Deliverable D 4.2 –

First Release of Visualisation and OLM Services and Tools

Document Version:	1
Document Status:	Draft
Document Type:	Deliverable
Diss. Level:	Public
Lead Partner:	UoB
Delivery Date:	3rd March 2015

Authors: Matthew Johnson, Drew Masci,
Susan Bull (UoB)

CONTENTS

1 INTRODUCTION

D4.2) First release of visualisation and OLM services and tools. This deliverable will cover the first release of the visualisation and OLM web services and tools, including manuals. For proper integration into the global system these components will be delivered in M11, one month prior to the global system platform release.

This deliverable is the accompanying documentation to the first software release of LEA's BOX services that relate to visualisation and open learner model components. Primarily it reports on the development of a generic, reusable visualisation service for use within the project, and also the reappropriation and revision of an open learner model component brought forward from the Next-TELL project. In alignment with the development plan outlined in D2.1, these two components form a baseline and flexible toolset upon which our tools and school studies will be built in Years 2 and 3. This software release is part of the first integrated solution of the LEA's BOX learning analytics toolkit that will be enriched during forthcoming development work.

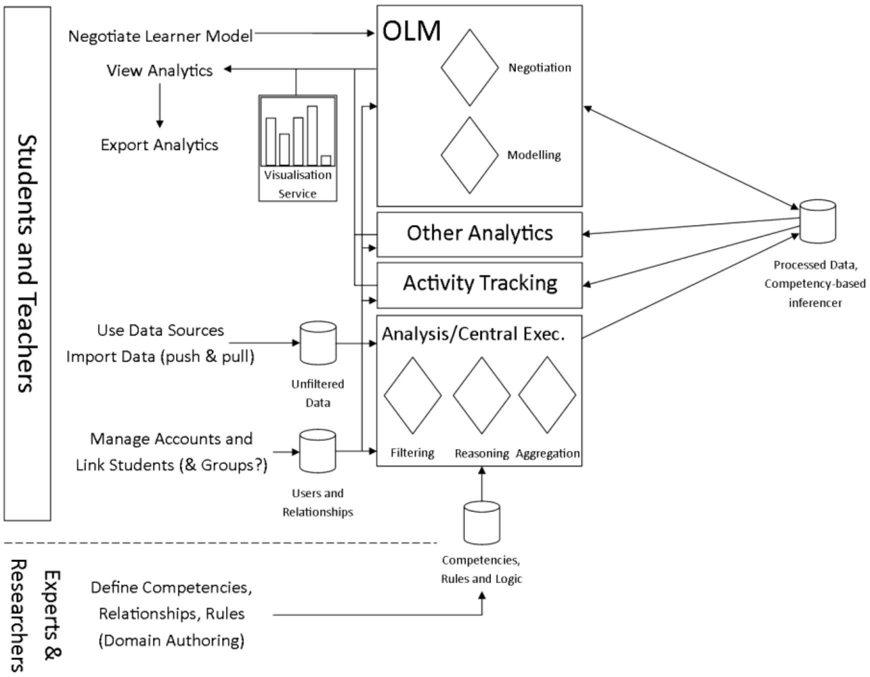


Figure 1: global architecture.

Within the global system architecture (see D2.1 for more information), this deliverable concentrates on components towards the centre top of the diagram (see Figure 1). For the purpose of a modular description of these services, we consider the boundaries of the entities described in this report as per Figure 2. Competency inferences are made available from the central executive or other imported data. These are used to build the learner model, together with information about users and additional pieces of context (such as grouping of users). Furthermore the visualisation services are available to other software elements within the project. The GUI for the visualisation services (namely an open learner model, and any specific calls to the visualisation service) are available for rendering within a web browser and also may be embedded within the LEA's BOX portal.

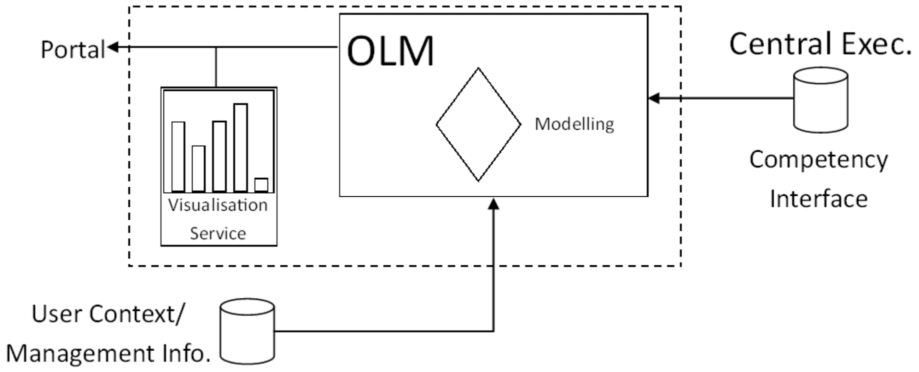


Figure 2: visualisation services – entity boundaries.

As specified in the development roadmap outlined in D2.1 (System Design Document 1) “the aim for the end of Y1 is to have a proof of concept level of integration between core project components, based on the adaptation of existing tangibles.” We therefore structure the remainder of the deliverable as follows: Section 2 summarises the user-accessible interfaces for the first release of the visualisation service; Section 3 describes the amendments made to existing software (namely the Next-TELL OLM) to allow it to form a base for this service; Section 4 details additional components developed (visualisation rendering service, integration functions) to allow for the first version to be flexible and complete enough for the project’s needs; and finally, Section **Fehler! Verweisquelle konnte nicht gefunden werden.** summarises briefly visualisation services available that are embedded within elements of the project’s software.

2 Manual and Description of User-Accessible Interfaces

For release one, the core of the visualisation service constitutes an open learner model application, together with generic facilities for the display of hierarchically structured datasets, such as competency frameworks. A basic level of integration with the LEA's BOX portal has been undertaken, such that competency-based inferences from the central executive may be passed to the open learner model as evidence. This information is then modelled and visualised as per the description contained in this deliverable. The core of the open learner model is taken from the Next-TELL project, and the adaptation and extension of this is covered further in Section 3. The additional services and integration issues of note are covered in Section 4.

2.1 Interface Structure and Components

The interface constitutes one primary webpage, which acts as a browser for the open representation of the learner model (Figure 3). This browser is then embedded within the LEA's BOX portal. It is the same for both teachers and students, with the exception that students see only their own data, whilst teachers can see data for all students with whom they share a group. For the ease of showing the general layout of the screen Figure n uses test data to show where information will appear on the screen. Visualisations are covered in Section 2.3.

