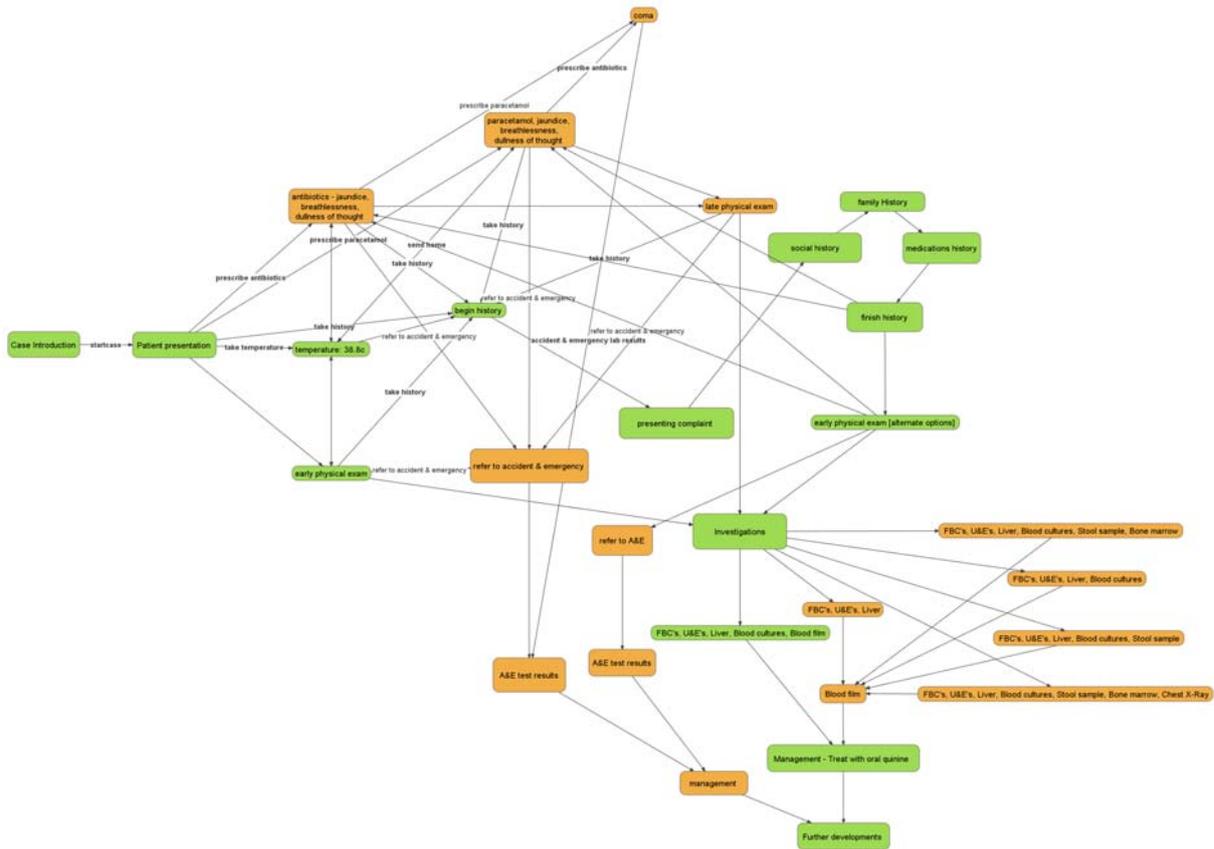


Fig. 1 Graphical model of a VP attached to the exported package (image created with the VUE tool [26] for the case evip:vp:1000007)

2.3 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.

2.4 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.

3 BPG2 – How to test for conformance

The eViP application profile defines four conformance levels (for detailed definitions and more information on conformance testing, we refer to chapter 3 of deliverable D2.2):

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.

Summary of the eViP conformance testing process (simplified)

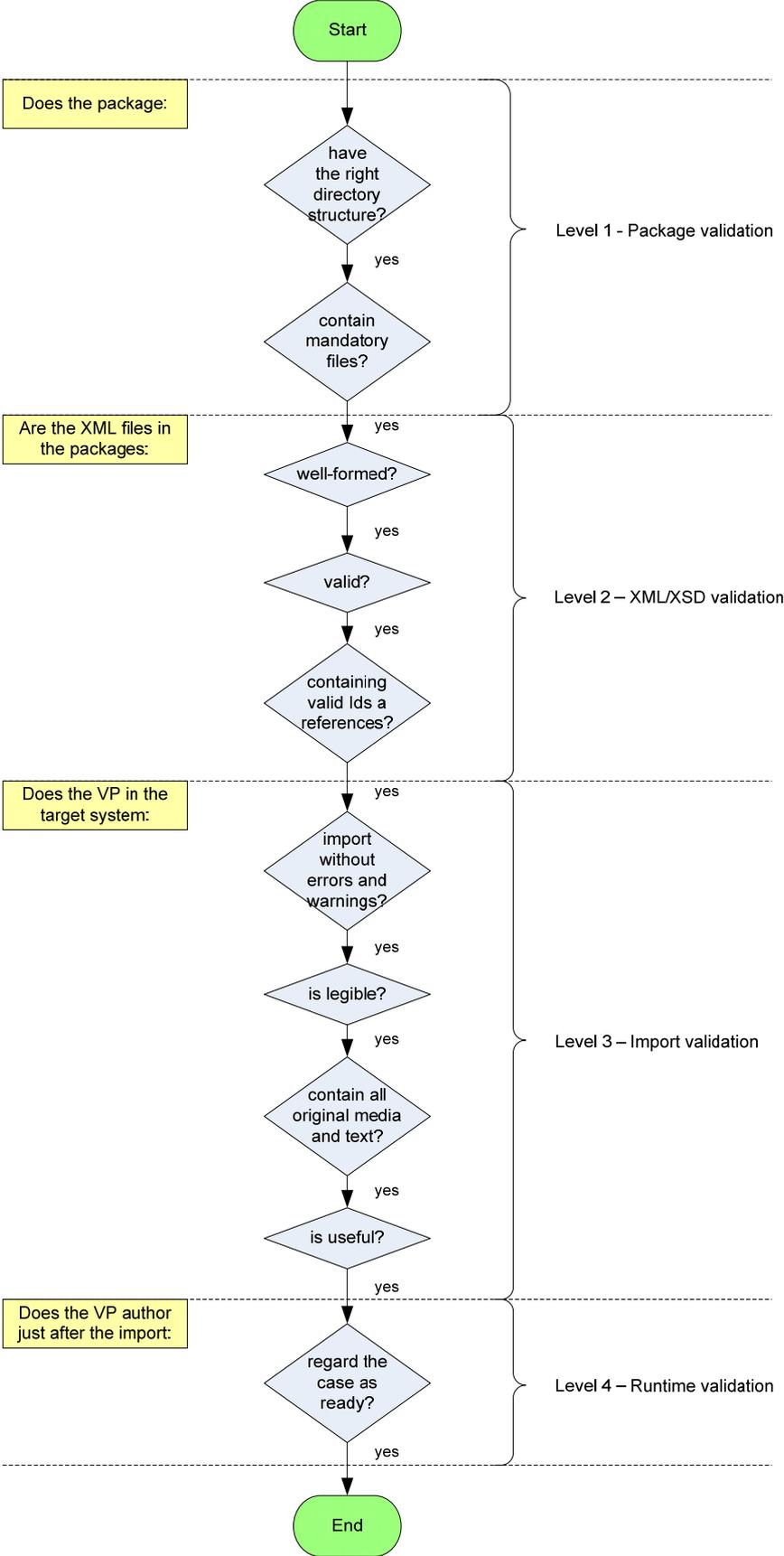


Fig. 2 eViP conformance testing protocol

3.1 eViP conformance testing - Example

The process of eViP application profile testing at consecutive conformance levels is illustrated by a practical example. As a test case a virtual patient package (evip:vp:1000007) from OpenLabyrinth was taken with the aim to be imported into the CASUS system. This process could also be shown on other combinations of cases and partner systems. For demonstration purposes some typical errors encountered while importing cases have been artificially added to the package. Two eViP test suites have been used to verify the package (one developed by KI and one by HD). Both tools may be executed either as web applications or standalone modules. The tools are freely available and can be downloaded (including sources) from the Internet:

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

3.1.1 Level 1 – Package validation

First, the zip package obtained from the source system is verified by the conformance suite developed by KI.

