DELIVERABLE D 4.3 –

SECOND RELEASE OF VISUALISATION AND OLM SERVICES AND TOOLS

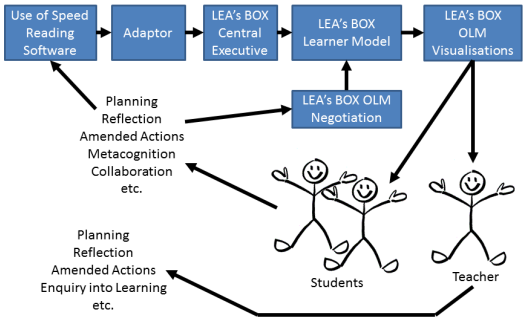
Document Version:	5
Document Status:	Final
Document Type:	Deliverable
Diss. Level:	Public
Lead Partner:	UoB
Delivery Date:	31st December 2015
Authors:	Matthew Johnson, Blandine Ginon, Drew Masci, Chris Baber, Susan Bull (UoB)

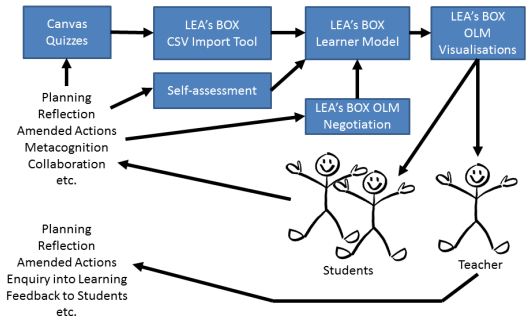
CONTENTS

1 Executive Summary

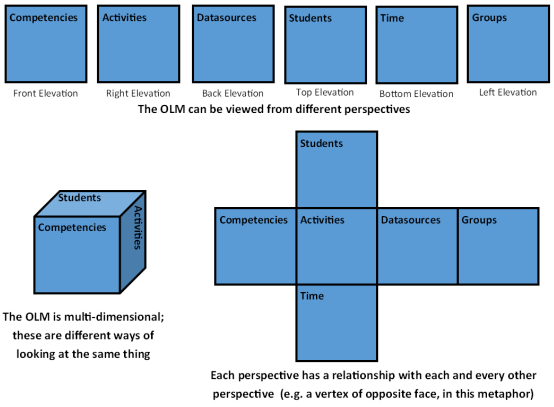
This report accompanies Release 2 of the LEA's BOX OLM. It reports on the status of development as at M22, gives an incremental update on D4.2, and considers technical development ahead, looking forward to Release 3.

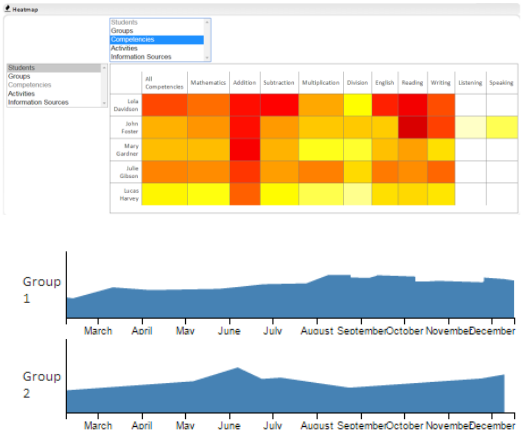
Development work is guided by real-world use cases. In Section 2 we report on these in terms of ongoing studies, and provide some clarification has to how the aims of OLMs sit with both general learning analytic frameworks, and also with ongoing evaluation.



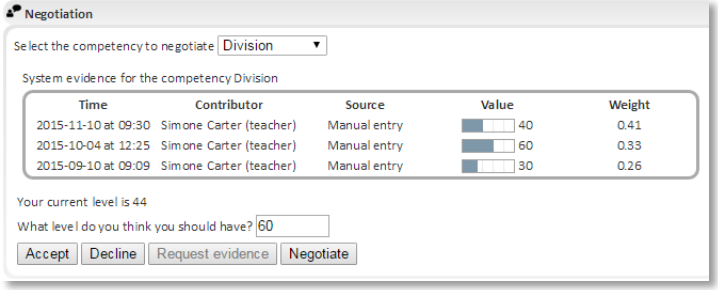


Section 3 gives an update for ongoing visualisation work, in terms of both multidimensional data visualisation, and also the potential of combining competency-based and non-competency based data type to produce hybrid analytics. We describe the attributes of existing visualisations in terms of established framework elements.