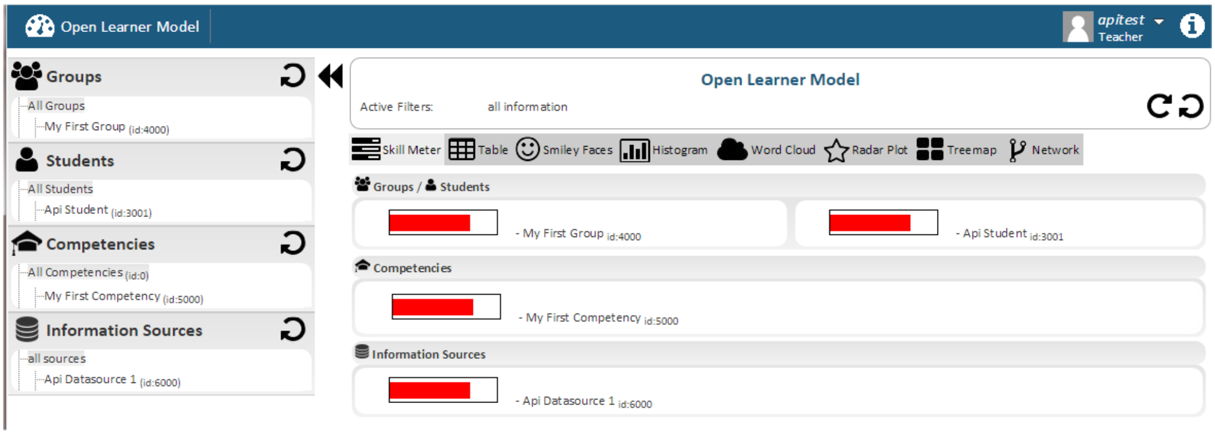


Figure 3: OLM browser interface.

The following key facilities are included:

- **Information filters** (left of Figure 3). These allow criteria to be specified to narrow down the scope of the information presented in the visualisations. These may be added in any combination or permutation. Specific groups, competencies or information sources may be specified. For the case of teachers, individual students may also be specified. Visualisations are automatically updated when criteria are amended. The filters may be hidden to allow more space for the visualisations.
- **Open learner model visualisations** (centre and right of Figure 3). Different visual methods are used to display the same underlying learner model information. These may be switched between using the tab structure. Each set of visualisations is broken down into a learner model opened from the perspective of groups, (students,) competencies and information sources. Each of the sections are collapsible, to allow greater space for individual

visualisations. The visualisations are rendered by posting the relevant modelled dataset to the visualisation service (see Section 4.1) and displaying the returned graphic or HTML content.

- **Breadcrumb and functions** (top centre and right of Figure 3). The filters currently applied and nature of the information in the visualisations are described using a breadcrumb to show where the user currently is within the 'browser'. To the right hand side of this section there are also refresh and reset functions. Additionally affordances such as the loading symbol will appear here whilst the visualisations are loading or updating.
- **Customisation functions** (very top right of Figure 3). The menu which is headed by the user's username allows the browser to be customised. The language may be localised to English, French, German, Czech or Norwegian. The visualisations that are displayed in the browser may also be turned on and off using the preferences page.
- **Help** ('i' icon, very top right of Figure 3). This will display basic guidance on how to operate the browser.

2.2 Iconography and Localisation

Icons are used consistently throughout the browser. These icons are mostly imported from use in the Next-TELL project (Figure 4).

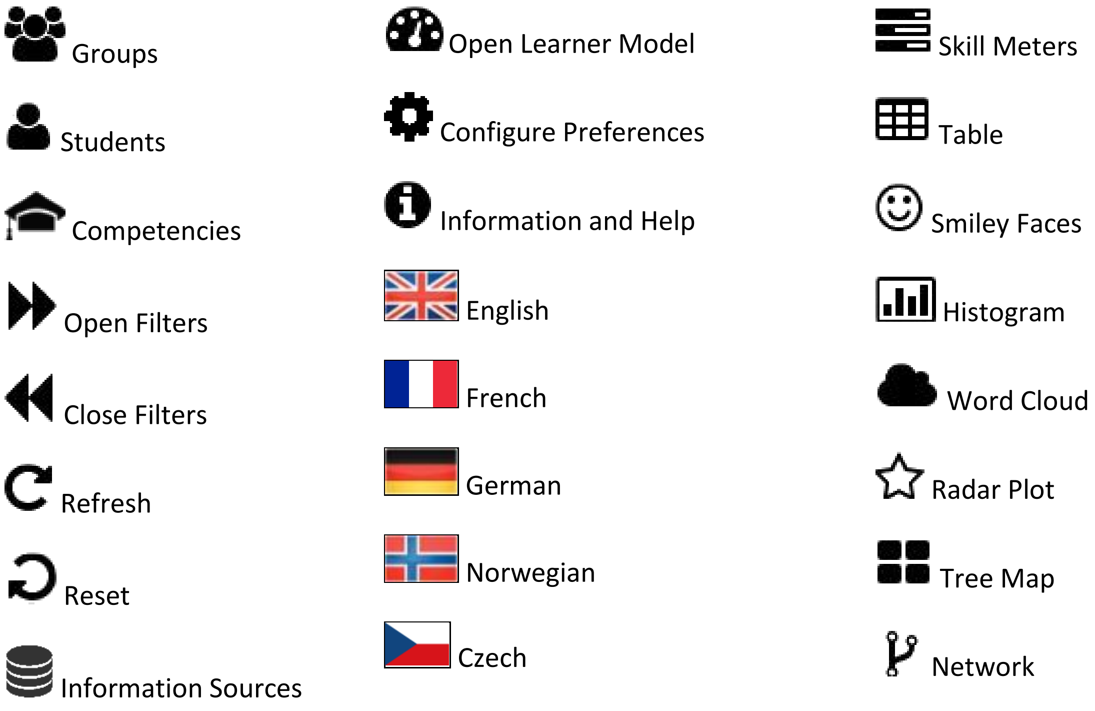


Figure 4: iconography.

The interface is localised into five languages: English, French, German, Czech and Norwegian. (Turkish not required at this stage.) The Czech language translation has been added for the purposes of the project.

2.3 Visual Methods

The visualisation set initially used in the visualisation services of LEA's BOX is the same as for the Next-TELL project. Initial engagement with teachers as part of WP5 work indicated that tables and radar plots were the most useful and actionable amongst the imported visualisation set, however this was with static screens and non-contextual use. All visualisations are retained as potentially appropriate analytics, building on the evaluation of them that has taken place in the Next-TELL project, until there is sufficient evidence for edit or omission of visual methods from this suite. The addition of new 'state-of-the art' visualisations have potential to be included in the second release, building on the work of Year 2. There may also be scope to release new visualisations as they become available within Year 2, building on feedback from studies engaging with students and teachers following the M12 release of the portal. See D2.1 for more information.

The current set consists of visualisations that are graphical and textual, some which show structure, some which are interactive, and some that quantise the data, whilst others use a continuous scale. (Table 1.)

Table 1: Next-TELL visualisation set (Next-TELL, 2014).