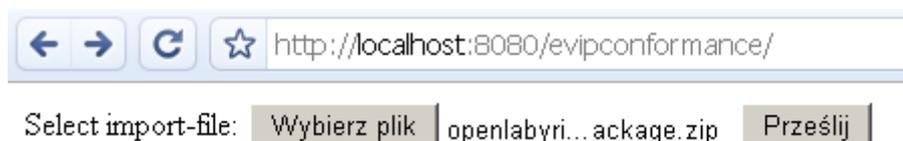


Fig. 3 Start of the conformance testing process – the user selects the package to be tested

On the first level, only the package structure is verified, not the content of files. All mandatory files need to be placed in correct folders in the zip archive beforehand. The most important XML files (e.g. virtualpatientdata.xml, activitymodel.xml or dataavailabilitymodel.xml) need to be placed in the root directory of the package. It is also required to insert all relevant XML Schema Definition files ensuring that they are of the appropriate versions. For the eViP application profile at least MVP version 0.48 is required. In the Fig. 4 the test suite points to some problems with missing XSD files.

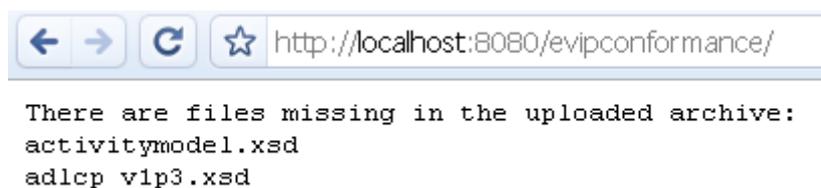


Fig. 4 Missing XSD files –error on the first conformance level

After insertion of the missing files into the packages, the conformance testing may be resumed at the next level.

3.1.2 Level 2 – XML/XSD validation

The major part of the second conformance level is to 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).

Since the MVP specification is still evolving, many of the errors that have been encountered while implementing the profile stem from conformance to outdated schemas.

For instance, up until version 0.47 MVP required the presence of an ItemID element in DAMNodeItem. Afterwards, the element changed to ItemPath. If the VP package contains an outdated element, both eViP conformance suites will detect it and display an error message (as shown in Fig. 5).

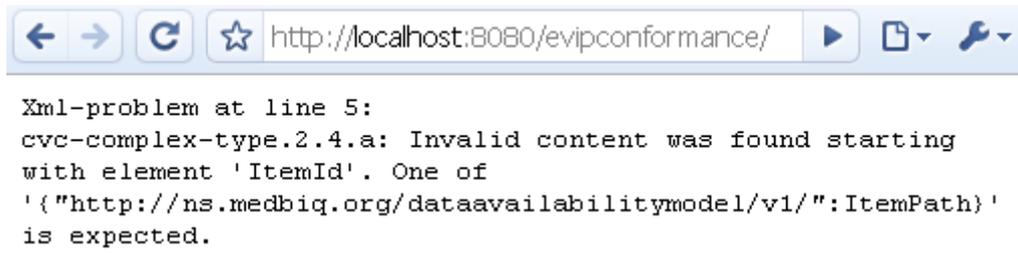


Fig. 5 Invalid XML content – error on the second conformance level

Examples of other 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. For more detailed analysis of the XML and XSD on the second level content editors like Altova XML Spy [13] or <oXygen/> XML [23] are recommended (Fig. 6).

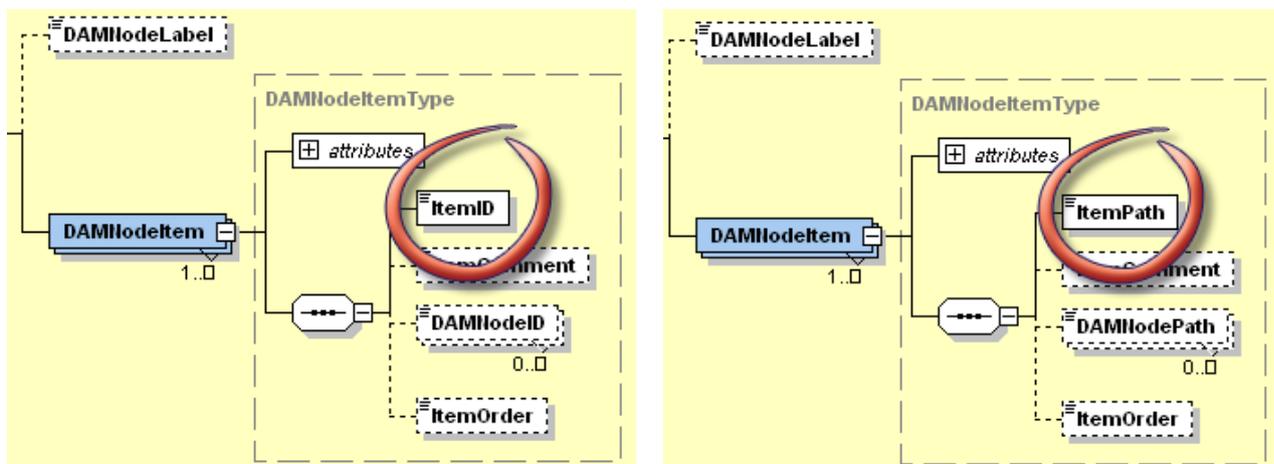


Fig. 6 Analysing of XML Schema files in Altova XML Spy as a method of debugging packages on the second conformance level – this figure demonstrates differences in DAMNodeItem content in MVP v 0.47 (left) and 0.48 (right)

In addition, eViP test suites detect errors caused by wrong identifiers or missing references. The results of detecting errors on this stage are demonstrated on the example of the test suite developed by HD.

It is recommended that identifiers in XML documents are unique and do not start with a digit (Fig. 7).