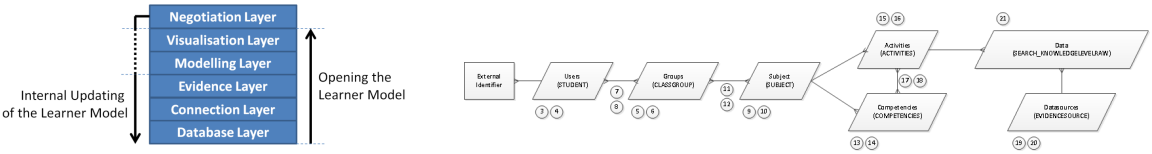




Section 4 reports on the implementation of facilities for dialogue-based learner model negotiation. Development work on this is in progress, and so a clarification is given between pure negotiation and persuasion, whilst implementation is finished.

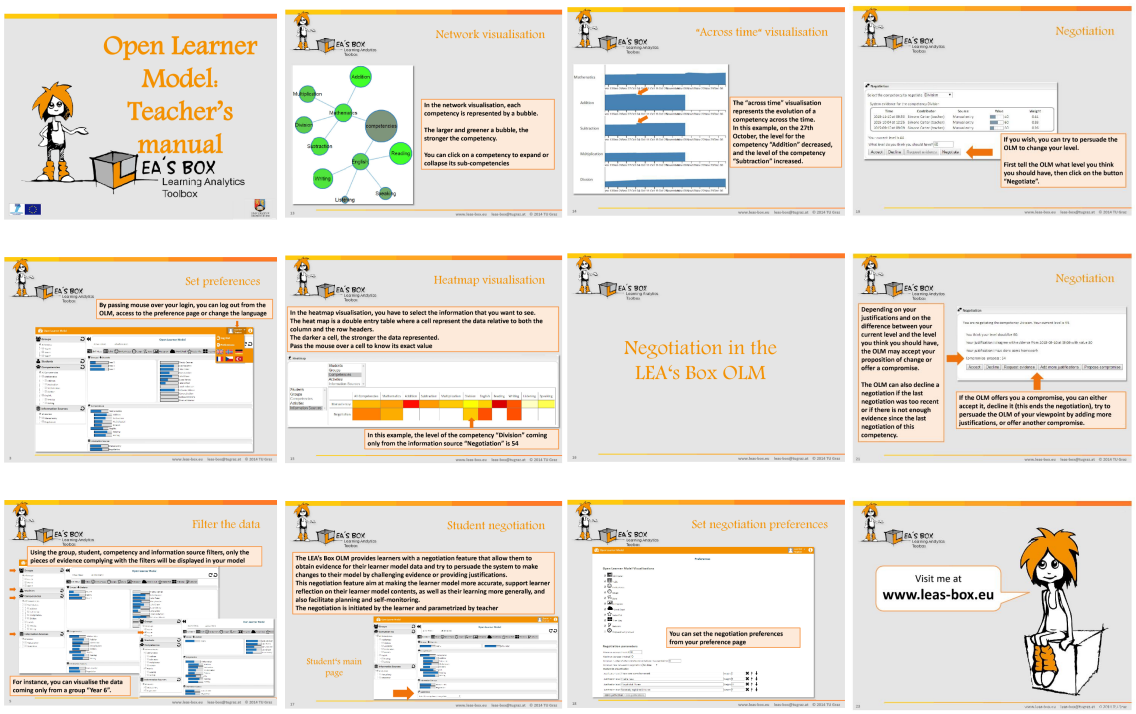


Section 5 gives updates to the technical specifications and inner workings of the OLM to facilitate the development work of Year 2, learner model negotiation, and a tightened level of integration between the OLM and the LEA's BOX portal. We also report on the extension of the visualisation service's facilities.