Visualisation	Graphical	Textual	Quantised	Structure	Interactive
Skill Meter	✓			✓	
Table		✓	✓	✓	
Smiley Face	✓		✓	✓	
Word Cloud		✓			
Histogram	✓		✓		
Radar Plot	✓				
Treemap	✓			✓	✓
Network	✓		✓	✓	✓

2.3.1 Skill Meters

Student competency is represented using a bar with a continuous scale. The proportion of colour is analogous to the extent to which the student is competent in the area. Indentation is used to show hierarchical structure. (Figure 5.)

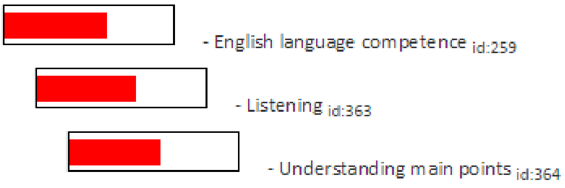
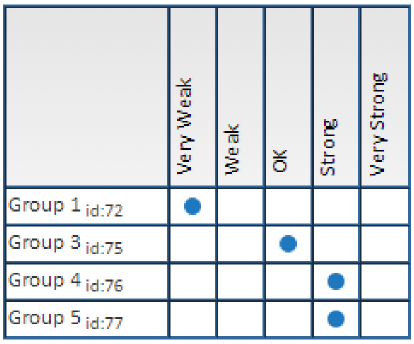


Figure 5: skill meter visualisation.

2.3.2 Table

Each element is a separate line in the table and hierarchical structure is shown using indentation. Competency is quantised into five categories, ranging from *very weak* to *very strong,* with a dot being placed in the appropriate table column to indicate this. (Figure 6.)

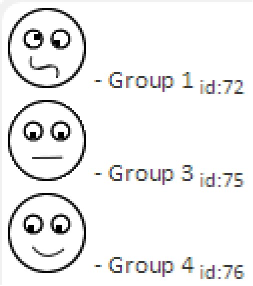


	Very Weak	Weak	OK	Strong	Very Strong
Group 1 id:72	●				
Group 3 id:75			●		
Group 4 id:76				●	
Group 5 id:77				●	

Figure 6: table visualisation.

2.3.3 Smiley Faces

The visual metaphor of a smiley face is used indicate competence. The scale is quantised into 5 images, from *confused* indicating little/no competence, through to *happy* indicating competence. Again, indentation is used to show hierarchical structure. (Figure 7.)

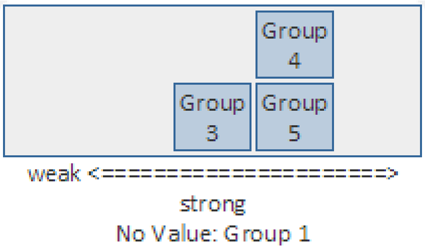


- Group 1 id:72
- Group 3 id:75
- Group 4 id:76

Figure 7: smiley face visualisation.

2.3.4 Histogram

Similar to the table view, competency is quantised into five categories ranging from *very weak* to *very strong*. Each item is a box on the histogram chart. Items with no data are omitted from the chart and are listed below. Whilst items are included in the same order as per a structured view, structure is not represented. (Figure 8.)



No Value: Group 1

Figure 8: histogram visualisation.

2.3.5 Word Cloud

Two complementary word clouds are included. The left hand one shows areas in which competence exists, whilst the right hand one shows area where there is no competency. Structure is not represented. The size of the word indicates the extent to which a competency is held or not held. (Figure 9.)

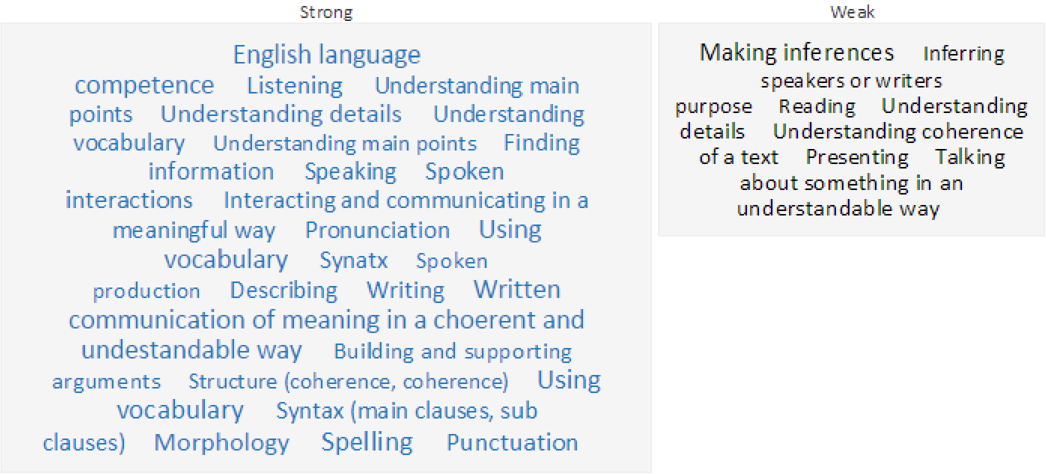


Figure 9: word cloud visualisation.

2.3.6 Radar Plot

Each axis displays a competency or data item. The further away from the centre the data point is, the greater the competency. Again, the structure of the information is not shown, however items are ordered clockwise. (Figure 10.)

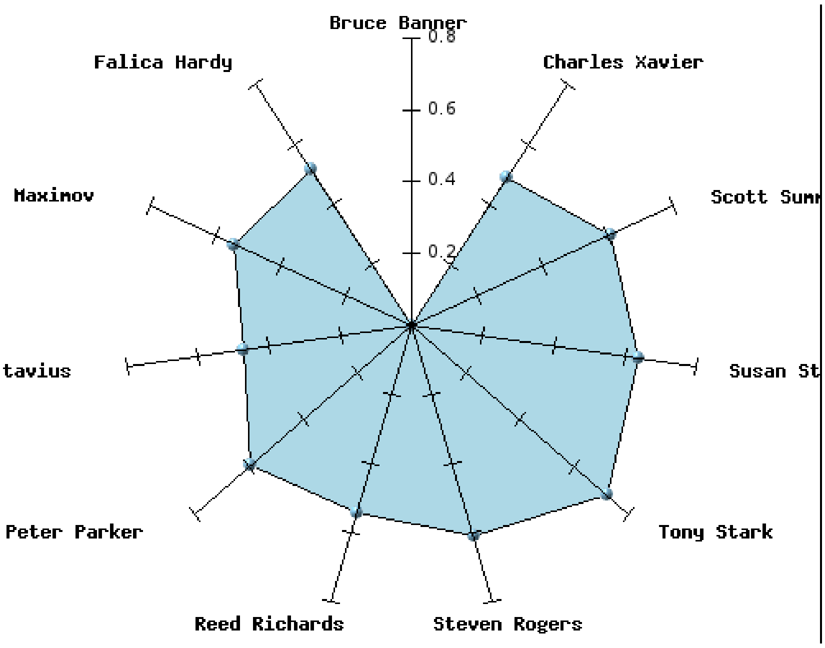


Figure 10: radar plot visualisation.