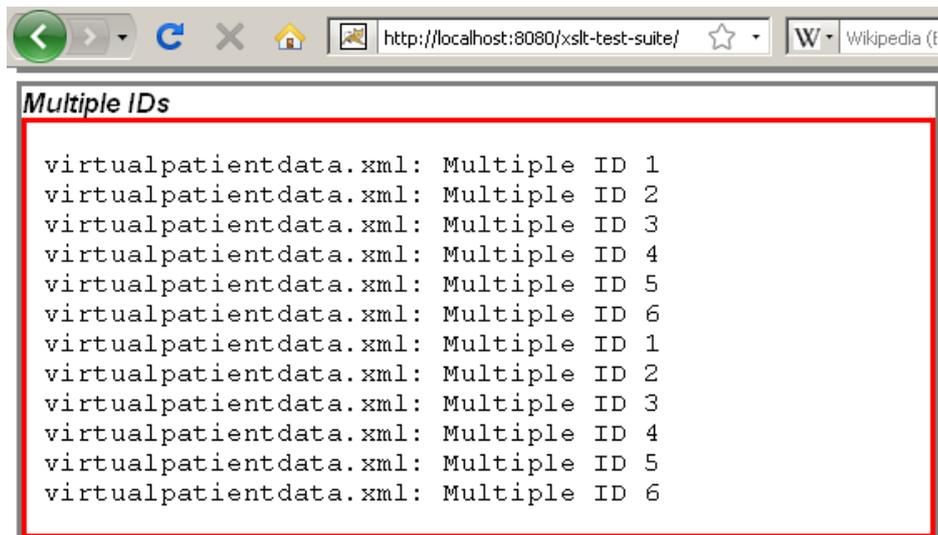


Fig. 7 Problems with wrong and duplicated identifiers – error on the second conformance level

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 (like the one in Fig. 8) is displayed.

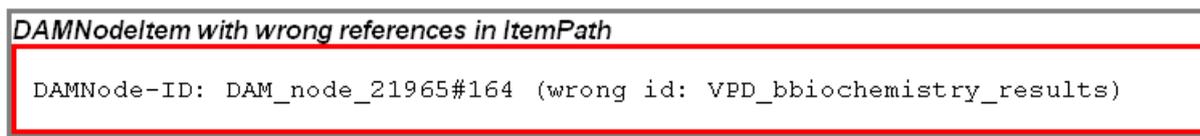


Fig. 8 Problem with a non-existent VPD element – error on the second conformance level

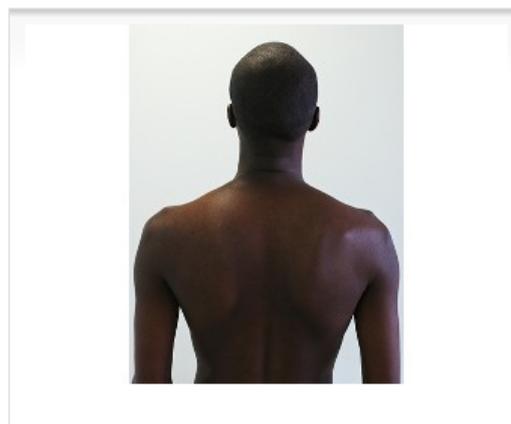
3.1.3 Level 3 – Import validation

Tests on the third and fourth conformance level need to be verified in the context of the source and target VP system (in our case adequately OpenLabyrinth and CASUS). Guided by the checklist from the previous chapter we validate the imported package. Since we have already verified and corrected the package on the 1st and 2nd conformance level the import is accomplished without any errors or warnings. Nodes from OpenLabyrinth are imported as CASUS cards. After opening a card its content becomes visible and is legible and well positioned (Fig. 9).

After prescribing paracetamol Joseph is not any better. He is experiencing breathlessness and dullness of thought. He appears jaundiced.

After prescribing paracetamol Joseph is not any better. He is experiencing breathlessness and dullness of thought. He appears jaundiced.

general observation of patient action= general observation of patient finding= Thin, distressed, no signs of jaundice. No skin lesions (marks, chancres or petechiae. Thin, distressed, no signs of jaundice. No skin lesions (marks, chancres or petechiae. temperature action= take temperature finding= 38.1 Celsius (°C). 38.1 Celsius (°C). Take Blood Pressure Arm action= Use Sphygmomanometer finding= 100/60 systolic=100mmHg,diastolic=60mmHg pulse rate action= feel pulse finding= 120/minute, regular 120/minute, regular auscultate heart action= auscultate heart finding= Ejection systolic murmur Ejection systolic murmur JVP (Jugular venous pressure) action= JVP (Jugular venous pressure) finding= JVP visible 1cm above the sternum JVP visible 1cm above the sternum Respiration action= Respiration finding= Respiration appears normal Respiration appears normal Palpate liver action= Palpate liver finding= liver is palpable 2cm liver is palpable 2cm Neurological exam action= Neurological exam finding= CNS - normal CNS - normal



Which of the following options would you consider doing next?

possible next cards: [[21940]:[21940]:Take patient history, [21941]:[21941]:Prescribe Paracetamol, [21950]:[21950]:Prescribe Antibiotics, [21952]:[21952]:Refer Joseph to Accident and Emergency, [21955]:[21955]:Investigations]

Instruct AG (dbg: id=375389)

Fig. 9 Testing on the third conformance level – Is the content legible?

Potential problems may cause different encoding of special or national characters like in the Fig. 10.

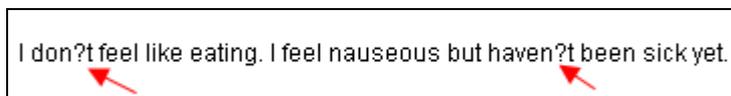


Fig. 10 Problem with the display of special characters – error on the third conformance level

A side-by-side comparison of the OpenLabyrinth and CASUS authoring tools shows that all nodes available in the original have been imported into the target system (Fig. 11).

OpenLabyrinth

Labyrinth is an open source educational pathway authoring and delivery system.

- home
- editor
- global
- nodes
- sections
- links
- authors
- counters
- files
- elements
- clusters
- feedback
- sessions
- preview

release v2.06
check for updates

Edit Nodes of Labyrinth

- 1: [21938]:[21938]:C
- 2: [21939]:[21939]:T
- 3: [21939]:[21939]:T
- 4: [21940]:[21940]:T
- 5: [21941]:[21941]:P
- 6: [21941]:[21941]:P
- 7: [21944]:[21944]:S
- 8: [21944]:[21944]:S
- 9: [21944]:[21944]:S
- 10: [21949]:[21949]:
- 11: [21947]:[21947]:
- 12: [21948]:[21948]:
- 13: [21950]:[21950]:
- 14: [21951]:[21951]:
- 15: [21952]:[21952]:
- 16: [21953]:[21953]:
- 17: [21954]:[21954]:
- 18: [21955]:[21955]:
- 19: [21956]:[21956]:
- 20: [21961]:[21961]:
- 21: [21962]:[21962]:
- 22: [21963]:[21963]:
- 23: [21964]:[21964]:

CASUS

Card Annotations Expert con

Card name: [8]:Case: Joseph Ansah

Comment:

Infotext:

Joseph Ansah is a 25 year old Ghanaian start card possible next cards: [[21953]:

Question text:

Answer type:

Save Preview

Fig. 11 Testing on the third conformance level – Side-by-side deep level checking

The layout of the cards in CASUS, even though not identical to the layout in OpenLabyrinth, retains the meaning of the original content (Fig. 12). The import definitely saves time in comparison to manual extraction of text content and images from the package and step-by-step insertion into the target authoring system.

The screenshot displays the CASUS interface for a patient presentation card. At the top, the OpenLabyrinth logo and navigation menu (HOME, CREATE, MY ACCOUNT, HELP, LOGOFF) are visible. The main content area is titled "Case: Joseph Ansah" and features a large image of a young man sitting at a desk. Below the image, the text reads: "Joseph Ansah is a 25 year old Ghanaian who presents at your surgery, complaining of feeling unwell." To the right, a sidebar contains three sections: "Case Information" (Case: Joseph Ansah, Malaria Demo (700), ID: 21938, Score: 100, with a "Restart Case" link), "Case Pathway" (with a "Review your pathway" link), and "Case Score". Below the main content, a "Patient Presentation" button is shown. The bottom navigation bar includes "Expert", "Clipboard", "Back", "Forward", and a dropdown menu showing "Card 1 of 34 | [21938]:[21938]:Case: Joseph Ansah". The main text area below the navigation bar contains the same text as the card, followed by "start card possible next cards: [[21953]:[21953]:Patient Presentation]" and "Instruct AG (dbg: id=375372)". To the right of the text is a smaller version of the patient's photo. At the bottom right, there are icons for information, search, and navigation, along with the text "Image 1 of 1".