Section 6 concludes the main body of the report, detailing the methods by which competency inferences can be added to the learner model evidence layer. We present a tool implmented to read information from CSV files for updating the OLM evidence layer using output from content management systems, such as Canvas.

Appendix 1 contains the user manual for the OLM, and Appendix 2 contains slides from several presentations used to introduce the OLM to students and teachers. Appexdix 3 gives the full and revised API specification.



2 Introduction: Use Cases Driving Forward Implementation

Behind the technical development work of WP4 is the need to be guided by real world examples, focus groups and feedback from users themselves, in addition to the literature that underpins both open learner modelling and data visualisation in this context. A core part of this is the realisation that display of the learning-based analytics is not the end point for the information, but more the mid-point in a cyclic process; more feedforward, than feedback.

Table 1, reproduced from [Elias, 2011] compares 5 high-level models/frameworks of analytics, highlighting the key components present in each. As an illustration, if applied to the application of an open learner model, which provides a specific form of learning analytic that visualises inferences about student-based competency, visualisations are used to inform different stakeholders (students, teachers, etc.) in terms of the predicted actuality of student competency: information intelligently modelled, rather than solely reported, aggregated or combined. The presented information (shown as the stages highlighted in each framework/model) may feed forward into such episodes of planning, reflection, amended courses of action, metacognition, collaboration, issues of accuracy or indeed trust of the formative assessment information, all of which are core to the very reasons for opening the learner model to informational stakeholders [Bull and Kay, 2010].

Table 1: comparison of analytics frameworks and models, reproduced from [Elias, 2011].

Knowledge Continuum	Five Steps of Analytics	Web Analytics Objectives	Collective Applications Model	Processes of Learning Analytics
Data	Capture	Define Goals	Select	Select
		Measure	Capture	Capture
Information	Report		Aggregate	Aggregate and Report
Knowledge	**Predict**		Process	**Predict**
Wisdom	Act	Use	**Display**	Use
	Refine	-	-	Refine
-	-	Share	-	Share

The modelled inferences regarding student competency are predictions of what is known about the students; knowledge. Actions such as ‘act’, ‘refine’, ‘use’ and ‘share’ clearly follow. As an indication of how this relates to the use of an open learner model, actions at these post-analytic stages could comprise the following (See Table 2).

Table 2: mapping stages of analytic process to actions and reasons for opening the learner model.

Wisdom	Act	Refine	Use	Share
Learner autonomy Metacognition Trust Scientific inquiry	Planning Amended action Collaboration Scientific inquiry	Accuracy Trust	Metacognition Reflection Planning	Collaboration Planning

Within the evaluation studies of LEA's BOX (as further described in the deliverables of WP5), the OLM is being used as a tool that is at the mid-point of an analytics process, with actions such as those indicated in Table 2 forming a staple component of the latter part of the use cases. It is these use cases that are being used to drive forward development, and also these elements of the analytics process being used to guide the design of the evaluation, in terms of the OLM being used as a learning analytics tool. As an example, we illustrate this with two LEA's BOX evaluation use cases, where there is need for up to date competency analytics information, from frequent interaction with technology: Sections 2.1 and 2.2. Development priorities are thus identified from these in Section 2.3.

2.1 Speed Reading

As part of completion towards a certificate, students are required to make use of piece of software that is designed to facilitate improvements in their ability to read at speed. Students complete up to 20 activities, in any order, with any frequency of repetition of the course of a two week period. Students are encouraged to do this for half an hour each day. Whilst students and teachers are given a basic level of feedback about performance in these activities, this is not competency based. A competency framework of 50 items is defined, and, through the use of an adaptor, data is sent to the LEA's BOX system, interpreted by logic that is part of WP3. It is sent to the learner model as a series of competency-based inferences, each time new information is available. The OLM is thus an open representation of this competency framework information, and is available to be accessed at any point. Furthermore, students are able to negotiate their learner model to increase its accuracy. The open learner model is thus formative assessment that they may use alongside the main software. (See Figure 1.) This use case is summarised in the use case template of Figure 2.