2.3.7 Treemap

Competence is indicated using the size of the rectangle on the treemap. Structure is represented in this visualisation using a series of layers. Clicking on a rectangle in the visualisation will show its sub-components. This visualisation is interactive. (Figure 11.)

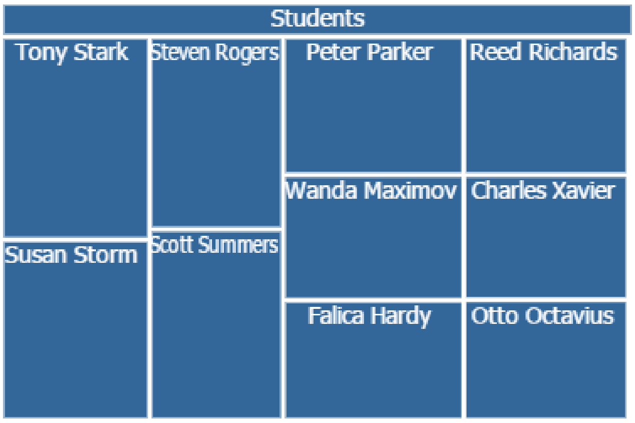


Figure 11: treemap visualisation.

2.3.8 Network

The network visualisation shows competency through the size of the nodes on the network. Nodes are quantised into 5 different sizes and shades of green; the larger the node, the greater the competence. Structure is shown by arcs between the nodes. The visualisation is a force-directed network and the nodes may be moved, and sub-nodes collapsed to increase readability. This visualisation is interactive. (Figure 12.)

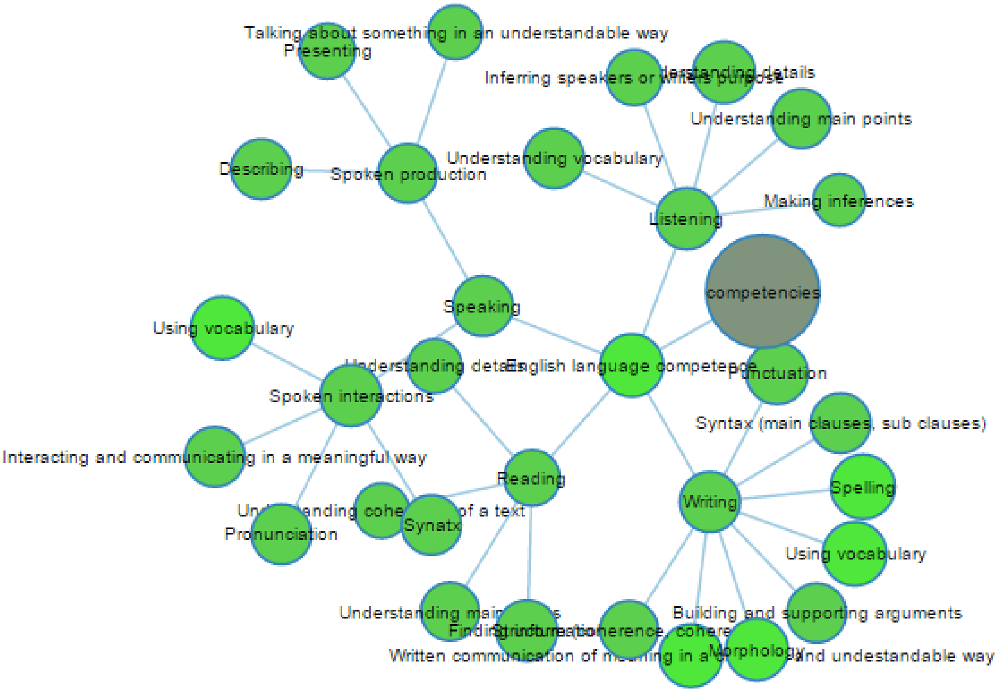

Talking about something in an understandable way
Presenting
Inferring speakers or writers purpose
Understanding details
Understanding main points
Describing
Spoken production
Understanding vocabulary
Making inferences
Listening
competencies
Speaking
Using vocabulary
Understanding details
English language competence
Punctuation
Spoken interactions
Syntax (main clauses, sub clauses)
Interacting and communicating in a meaningful way
Reading
Spelling
Understanding coherence of a text
Synatx
Writing
Pronunciation
Using vocabulary
Understanding main points
Building and supporting arguments
Finding information
Structure (coherence, coherence
Morphology
Written communication of meaning in a clear and undestandable way

Figure 12: network visualisation.

3 Re-purposing and Extension of the Next-TELL OLM

The final release of the Next-TELL OLM is detailed in Next-TELL D4.7 Student Model Tools Release 4 (Next-TELL, 2014). The code and functionality description is imported from this deliverable and release. In this section we summarise the amendments made to this software to enable it to provide a base for the visualisation services of LEA's BOX, key algorithms of relevance, and additional pieces of software development required for integration.

3.1 Functionality Summary

Additional configuration options have been added to the software's setup file to allow for customisation of features available to LEA's BOX. For the first release we aim to build a core visualisation service that concentrates on key elements of opening the learner model in our learning analytics context. We have therefore allowed for non-essential features to be turned off until initial engagement with teachers and students shows that they may be relevant to our line of research and overall product. High level customisations are as per Table 2.

Table 2: high level customisation of features from the Next-TELL OLM software.

Feature	Status	Rationale
Homepage	Disabled	Only the OLM facility is embedded into the portal. Links to its other facilities are managed from there.
OLM	Enabled	This is core to the visualisation service.
Show list of students in OLM	Enabled	This is core to the visualisation service.
Show list of groups in OLM	Enabled	This is core to the visualisation service.
Show competency structure in OLM	Enabled	This is core to the visualisation service.
Show list of data sources in OLM	Enabled	This is core to the visualisation service. This has been added.
Show list of activities in OLM	Disabled	This has been removed for now to allow a greater emphasis on the evolution of competencies from different sources of information.
Show notifications and updates	Disabled	Removed to allow focus on the evaluation of learner model visualisations in early studies of Year 2. To be re-added when appropriate according to the evaluation strategy.
Allow users to input assessments	Enabled	Evidence will constitute competency-based inferences output from the central executive or otherwise imported via the portal. All information is added through the API in accordance with the architecture outlined in D2.1.
Show learner modelling description	Disabled	Removed to allow focus on the evaluation of learner model visualisations in early studies of Year 2. To be re-added when appropriate according to the evaluation strategy.
Show evidence list	Disabled	Removed to allow focus on the evaluation of learner model visualisations in early studies of Year 2. To be re-added when appropriate according to the evaluation strategy.
Allow learner model discussion	Disabled	This has been removed for now to allow a greater emphasis on the evolution of competencies from

		different sources of information.
Localisation of language	Enabled	Important issue for technology acceptance.
User authentication and account management	Disabled	This is now handled by the portal.
Students able to release their learner models to each other	Disabled	This has been removed for now to allow a greater emphasis on the evolution of competencies from different sources of information.