Fig. 12 Testing on the third conformance level – Does the VP contain all original text and images?

3.1.4 Level 4 – Runtime validation

Regardless of the usefulness of the imported package demonstrated on the previous level, it cannot be claimed that the package is ready to use by students at this stage. Due to the differences in the navigation model between the systems (OpenLabyrinth has a branched model, CASUS a linear), the navigation elements (even though visible) are inactive (Fig. 13).

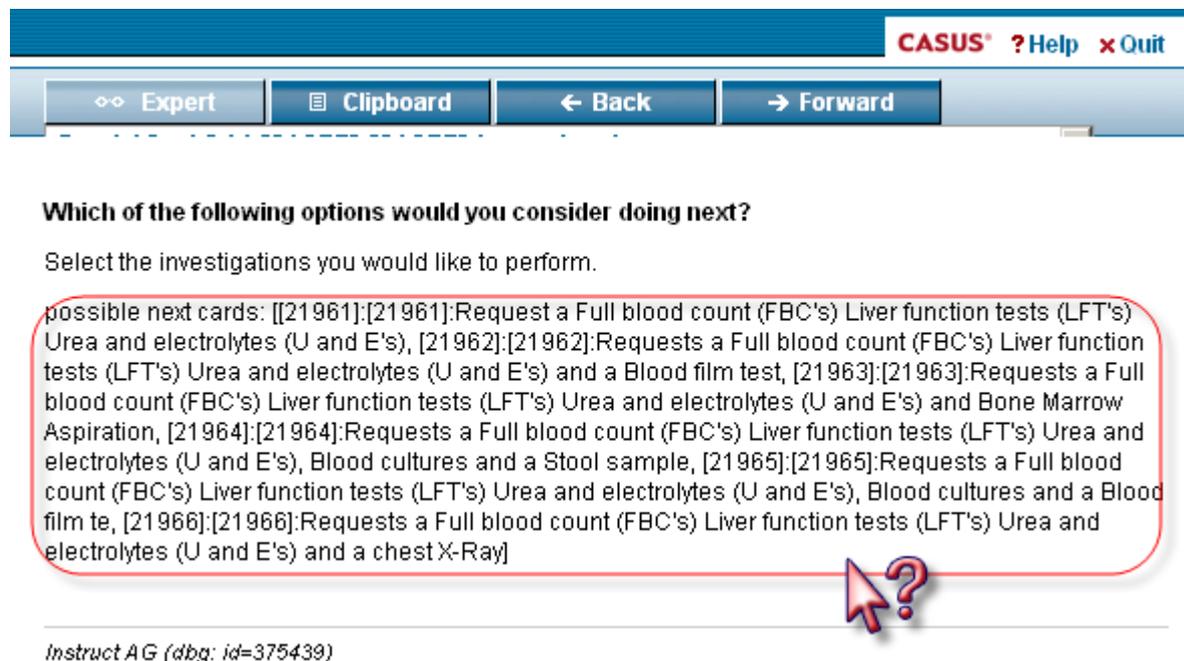


Fig. 13 Challenges of importing VPs from systems with different navigation models

In addition, the order in which the cards are displayed in the player is not yet perfect. It is the task of the content author to select the desired path through the case and delete all the branches. Since there is no reasonable way of doing that automatically we cannot say that the import of this OpenLabyrinth package into CASUS is eViP conformant on the fourth level. We expect that this will be the case for most OpenLabyrinth packages: and conclude that OpenLabyrinth is not compatible at level four for export to CASUS. However, it is clear that the third level of conformance can be achieved.

4 BPG 3 – How to handle external specifications

4.1 Introduction

The Medbiquitous Virtual Patient specification (MVP) [10], which is part of the eViP profile, standardises many aspects of virtual patients. Nevertheless, it is not possible (or desired) for the specification to cover all kinds of content that might be potentially useful in virtual patients. For that reason the authors of MVP allowed the extension of the packages with non-standard elements. The goal of this guideline is to demonstrate how to embed external specifications into MVP using the IMS QTI technology as an example.

4.2 Possibilities for extension of the MVP specification

It is recommended to extend the MVP specification through the XtensibleInfo element from the Virtual Patient Data Schema. This element can be root of any thinkable XML subtree,

provided that the user adds the corresponding schema files to the package and declares a new namespace for the imported content on top of the XML document.

4.3 Assessment items in virtual patients

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.

4.4 The IMS QTI specification

The Question and Test Interoperability specification (QTI) [19] [20] [21] is the world's 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.

4.5 Implementation

The following figure (Fig. 14) presents a CASUS card containing a MCQ question to be included into an eViP profile conformant package.



Fig. 14 Assessment items in CASUS

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).

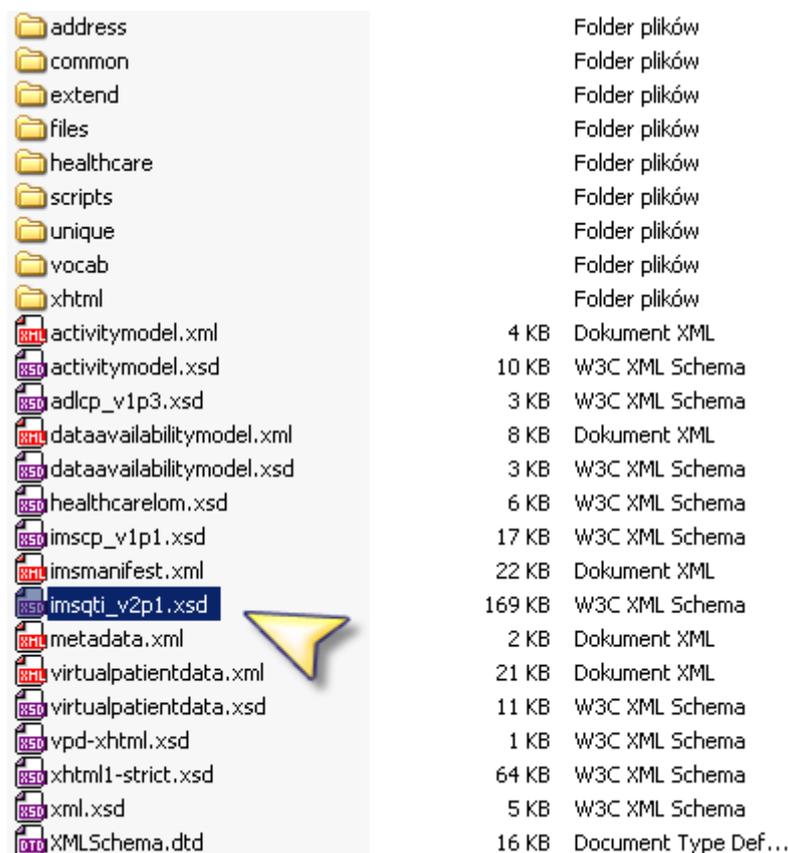


Fig. 15 Adding external schema definition file to the package

Next, the QTI's namespace is to be declared on top of the virtualpatientdata.xml:

```
<VirtualPatientData
  xmlns="http://ns.medbiq.org/virtualpatientdata/v1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:qti="http://www.imsglobal.org/xsd/qti_v2p1"
  xsi:schemaLocation="http://ns.medbiq.org/virtualpatientdata/v1/ virtualpatientdata.xsd
  http://www.imsglobal.org/xsd/qti_v2p1 http://www.imsglobal.org/xsd/qti_v2p1.xsd">
```