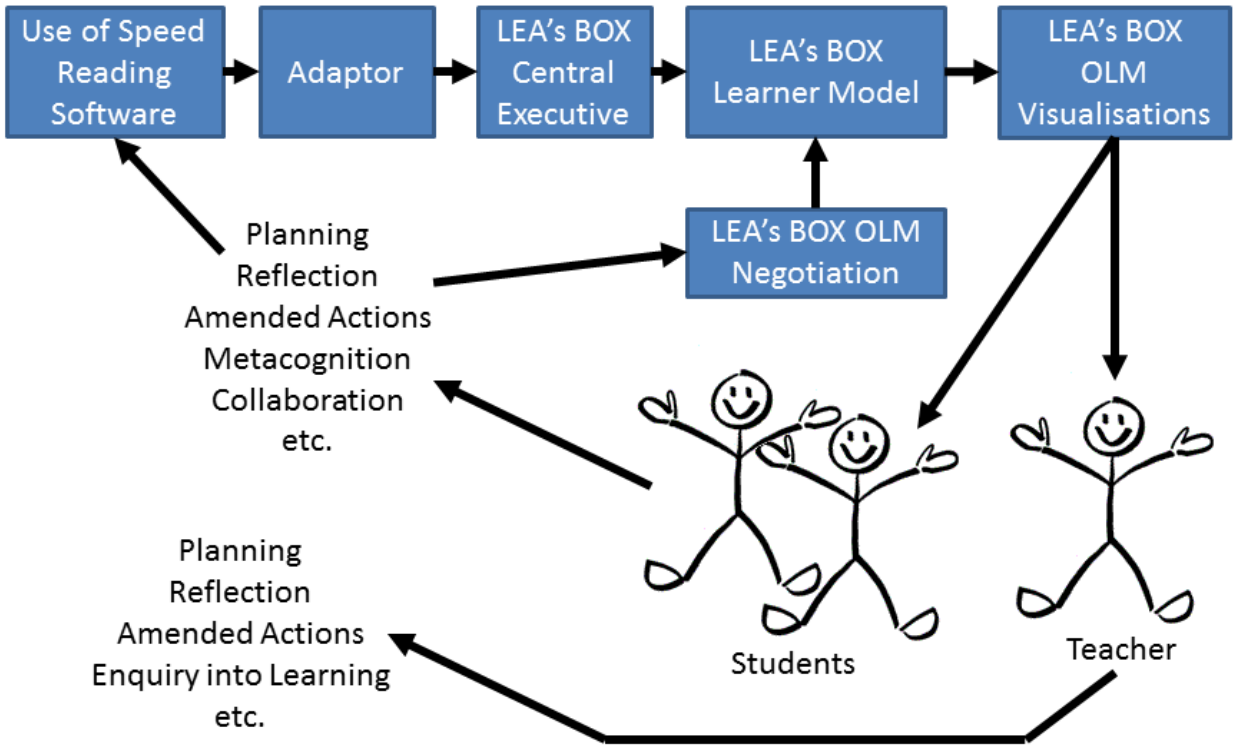


Figure 1: Speed Reading

Title: Speed Reading

Goal:

- Improved ability for students to read at speed following 30 minutes a day usage of a tool designed for this purpose.
- Analytics are available via LEA's BOX that give modelled predictions of student competency in a visual form, and that can be fed forward formatively into future action.

Initiating Stakeholder(s): Teacher(s)

Affected Stakeholder(s): Students, Teacher(s)

Start Conditions:

- Teacher has decided to teach speed reading, as is pedagogically appropriate to the current plan for their students.

End Result:

- Students have completed 2 weeks of speed reading, with use of the LEA's BOX OLM available throughout to give finer grained competency-based formative feedback.
- Students are assessed summatively at the end of the two week period and gain a certificate, if participation is consistent throughout the period of study.