3.2 Overview of Software Modules

The structure of the active high level algorithmic and repository components of the software is now as shown in Figure 13. Evidence and configuration settings are sent to the software via the API (left of figure). The system uses the configuration logic to contextualise and denormalise the evidence and prepare it for the active learner modelling algorithm that is employed. The learner model may be opened from different perspectives (groups, students, competencies, datasources) and in a range of visual forms (skill meter, table, smiley faces, histogram, word cloud, radar plot, treemap, network) (right of Figure 13). For each visual method the underlying open learner model data representation is the same. Through the addition of filter parameters, the scope of the information that contributes to the learner model may be amended to open different parts of it (namely data sources, groups, students and competencies may be specified).

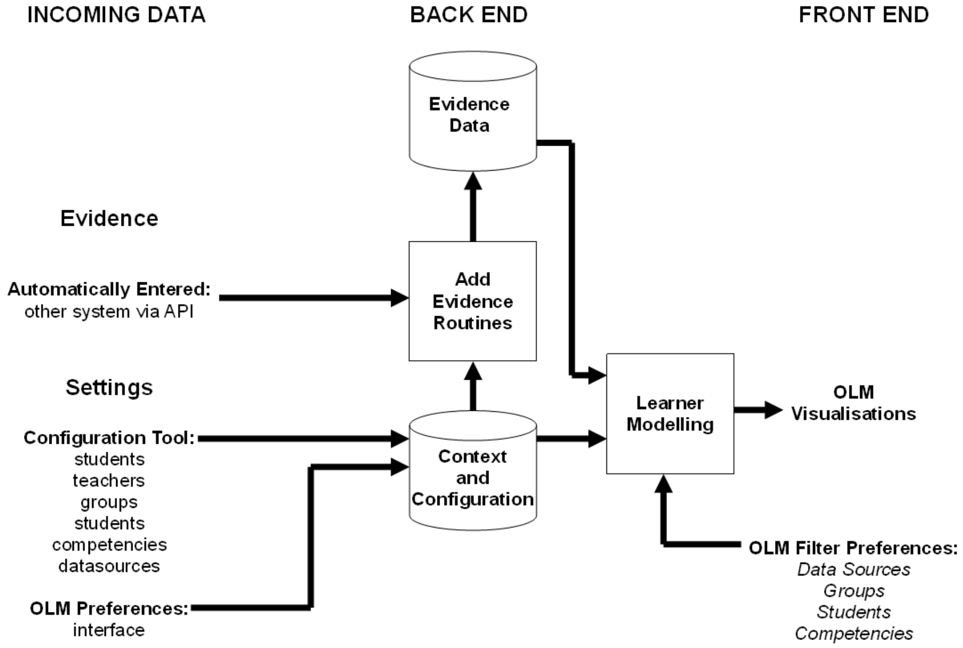


Figure 13: active components of the visualisation service software (open learner model).

The technical implementation of the Next-TELL learner model was done as a series of layers (shown in Figure 14). Requests to open the learner model require the physical transport of data through each of these layers. In this section we provide commentary on the repurposing of the Next-TELL OLM for LEA's BOX use in terms of each of these imported entities (Sections 3.3 to 3.7).

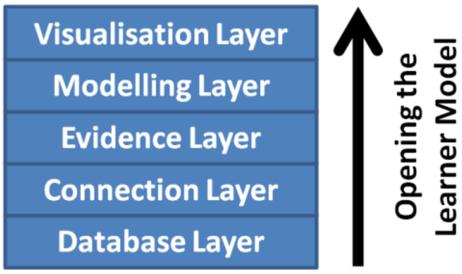


Figure 14: opening the learner model for visualization, reproduced from (Next-TELL, 2014).

3.3 Visualisation Layer

The imported visualisation layer from Next-TELL contains two highly interwoven components: firstly the OLM browser interface, and secondly the methods and tools that are used to render the visualisations.

The first aspect of this has been retained for use in LEA's BOX. This meets the initial requirements for a mechanism/platform for displaying learning analytics that are multi-dimensional; that is to say, where the data is from many different sources, students, groups and competency frameworks. This existing solution employs a filter mechanism to allow the content to be both browsed and searched by adding filter criteria. This is a suitable solution for the initial work of LEA's BOX as it collates the open learner model based learning analytics information together as one central configurable resource. This resource may also be extended to provide different sorts of learning analytics at a later stage, as required. Several amendments are made to the filters (Table 3), in particular, to tailor and simplify this for our current aims, consistent with the amendments outlined in Table 2, above.

Table 3: changes to filters.

Component	Status	Rationale
Information Sources	Disabled	Information now only comes from the API. There is no information to filter from student-, teacher-, or peer-assessments.
Tags	Disabled	Removed to allow a more competency-based focus.
Students	Retained	Key to the aims of the visualisation service.
Groups	Retained	Key to the aims of the visualisation service.
Competencies	Retained	Key to the aims of the visualisation service.
Subjects, Units, Activities	Disabled	Removed to allow a more competency-based focus.
Data sources	Added	Key to the aims of the visualisation service.

However, with regard to the second aspect of this, the methods and tools for visualisation themselves to a lesser extend meet the current needs of LEA's BOX. The visual methods of Next-TELL were fixed, and also specific code was written for the rendering of each. This is restrictive and an added layer of complexity. Within the architecture of LEA's BOX we have abstracted this to a separate reusable service that can be used to render not only open learner model information, but also further learning analytics that are to be added during Years 2 and 3 (see Section 4.1). As a starting point for this service, the same visual methods are retained until their suitability is evaluated sufficiently. The abstracting of this service means that visualisations can be easily upgraded and added to without needing to amend the core of the visualisation service.

The modularity of the visualisation service also allows it to serve API functions for embedding the results of learner modelling (or other algorithmic analytics) in to other components of LEA's BOX, without the need to use OLM browser. This increases the flexibility by which the learner model may be opened and by which learning analytics can be presented, with the addition of minimal extra code and logic.

3.4 Modelling Layer

For the first release of the LEA's BOX visualisation services, we focus on the open learner modelling aspect of technology. The learner modelling process from the Next-TELL project is used as a basis for this. It is suitably flexible that the changes to the front end service do not mandate that the existing code be changed. To cater for the need to emphasise the multi-datasource nature of the analytics in LEA's BOX, the learner modelling process is extended to allow the learner model to be opened for specific datasources. Thus in Figure 15 if we consider the cube as the content of the evidence layer (competency-based inferences), then the different faces of the cube are the different perspectives from which it may be modelled. In this release we focus on groups/students, competency frameworks, and datasources.