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

```
<XtensibleInfo>
  <qti:assessmentItem adaptive="false" timeDependent="false"
  title="no title set" identifier="QTI_12965">
    <qti:responseDeclaration baseType="identifier"
    identifier="RESP_12965" cardinality="single">
      <qti:correctResponse>
        <qti:value>s12965_1</qti:value>
        <qti:value>s12965_2</qti:value>
        <qti:value>s12965_3</qti:value>
      </qti:correctResponse>
    </qti:responseDeclaration>
    <qti:itemBody>
      Which diagnostic procedures are necessary to confirm your
      diagnosis and to evaluate the intensity of the pathologic changes?
      <qti:choiceInteraction responseIdentifier="RESP_12965"
      shuffle="false" maxChoices="3">
        <qti:simpleChoice identifier="s12965_1" fixed="false">x-ray of the chest</qti:simpleChoice>
        <qti:simpleChoice identifier="s12965_2" fixed="false">pulmonary function test</qti:simpleChoice>
        <qti:simpleChoice identifier="s12965_3" fixed="false">sonography for pleural effusion</qti:simpleChoice>
      </qti:choiceInteraction>
    </qti:itemBody>
  </qti:assessmentItem>
</XtensibleInfo>
```

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 (Fig. 16 presents such a message displayed in an early version of the CAMPUS player)

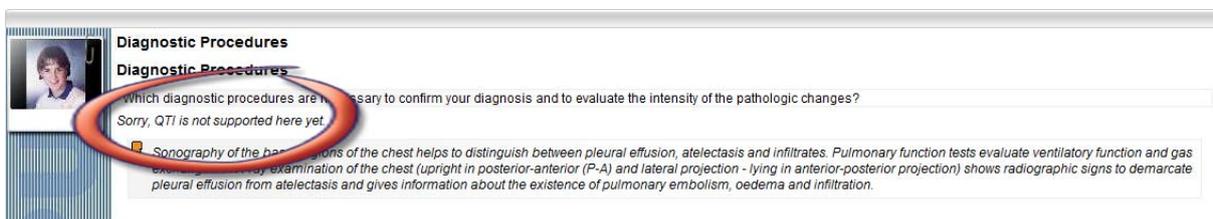


Fig. 16 Handling of unsupported extensions

However, if QTI support is available in the target system, such as in the present version of CAMPUS, the question is detected and imported. This step is presented in the next two figures showing the CAMPUS authoring system just after importing a CASUS package containing questions in QTI format (Fig. 17) and the CAMPUS eViP player displaying the QTI question at runtime (Fig. 18).

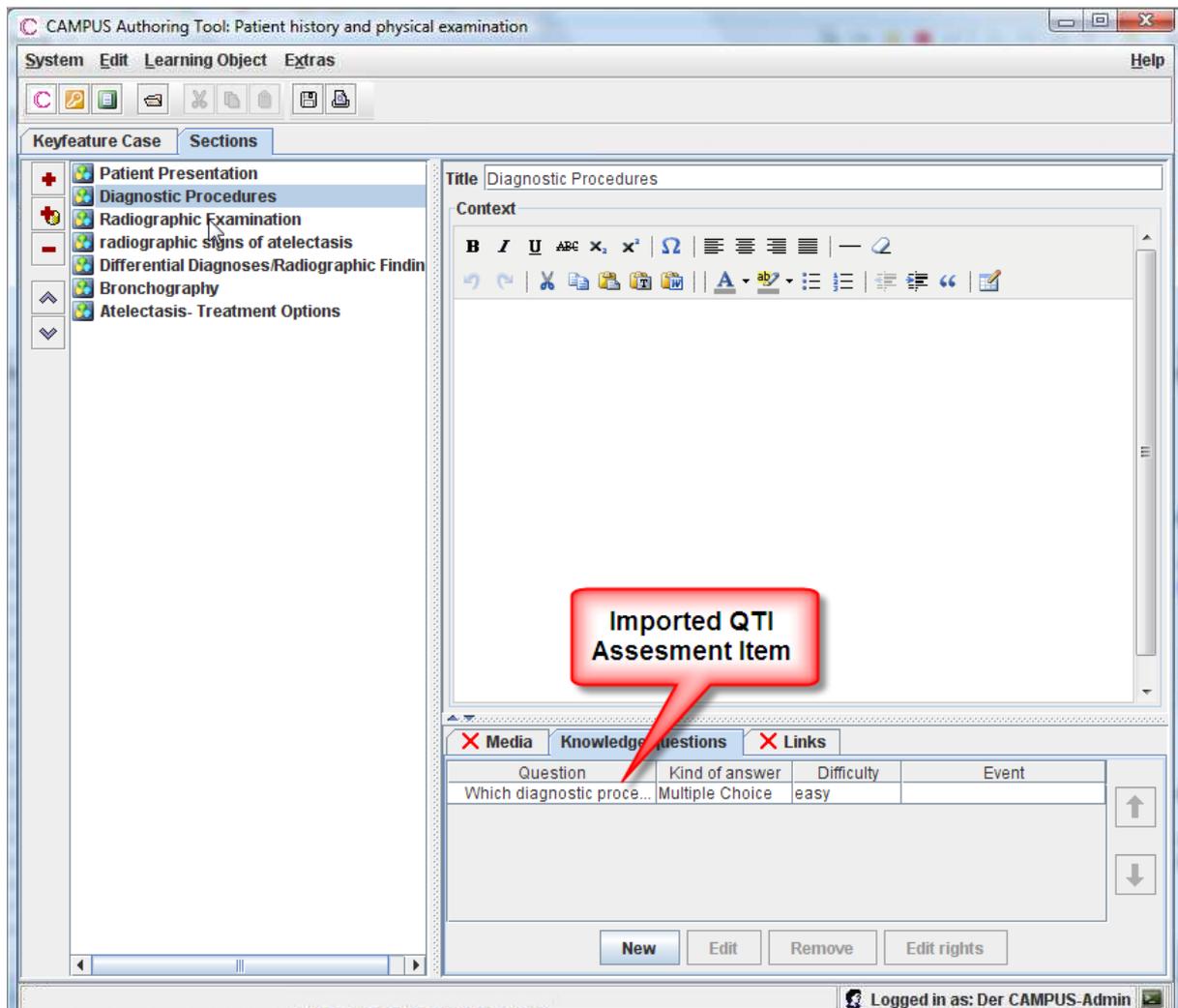


Fig. 17 Imported QTI assessment item in the CAMPUS authoring system

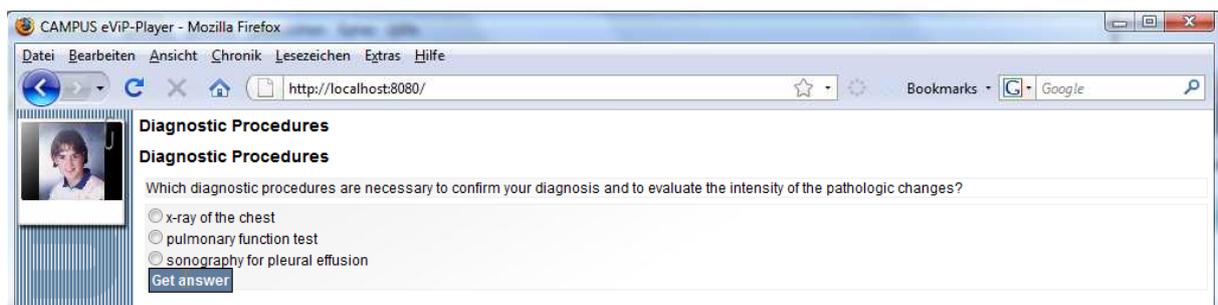


Fig. 18 Imported QTI assessment item in the CAMPUS eViP-Player

4.5.1 Other external specifications

There are many other external specifications beyond QTI that could be potentially useful in eViP profile conformant 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