Normal Course for the Use Case:

- Teacher wants to teach speed reading
- Teacher configures groups, students, subjects etc. in the LEA's BOX configuration tool. The competencies, activities and datasource are also added as part of the adaptor synchronising with the LEA's BOX system.
- Students are introduced to the speed reading application, and its activities. Students are also introduced to the LEA's BOX OLM, and its negotiation facilities.
- Students use the speed reading tool for half an hour each day (from anywhere and at anytime). Data is pushed automatically to LEA's BOX at the end of an activity or at the end of a learning session/use period (whichever comes first).
- Teacher and students may log in to the LEA's BOX portal to see how their OLM has updated, using a variety of visual methods. The multiple analyses may also show how the model has changed across time. Teachers can flexibly query any combination of the data presented. All may compare different aspects of the information and the relationships between them, e.g. between competencies and activities.
- Students may negotiate their learner model to increase its accuracy, adding additional evidence, if disagreement occurs.
- Students (and teachers) may use the information in the OLM to aid planning, selection of tasks, promote reflection, amend actions, consider metacognitive aspects of their learning based interaction, or as an initiation point for working collaboratively with other students.

Exceptions:

- Students do not complete the required half an hour a day interaction with the speed reading tool and so there may be an absence of information in the learner model.
- Students complete the same small selection of activities over and over again, and the learner model is thus only able to display information for a subset of competencies.
- Interaction with the open learner model is voluntary and students may not choose to engage with it.

Figure 2: Speed Reading: Use Case

2.2 Italian as a Second Language

During several concurrent language courses, the LEA's BOX OLM is made available alongside students' learning. In addition to other formative exercises, and self-regulated learning tasks, students are encouraged to complete quizzes on a content management system (Canvas) and the teacher may also assign marks to students. The Canvas quizzes give a score to the student, at the end of the quiz, as formative feedback. More detailed competency-based information may be imported into the LEA's BOX OLM from Canvas, and the teacher and students should also be able to complete (self-) assessments. This information is then added to the learner model and the visualisations may be accessed at any time by students and teachers alike. The learner model may also be negotiated by students, including with the consideration of additional evidence and other predefined justifications. This is summarised in Figure 3 and the use case in Figure 4.