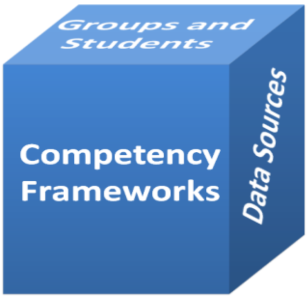


Figure 15: opening the learner model from different perspectives

The full algorithm is included as Appendix 1. In summary the stages of the algorithm are:

- Organise evidence according to modelling type (e.g. group, datasource, student).
- For each item to be modelled, further split evidence by student then competency
- Order evidence by age
- Apply basic depreciation algorithm to calculate level of influence for each item of evidence
- Apply further weightings and influence factors to refine the level of influence (LEA's BOX interface for this stage not yet implemented.)
- Normalise the levels of influence so that they total 1.0
- Multiply the level of influence by the value for the evidence, and total these
- Combine the value for each competency in the framework, taking into account if the competency has sub-competencies.
- The modelled value for each item is the root of the competency framework

3.5 Evidence Layer

Each competency-based inference is considered an item of evidence. It must have a numerical value in order to contribute to the modelling process. These are normalised as being in the 0.0 (no competence) to 1.0 (competence). The method by which evidence is selected for modelling is based upon the parameters/filter criteria that are specified. The attributes stored for each item of evidence are a subset of the Next-TELL OLM schema (Table 4):

Table 4: database table "Search_KnowledgeLevelRaw".

Field	Data Type	Description
RawID	Integer	Identification number of the evidence entry
Time	TimeStamp	Time of the evidence entry
StudID	Integer	Identification number of the student
Forename	VarChar	Forename of the student
Surname	VarChar	Surname of the student
EvidenceSourceID	Integer	Identification number of the evidence source
EvidenceSourceName	VarChar	Name of the evidence source
ContributorID	Integer	Identification number of the user who added the information (teacher, student or peer)
ContributorForename	VarChar	Forename of the user who added the information
ContributorSurname	VarChar	Surname of the user who added the information
ContributorType	VarChar	Type of user who added the information (student, teacher, peer etc.)
ClassID	integer	Identification number of the class group
ClassName	VarChar	Name of the class group
CompetencyID	Integer	Identification number of the competency
CompetencyPath	VarChar	Name of the competency's super-competency (recursive to the root node)
CompetencyName	VarChar	Name of the competency
CompetencyInfluence	Double	Weighting that specifies the competency's relative influence to other competencies, as part of the modelling process (Value between 0.0 and 10.0)
Depreciation	Double	Depreciation factor that specified the level of influence that new information has, as part of the modelling process
KLCorr	Double	The extent to which the competency is held (value between 0.0 and 1.0)
Approved	Integer	Whether the piece of information should contribute to the modelling process

The following schema attributes are unused:

- {teacherid, teacherforename, teachersurname} co-teaching is permitted, so this is derived algorithmically;
- {activityid, activityname, activitytemplateid, subjectid, subjectname, unitid, unitname, unitinfluence, activityinfluence} a more competency and datasource based approach has meant that a curriculum based approach is less relevant in this release of the system;
- {artefact,tags} not currently implemented;
- {teacherread,studentread} notification are not currently used;
- {KLProb,KLMisc} the system does not currently deal with misconceptions.

3.6 Connection layer

The connection layer is used to make a physical connection between the system logic and the persistent information residing in the database layer. This entity is imported unchanged from the

Next-TELL OLM software and contains a series of classes and packages for accessing and communicating with the database layer. The protocol used is JDBC.

3.7 Database Layer

Underlying data structure is now largely maintained by the portal and the portal's configuration tool. The tables updated are simplified for the current instantiation. The algorithmic components of the method used for opening the learner model calculate some of these relationships and definitions dynamically, based on the information in the data repository (e.g. search_knowledgelevelraw, which is denormalised). This allows for a much more flexible use of the open learner model within the scope of LEA's BOX. The database engine is retained as Apache Derby and all database content is password protected for local access only. The database tables currently in use for the first deployment comprise (Table 5):

Table 5: database tables and their function.

Table Name	Description	Type	Key Externally Maintained
COMPETENCIES	Information about competencies	Informational	Yes
EVIDENCESOURCE	Information about data sources	Informational	Yes
LANGUAGE	Definition of all phrases and words displayed on the interface in each supported language	Informational	No
LOGDATA	Information recorded anonymously about usage	Informational	No
NEXTAVAILABLEID	Next available ID numbers for key fields in database tables, where these are not externally maintained	Informational	N/A
SEARCH_KNOWLEDGELEVELRAW	Evidence for the learner modelling algorithm. This is a denormalised table and contains duplicate data from many of the other tables categorised as "informational"	Informational	No
SESSION	A record of each login by each user	Informational	No
STUDENT	Information about users. e.g. usertype.	Informational	Yes
STUDENTCLASS	Specifies the students linked to each class/group	Relational	N/A
VISUALISATIONPREFERENCES	Which visualisations users wish to include in their open learner model	Informational	No

Database tables available, but not currently in use include: ACTIVITY, ACTIVITYCLASS, ACTIVITYTEMPLATE, CASID_USERID, CHAT, CHATREAD, CLASSGROUP, COLOURDEFINITION, COURSEMATERIALS, GENERALPREFERENCES, SEARCH_TEXTUALRAW,STUDENTACTIVITY, STUDENTSUBJECT, STUDENTTEACHER, STUDENTUNIT, SUBJECT, TEACHERCOMPETENCY, UNIT, UNITCLASS.

3.8 Authentication and Authorisation

The user interface to the authentication mechanism is handled by the portal. The visualisation services of WP4 are an embedded component within this and so authentication information is passed on through the use of API events (Table 6). Whilst not authenticated, a notification message is displayed in the browser, upon page access.

Table 6: authentication API functions.

Function	Function Name	Parameters	Description
Log In	login	sharedsecret leasid password	access password id number of the user user's password
Log Out	logout	sharedsecret	access password

The authorisation policies of Next-TELL are carried through to the LEA's BOX visualisation service. Namely, a teacher may see all information related to students that they are in the same group as. Students may see only information related to their own learning. Whilst the Next-TELL instance of this software had a facility to allow students to release their learner models to each other, this facility is currently not enabled, to allow greater focus on the core of the learning analytics aims, within the first release. Log in and log out must be done under the same user session as the browser-based access. All other back end API functions may be done by external HTTP sessions.