| | | | |
|------------------------------------|--|----------|------------|
| XtensibleInfo | | | |
| campus:case | | | |
| campus:uci | CAM-1210080739078-68CBC64AF5B7EC819BD4-638095103 | | |
| campus:age-dis... | 5 Monate | | |
| campus:medical-history | | | |
| summary-ref-id | medical-history-summary | | |
| campus:date | 2001-10-29T12:00:00+01:00 | | |
| campus:stage-of-life | voc-id=11 | | |
| campus:medical-history-question | (48) | | |
| campus:knowledge-questions | | | |
| campus:physical-examinations | | | |
| summary-ref-id | physical-examination-summary-dtl-0 | | |
| campus:physical-examination | (41) | | |
| campus:knowledge-questions | | | |
| campus:main-diagnosis | | | |
| campus:diagnosis | ref-id=diagnosis-main-1 | | |
| campus:diagnostic-therapy-loop (2) | | | |
| | progression-ref... | order-no | from-date |
| 1 | progection-dtl-0 | 0 | 2001-10-29 |
| 2 | progection-dtl-1 | 1 | 2001-10-30 |

Fig. 19 System specific (CAMPUS) extension of the eViP profile

5 BPG4 – Internationalisation of multimedia content using subtitles

5.1 Introduction

Virtual patient packages often contain video clips illustrating various stages of a patient's diagnosis or treatment process. Adding text captions is one of the fastest and most convenient methods to internationalise movies and to aid hearing impaired. There are currently two different video standards supported by the eViP VP systems: QuickTime (CASUS, Web-SP) [14] and Adobe Flash Video (CAMPUS, OpenLabyrinth and Web-SP) [12]. For QuickTime, a proprietary format for subtitles (QT Text Tracks) is available, whereas Adobe Flash does not possess its own format for subtitles and uses external specifications. While converting between video standards, subtitles are usually lost. For that reason, and also to facilitate the repurposing process, it has been proposed to add subtitles to virtual patient data description in the eViP profile. Using QT Text Tracks is not the best solution for an interchange format since it is supported only by Apple products and is not XML-based. A better possibility is offered by the emerging W3C Timed Text (TT) Authoring Format 1.0 Distribution Format Exchange Profile (DFXP) [27].

5.1.1 The W3C (TT) Authoring Format 1.0

The Timed Text Authoring Format (TT AF) Distribution Format Exchange Profile (DFXP) is a W3C candidate recommendation for a format to exchange of timed textual information with stylistic and layout semantics. TT AF DFXP has XML syntax defined under <http://www.w3.org/TR/2006/CR-ttaf1-dfxp-20061116>. Documents conformant to TT AF DFXP are composed of two main sections <head> and <body>. The <head> section contains elements specifying the metadata of the subtitles, styling and layout information, whereas the body section contains subtitle sections with specified time intervals in which the captions should be displayed.

5.1.2 The eViP profile of W3C TT AF DFXP

To be able to export and import those texts through different systems, the XtensibleInfo in the dataavailabilitymodel.xml was extended and a self defined namespace was created:

For the purpose of exchange of subtitles from video clips in eViP the TT AF DFXP has been limited to captions with one layout per document, characterised by the following parameters:

- Extent – the size of the caption canvas
- Background colour
- Font family, size and colour
- Caption text alignment
- Text captions are displayed usually in 1-2 lines in white on black or transparent background.

The files are embedded in MVP in the dataavailabilitymodel.xml, in a self defined namespace called evip-captions as presented below:

```
<XtensibleInfo>
  <evip-captions:media-captions>
    <evip-captions:media itemPath="/manifest/resources/resource[@identifier='medium-N100A8']">
      <tt:tt xml:lang="de">
        <tt:body>
          <tt:div xml:id="captions">
            <tt:p end="00:2.0" begin="00:0.0">Test Text 1</tt:p>
            <tt:p end="00:8.0" begin="00:2.0">Test German characters: äöüß</tt:p>
            <tt:p end="00:14.0" begin="00:8.0">Check Polish characters: ąęłśóć</tt:p>
          </tt:div>
        </tt:body>
      </tt:tt>
    </evip-captions:media>
  </evip-captions:media-captions>
</XtensibleInfo>
```

Inside the self-defined media tag which references to a media resource, W3C's Timed Text is used to represent the captions. With the schemas attached, the XML file will completely validate against the W3C's schemas.

5.1.2.1.1 How to add text tracks in Quick Time?

Quick Time Text Tracks [25] are text files with:

- descriptors defining the layout of the subtitles - eg. textColor: 65535, 65535, 65535} for setting font colour
- subtitles cards separated by timestamps in square brackets

The following exemplary QuickTime Text Track presents the above listing:

```
{QTtext}
{font:Geneva}
{plain}
{size:12}
{textColor: 65535, 65535, 65535}
{backColor: 0, 0, 0}
{justify:center}
```

```

{timeScale:10}
{width:320}{height:32}
{timeStamps:absolute}
[00:00:00.0]
Zaczynamy znacznie poniżej łuku żebrowego,
[00:00:05.0]
następnie czubkami palców naciskamy powoli
w kierunku łuku żebrowego,
[00:00:13.0]
prosimy pacjenta, żeby nabrał głęboko powietrze
i wypuścił
[00:00:21.0]
jeszcze raz proszę nabrać powietrza i wypuścić powietrze
[00:00:26.0]

```

Text Tracks are added to QuickTime video clips by dedicated authoring tools e.g. Quick Time Pro (Fig. 20) or directly via the QuickTime for Java API [24].