3.9 Revised API

A lightweight API has been implemented for the purposes of LEA's BOX integration work. Initially it comprises the following functions, and will be continuously developed throughout the course of the project's integration work. These are designed to take ID numbers as parameters, so that the portal (and in particular the configuration tool) is the only entity that needs to define them. The ID numbers are thus the same across systems based at TUGraz and UoB. For this reason use of the existing Next-TELL API interface to the OLM is depreciated, although any functions needed can be repurposed. The same function call will complete both the *add* and *update* actions. If an entity does not exist, then it will be automatically created. This is a simplification of the existing protocol. The list of API functions available is as per Table 7. Delete functions require an override if the action will lead to a loss of student data. All API functions return suitable error and feedback messages. A full definition of functions is included as Appendix 2.

Table 7: API functions.

Function	Mode	Function Name	Parameters
Update User	add, update	updateuser	sharedsecret, leasid, username, forename, surname, password, school, type
Update Group	add, update	updategroup	sharedsecret, leasid, groupid, groupname, position
Add User to	add	addusertogroup	sharedsecret, leasid, groupid, userid

Group			
Update Competency	add, update	updatecompetency	sharedsecret, leasid, competencyid, competencyname, position, competencyparentid
Update Datasource	add, update	updatedatasource	sharedsecret, leasid, datasourceid, datasourcename
Add Information	add	addinformation	sharedsecret, leasid, competencyid, groupid, userid, datasourceid, value
Delete User From Group	delete	deleteuserfromgroup	sharedsecret, leasid, groupid, userid, (override)
Delete Group	delete	deletegroup	leasid, groupid, (override)
Delete Competency	delete	deletecompetency	leasid, competencyid, (override)
Delete Datasource	delete	deletedatasource	leasid, datasourceid, (override)
Delete User	delete	deleteuser	leasid, userid, (override)

3.10 Use of the API and Configuration

A simplified implementation of the API has allowed for the number of setup stages to be reduced prior to evidence being added. More specifically the new protocol just requires that entities exist (user, group, competency, datasource) and that the student is a member of the group. This does however mean that until there is evidence in the model for a specific competency or datasource, then it will not appear in the open learner model visualisation; the open learner model shows information for items relevant to learning using an evidence based approach. The recommended sequence of API calls is (Table 8):

Table 8: simplified configuration.

Action	Requires	Outputs from Action
add teacher(s), add students		userid (teacher), userid (student)
create group		groupid
add teacher(s) and students to group	groupid, userid (teacher) or userid (student)	
add competency		competencyid
add datasource		datasourceid
add competency inference	groupid, userid (student), competencyid, datasourceid, <user to be in the group>	

3.11 Interaction Logging Protocol

This is retained from the Next-TELL project and imported. It records all key informational and interaction events for the purpose of software evaluation. All data is stored against the user ID number and no data is recorded to directly link the information with any specific individual without first knowing the number. No names or identity information is stored. For each entry the following information is persisted:

- ID number for the log event
- Timestamp

- User currently logged in that instigated the event
- User type (e.g. student, teacher)
- Which {teacher, group, student, competency} is affected by the event
- The type of event (e.g. configuration, read, update)
- Number of the event and name of the type of event
- Up to 5 custom data fields (depending on the event)
- IP address of the device
- Longitude, latitude and location of the point of access
- Session ID of the user's usage
- Group identifier for the user (ie the associated school)
- Ethics approval flag, to state whether the data captured can be used anonymously for software evaluation and research purposes.

4 Additional Components and Development Work

In addition to the reconditioning and extension of the Next-TELL OLM as the basis of the visualisation service, a modular component is implemented to render datasets in a visual form. This service is integrated with the visualisation service described in Section 3.3, but is also available for use in other project software components. It has its own API and data format. (Section 4.1.) Furthermore, issues related to integration, such as cross-origin communication and server configuration have been required to be addressed to enable effective integration between the WP4 services and the portal. (Section 4.2.)

4.1 Rendering Datasets in a Visual Form

Communication between any software packages with LEA's Box and the Visualisation Service are done via HTTP requests. The service is built to accept either HTTP GET or POST requests though for large volumes of data POST requests are preferable. The method of sending the request is as follows (Figure 16 and Table 9):

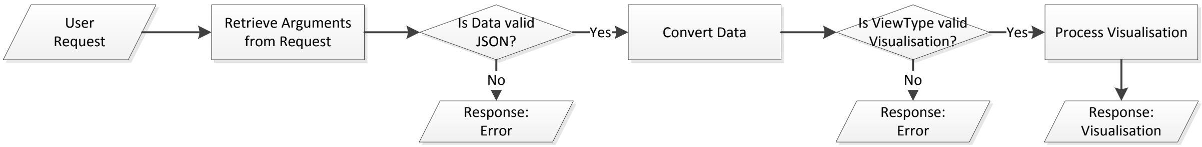


Figure 16: data flow diagram of visualisation rendering process.

Table 9: visualisation service request breakdown.

Resource	Type	Example
Destination of Visualisation Service	URL	http://.../VisualisationService/service.php
ViewType	String	'skillmeter' or 'network'
Data	JSON object	"obj0": {"name":"John Smith", "value":"0.674532341", "title":"0.674532341", "id":"1001", "parent":"1000" "source":"students"}
Dimensions	String	'500'

As described in Figure 16, upon receipt of the data, the Visualisation Service will break down the different elements of the JSON object and use them for each visualisation output differently. If an invalid JSON object is supplied as the Data argument of the request, the Visualisation Service will send back "nothing to display." As per the example shown in Table 9, each element of the JSON object is used in similar ways in the visualisation process but not all of them are used in each process:

- The name element is used as a label to identify what the visualisation represents.
- The value element is used as the defining factor of the visualisation output.
- The title element is used as the alt text for the visualisations that implement this.
- The id element is the unique identification of the dataset.
- The parent element is used to identify super-components within the data structure – used for rendering hierarchical relationships.
- The source element is used as an identifier for the data collection, where multiple entities are combined within the same data set.

After validation of the Data, the service will then determine the viewtype desired as the output. The available visualisation types are as per Table 10. Visual examples of each of these visualisation types are described in Section 2.3.

Table 10: visualisation types.

Name	Argument
Skill Meters	skillmeter
Tables	table
Smiley Faces	smilie
Word Clouds	wordcloud
Radar Plots	radar
Histograms	histogram
Network Map	network
Tree Map	tree

If the viewtype specified isn't a valid selection from Table 10 then the service will report "no view type selected."

The Dimensions argument is used where the visual method requires the size parameters of the screen real estate, into which it will be embedded. This is used for the rendering of the Network Map, Tree Map and Radar Plot visualisations. The D3 and JPGraph libraries used to render these visualisations require specified dimensions to create the output. The Dimensions argument is used to define both the width and the height of the visualisations. (Sections 4.1.1 and 4.1.2 cover issues related to these visualisation libraries in greater detail.)

The Visualisation Service creates a HTML rendered output that is sent back to the requesting host. The only visualisation that requires special consideration is the Radar Plot as this visualisation needs to be rendered and saved to the local disk of the hosting server and then is referenced by the HTML.

4.1.1 JPGraph

JPGraph is a PHP graphing library used to create the Skill Meter and Radar Plot visualisations. In the Next-TELL OLM the Skill Meter and Radar Plot visualisations were created by building the image from scratch and using the D3 graphics library respectively. For ease of development, to reduce overhead on the client side and to increase efficiency in producing the results, the JPGraph library was used to render these visualisations.

Each Skill Meter is created independently of each other using a PHP script as the reference source of a HTML image tag. The Skill Meter receives its criteria via a HTTP GET request and processes the information to generate the output.

The Radar Plot is generated from the dataset provided in the initial request. Upon rendering the visualisation it is then saved to the local disk under the ./radar_temp/ directory. These files are named with the prefix 'temp' followed by the time to the micro second and is saved in the PNG image format. Each time a radar plot is called the precursor process to saving the render is to go through the ./radar_temp/ directory and determine how many of the png files are over 60 seconds old, then delete them.

4.1.2 Data Driven Documents

As it was described in D2.1 Section 3.3.4, the visualisation components of the Next-TELL OLM will be brought forward into the LEA's Box OLM. This required the implementation of the interactive visualisations for the Network Map and the Tree Map. This would require the use of a client side library that would allow for complex manipulation of the visualisations. It was seen that the best method for this was to use the Data Driven Documents (D3) JavaScript library.

4.1.3 Prerequisite Requirements

In order for any and all software packages to get full use of the Visualisation Service a sustained internet connection must be provided.

4.1.4 Data Persistence

Development of the Visualisation Rendering Service will be designed in such a way that no data will be held by the service. Information is dynamically generated and returned in response to the request of a client. Only those parts of the Visualisation Service related to learner modelling persist the information in a pseudo-anonymous form (by allocated ID number). See Section 3, above.

4.2 Integration Support / Cross-origin resource sharing (CORS)

One further development issue of note is that of cross-origin resource sharing (CORS). This has been a key aspect of integration work between our services. This is a policy recently enforced in January 2015 by Webkit-based web browsers such as Chrome and Firefox and has mandated development work to be compliant with this.

CORS was established by the World Wide Web Consortium (W3C) as a security measure for further developing internet based technologies. CORS is a preventative measure to secure content across domains from people trying to access it for malicious purposes. For standard HTML, CORS isn't strictly needed. When JavaScript is embedded into a HTML web page the JavaScript can be designed to take content from other websites and pretend to be the destination a user believes they're going to, send sensitive data from client to server (much like that of a keyboard logger) or user's own web browser to maliciously attack a server to bring it down from a Denial of Service attack without the user even being aware that this is happening. This is when the application of CORS is advised.

CORS automatically restricts JavaScript access of other websites when attempting to connect to one domain from another. Because of CORS being a forced feature of newer web browsers, LEA's OLM or other software packages will not be granted access to the Visualisation Service by default. This can be circumvented by the acceptance of difference sources via the header settings on the server.

For development purposes, the Visualisation Service will accept requests from all domains but before the third system release it's advised that it becomes more restricted to domains that are explicitly known to require access.

5 Summary

This deliverable has described the first release of visualisation services for use within the LEA's BOX project. Specifically we have presented a description of the graphical user interface to the tool and also the application programming interface. We have described the reconditioning of an existing tool brought forward from a previous project (Next-TELL), in alignment with the plan outlined in the description of work. Furthermore we have detailed extensions that have been made to make this an appropriate platform/baseline for data visualisation services in LEA's BOX, with specific reference to open learner modelling.

6 References

[McCalla, 2004] McCalla, G. (2004). The Ecological Approach to the Design of ELearning Environments: Purpose-Based Capture and Use of Information about Learners, Journal of Interactive Media in Education, Vol. 7.

[Next-Tell, 2014] Next-Tell (2014). Student Model Tools Code Release and Documentation: Explanatory Notes. [Online] Available from: http://next-tell.eu/wp-content/uploads/NEXT-TELL-D4.7-BHAM-Student_Model_Tools_R4_v06.pdf [Accessed: 12th February 2015].

7 Appendix 1: Next-TELL OLM Algorithm

The following algorithm is reproduced from Next-TELL deliverable D4.7 (Next-TELL, 2014). We have included this for the sake of completeness as it is of key relevance to data/modelling layer of the visualisation service.

The Next-TELL learner modelling process is competency-based and executes over the raw data (individual inferences) at the point at which the learner model is required to be opened. This allows the OLM to take a more ecological approach (McCalla, 2004) and flexibly visualise different combinations of the underlying data. The learner model may be calculated for any combination of students, groups, competencies, or activities, from any custom combination of the raw data. The following algorithm is applied:

1. Retrieve all relevant items of evidence, according to the sub-set of inferences to be modelled. Identify how the visualisation is to break down the information (e.g. list of students, groups, competencies or activities). Split the evidence items by student, and then by competency.
2. For each sub-set of information (split by student, and then by competency) order the evidence newest to old- est. Determine influence factor for each item of evidence by applying the following formula iteratively over each dataset item. This provides a basic weighting for each item of evidence, with the most recent item receiving a greater influence weighting. b = basic influence (initially 1.0) and d = basic depreciation factor (default 0.2).

$$b_i = b_i\text{-}1\ (1.0 - d_i) \qquad (1)$$

3. Once each data item has its basic depreciation factor (b), the level of influence (i) the item has in the learner model is further influenced by parameters that the teacher has control over, according to the following calculation. f = teacher specified depreciation factor, affecting level of influence of new information; c = level of influence that the competency has; a = level of influence the activity has, with which the item of evidence is associated.

$$i = b\ (0.5 + f/2)\ (0.5 + c/2)\ (0.5 + a/2) \qquad (2)$$

4. For each sub-set of information, the levels of influence (i) are normalised so that they total 1.0.
5. For each sub-set of information, the level of influence (i) and each value (v) are multiplied together and then summed, where v>=0.0 and v<=1.0.

$$\Sigma\ iv \qquad (3)$$

6. For each sub-set of data, the values (V) for competencies (C) and their sub-competencies (S) are combined recursively according to their structure, using the following formula, where D indicates the presence of data for the competency or sub-competency:

$$Vi\ = \begin{cases} 0.5V_C + 0.5V_S & if\ \ D_S \wedge D_C \\ V_C & if\ \ \neg D_S \wedge D_C \\ V_S & if\ \ D_S \wedge \neg D_C \\ 0.0 & if\ \ \neg D_S \wedge \neg D_C \end{cases} \qquad (4)$$

7. The value for each student (see point 1) is taken as the root node in the respective competency structure. The value for the item in the list (e.g. group, competency or activity – see point 1) is determined by taking an average of all students for whom there is data.