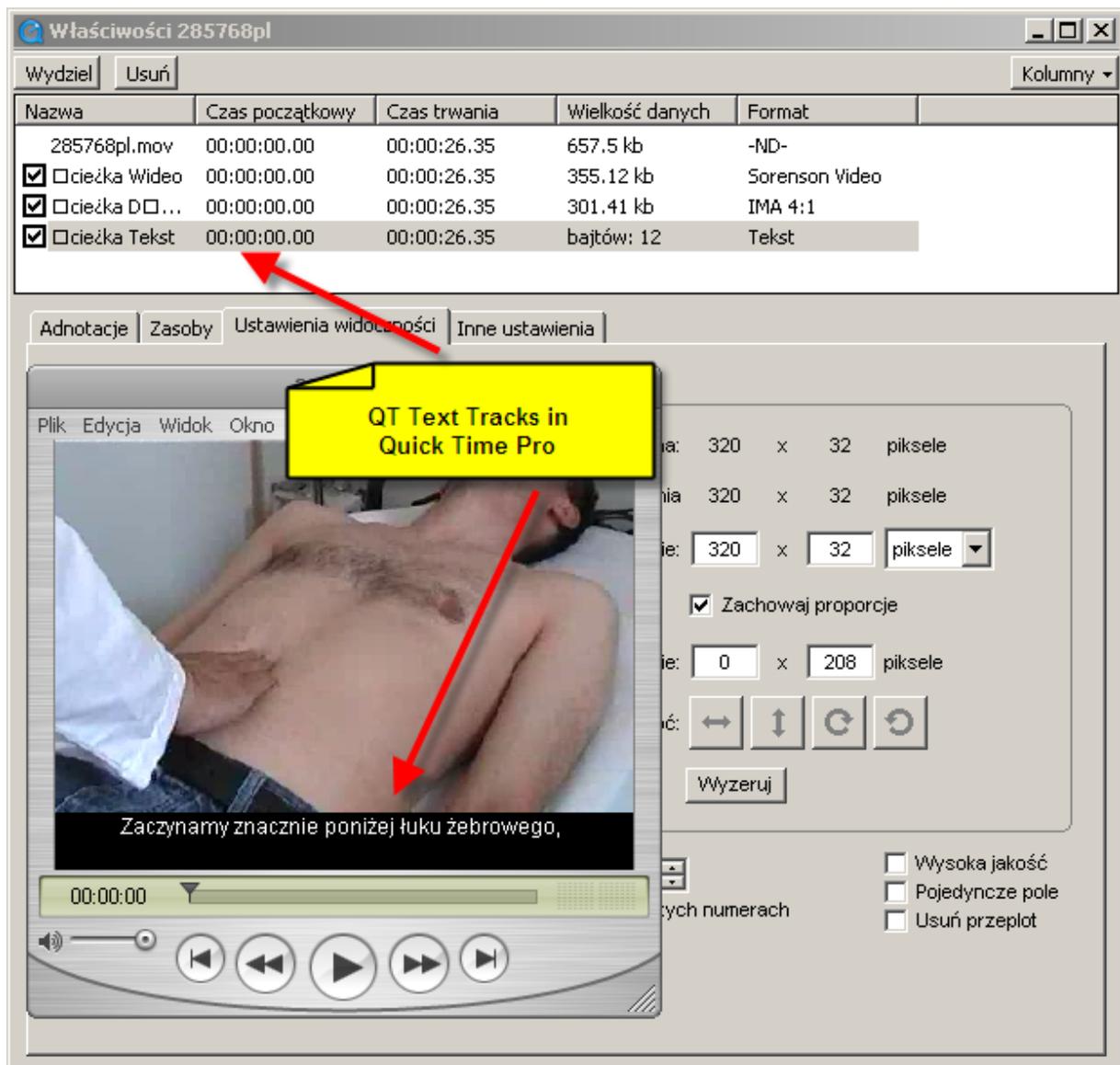


Fig. 20 Adding subtitles in QTtext format in Quick Time Pro

In Fig. 21, a QuickTime movie with added text tracks is presented in the CASUS system.

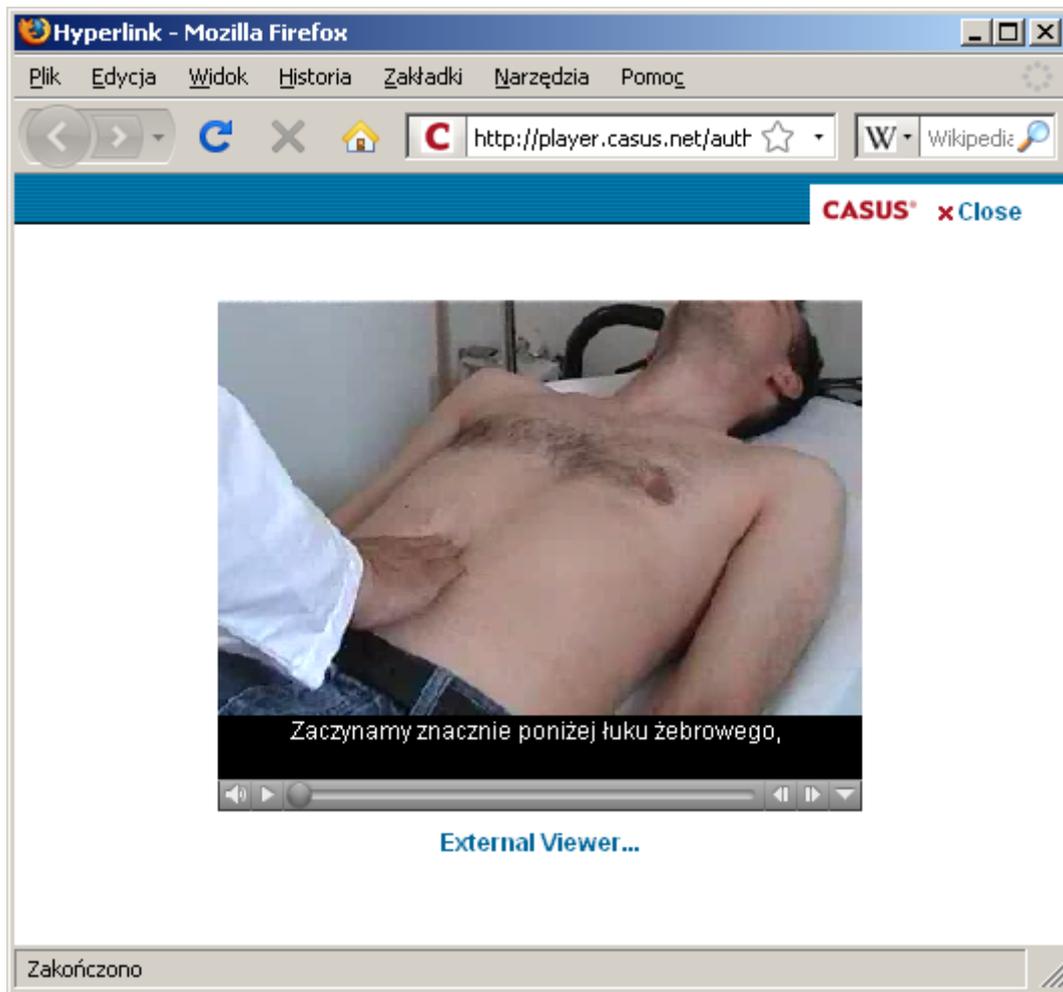


Fig. 21 Subtitles in QTtext format displayed by a Quick Time plug-in in the CASUS player

eViP Profiled subtitles can be converted between Quick Time Text Tracks and W3C TT AF DFXP XML by a command line tool, developed especially for the purpose of the eViP project. The tool is free available and can be downloaded (including sources) from the Internet:

<http://code.google.com/p/evip-qttext-dfxp-converter/>

The program requires JRE 1.5+. Its usage is simple:

To convert eViP DFXP to QTtext use the batch file:

```
dfxp2qt.bat [-ie encoding] [-oe endcoding] [-o file] filename
```

eg.

```
dfxp2qt.bat -oe Cp1250 -o qttext.txt dfxp.xml
```

To convert eViP Qttext to DFXP qt2dfxp.bat:

```
qt2dfxp.bat [-l langCode] [-ie encoding] [-oe endcoding] [-o file] filename
```

eg.:

```
qt2dfxp.bat -l pl -o qttext.txt dfxp.xml
```

The next figure presents the converted QTtext file from the listing above to the TT AF DFXP format.

```

<tt:tt xmlns:tt="http://www.w3.org/2006/10/ttaf1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml:lang="pl">
  <tt:head>
    <tt:layout xmlns:tts="http://www.w3.org/2006/10/ttaf1#style">
      <tt:region xml:id="subtitleArea" tts:extent="320px 32px" tts:backgroundColor="rgb(0,0,0)"
        tts:color="rgb(255,255,255)" tts:textAlign="center" tts:fontFamily="Geneva" tts:fontSize="12px"/>
    </tt:layout>
  </tt:head>
  <tt:body region="subtitleArea">
    <tt:div xml:id="captions">
      <tt:p begin="00:00:00.0" end="00:00:05.0">Zaczynamy znacznie poniżej łuku żebrowego,</tt:p>
      <tt:p begin="00:00:05.0" end="00:00:13.0">następnie czubkami palców naciskamy powoli
        <tt:br/>w kierunku łuku żebrowego,
      </tt:p>
      <tt:p begin="00:00:13.0" end="00:00:21.0">prosimy pacjenta, żeby nabrał głęboko powietrze
        <tt:br/>i wypuścić
      </tt:p>
      <tt:p begin="00:00:21.0" end="00:00:26.0">jeszcze raz proszę nabrać powietrza i wypuścić powietrze</tt:p>
    </tt:div>
  </tt:body>
</tt:tt>

```

5.1.2.1.2 How to add text tracks in Flash?

One way to add captions to a FLV-player is the usage of the JW Media Player [22]. Its accessibility support uses external W3C TT AF DFXP XML files for captions.

The next listing attaches a DFXP caption file obtained with the qt2dfxp.bat tool to the test video from the section before (converted to FLV format by Riva FLV Encoder 2) to be played in JW Media Player.

```

<p id='preview'>The player will show in this paragraph</p>
<script type='text/javascript' src='swfobject.js'></script>
<script type='text/javascript'>
var s1 = new SWFObject('player.swf','player','320','240','9');
s1.addParam('allowfullscreen','true');
s1.addParam('allowscriptaccess','always');
s1.addParam('flashvars','file=285768pl.flv&captions=dfxp.xml&plugins=accessibility');
s1.write('preview');
</script>

```

The obtained result is provided in the next figure (Fig. 22).



Fig. 22 DFXP subtitles converted from QTtext attached to Flash Video files and displayed by the JW Media Player

Following the pattern of QTtext, other potential subtitle formats may be converted to W3C TT AF DFXP and embedded into/obtained from eViP packages.

6 Summary

This document presented a set of recommendations for future implementers of the eViP application profile. Using the basic elements of the MVP specification as a starting point, the eViP project team went through conformance testing of VP packages and continued to address progressively more complex issues such as embedding QTI assessment items and handling of subtitles in VPs. For the reason of dependencies with other work packages, some of the potentially interesting recommendations could not be included in this deliverable and will be handled with more detail in other deliverables. This includes e.g. the storage of consent forms, new eViP-profiled metadata, identifiers and vocabularies, as well as a consistent way of storing VP's repurposing life cycle.

7 References

- [1] Ellaway, R.; Poulton, T.; Fors, U.; McGee, J. B. & Albright, S. (2008), 'Building a virtual patient commons.', *Med Teach* 30(2), 170–174.
- [2] Ellaway, R.; Cameron, H.; Ross, M.; Laurie, G.; Maxwell, M. & Pratt, R. (2006), [Clinical Recordings for Academic Non-clinical Settings](#), JISC Project.
- [3] Ellaway, R.; Cameron, H.; Ross, M.; Laurie, G.; Maxwell, M. & Pratt, R. (2007), Digital Images in Education: Realising the Vision, JISC Collections, chapter [Towards a Clinical Commons: Using Clinical Recordings in Academic Non-Clinical Settings](#), pp. 101-120.
- [4] Herry R.;Patel M. (2000), 'Application profiles: mixing and matching metadata schemas', *Ariadne Issue 25*, <http://www.ariadne.ac.uk/issue25/app-profiles/intro.html> [accessed 22/12/2008]
- [5] Grunwald, T. & Corsbie-Massay, C. (2006), 'Guidelines for cognitively efficient multimedia learning tools: educational strategies, cognitive load, and interface design.', *Acad Med* 81(3), 213–223.
- [6] Horton, W. & Horton, K. (2003), *E-learning Tools and Technologies*, Wiley Publishing, Inc..

- [7] Pfaehler M & Holzer M.(2008) 'Erweiterung des QTI-Standards zur Unterstützung von Long-Menu-Fragen sowie Abbildung spezifischer Metadaten in QTI 2.1', Workshop der Arbeitsgruppe "Computerunterstützte Lehr- und Lernsysteme in der Medizin" der GMDS, Saarbrücken, 34-36
- [8] Rubin, J. & Chisnell, D. (2008), Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct Effective Tests, Wiley Publishing, Inc..
- [9] Sedgewick, R. (2003), Algorithms in Java, Third Edition, Part 5: Graph Algorithms, Addison Wesley.
- [10] Smothers, V. & Azan, B. (2008), 'MedBiquitous Virtual Patient Specifications and Description Document v.0.48', Technical report, MedBiquitous Consortium.
- [11] Smothers, V. & Azan, B. (2008), 'MedBiquitous Virtual Patient Player Specifications and Description Document v.0.48', Technical report, MedBiquitous Consortium.
- [12] Adobe Flash Video <http://www.adobe.com/devnet/video> [accessed 22/12/2008]
- [13] Altova XML Spy <http://www.Altova.com/XMLSpy> [accessed 22/12/2008]
- [14] Apple QuickTime <http://www.apple.com/quicktime> [accessed 22/12/2008]
- [15] eViP Consortium (2008), D2.2 - Report on eViP Application Profile implementation and conformance testing
- [16] IMS Application Profile Guidelines Overview http://www.msglobal.org/ap/apv1p0/imsap_oviewv1p0.html [accessed 22/12/2008]
- [17] IMS Meta-data Best Practice Guide IEEE 1484.12.1-2002 Standard for Learning Object Metadata http://www.msglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html [accessed 22/12/2008]
- [18] IMS QTI Homepage <http://www.msglobal.org/question> [accessed 22/12/2008]
- [19] IMS Question and Test Interoperability Overview Version 2.1 Public Draft (revision 2) Specification http://www.msglobal.org/question/ktiv2p1pd2/imsqti_oviewv2p1pd2.html [accessed 22/12/2008]
- [20] IMS Question and Test Interoperability Implementation Guide Version 2.1 Public Draft (revision 2) Specification http://www.msglobal.org/question/ktiv2p1pd2/imsqti_implv2p1pd2.html [accessed 22/12/2008]
- [21] IMS Question and Test Interoperability Assessment Test, Section, and Item Information Model Version 2.1 Public Draft (revision 2) Specification http://www.msglobal.org/question/ktiv2p1pd2/imsqti_infov2p1pd2.html [accessed 22/12/2008]
- [22] JW Media Player <http://www.longtailvideo.com/players/jw-flv-player> [accessed 22/12/2008]
- [23] <oXygen/> xml editor <http://www.oxygenxml.com> [accessed 22/12/2008]
- [24] QuickTime for Java API <http://developer.apple.com/quicktime/qtjava> [accessed 22/12/2008]
- [25] Quick Time Text Descriptors <http://www.apple.com/quicktime/tutorials/textdescriptors.html> [accessed 22/12/2008]
- [26] VUE (Visual Understanding Environment), <http://vue.tufts.edu> [accessed 22/12/2008]
- [27] W3C Timed Text (TT) Authoring Format 1.0 – Distribution Format Exchange Profile (DFXP) <http://www.w3.org/TR/2006/CR-ttaf1-dfxp-20061116> [accessed 22/12/2008]