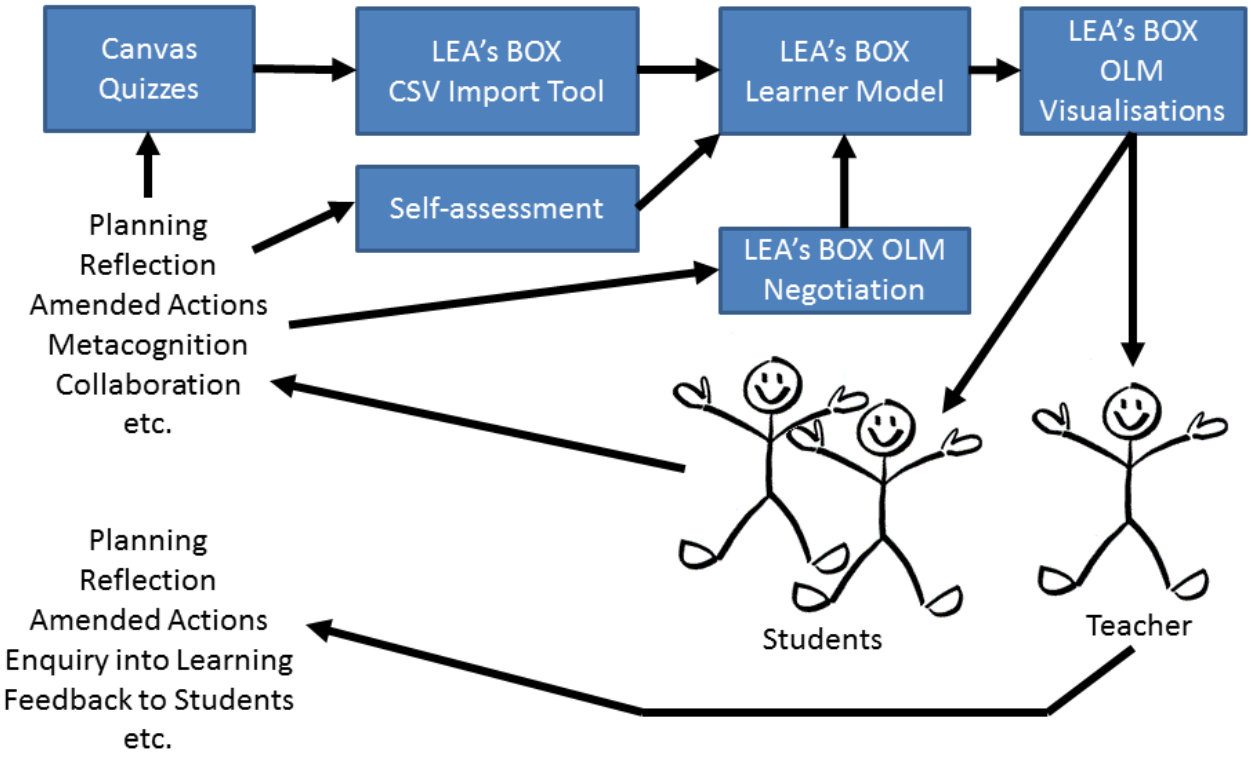


Figure 3: Italian as a Second Language

Title: Italian as a Second Language

Goal:

- To support learning in the context of an Italian language course by adding competency-based analytics to several hundred existing Canvas-based quizzes, with view to increase their utility, flexibility and to allow them to be used as regular homework exercises, rather than solely for exam preparation.
- To give students access to competency inferences about their understanding to help them comprehend their strengths and weaknesses, plan what to work on next and allow them the opportunity to create a stronger relationship with the competencies involved in their course.

Initiating Stakeholder(s): Teacher(s)

Affected Stakeholder(s): Students, Teacher(s)

Start Conditions:

- An Italian teacher has defined the competencies, activities and links between them
- An Italian teacher has defined the groups and students
- It is pedagogically appropriate for students to complete the Canvas quizzes and/or self-assessment.

End Result:

- Alongside their current learning, students have had access to modelled predictions of their competencies, and have had the opportunity to use this information in their planning, decision making and general learning.
- Students have had the opportunity of increase the accuracy of the model by using the learner model negotiation facilities
- Teachers have had access to individual and aggregated information about their students' competencies, progress, and levels of activity.

Normal Course for the Use Case:

- Students are introduced to Canvas quizzes and also to the LEA's BOX OLM.
- Students complete quizzes in Canvas, of their choosing and at times of their choosing.
- Every day the quiz results from Canvas are exported as a CSV file and then imported to the LEA's BOX system, via facilities in the teacher's portal.
- Teacher and students may log in to the LEA's BOX portal to see how their OLM has updated, using a variety of visual methods. The multiple analyses may also show how the model has changed across time. Teachers can flexibly query any combination of the data presented. All may compare different aspects of the information and the relationships between them, e.g. between competencies and activities.
- Students have the opportunity to complete self-assessments for each competency, and teachers may add additional information.
- Students may negotiate their learner model to increase its accuracy, adding additional evidence, if disagreement occurs.
- Students (and teachers) may use the information in the OLM to aid planning, selection of tasks, promote reflection, amend actions, consider metacognitive aspects of their learning based interaction, or as an initiation point for working collaboratively with other students.

Exceptions:

- Interaction with both the open learner model and Canvas is voluntary and students may not choose to engage with it.
- Students complete the same small selection of quizzes over and over again, and the learner model is thus only able to display information for a subset of competencies.

Figure 4: Italian as a Second Language: Use Case

2.3 Development Priorities for Use Cases

With evaluative use cases driving forward development, there comes several high priority areas of development and features that are new for Release 2. The remainder of this report is structured with these in mind. Features that are previously reported are retained; please refer to previous WP2 and WP4 deliverables for associated software specification documentation and previous release notes. Complete 'user manual' documentation is thus included as Appendix 1, in order to avoid much duplication. Requirements include:

- The display of competency information for specific activities, and ability to see how different activities have contributed to the model. (Sections 3, 5)
- The links between activities, competencies, students, subjects and groups being specified prior to usage and adding evidence. (Section 5)
- The display of how the learner model has changed over time. (Section 3.4)
- The display of information related to when information was added, and other more general aspects of learning analytics. (Section 3.4)
- The display of data in a multi-dimensional manner for the ease of comparison. (Section 3.3)
- Permitting comparison between custom combinations of competencies, students, groups, activities and data sources, where multiple items may be selected at any one time. (Section 3.2)
- There need for data to be easily or automatically added from existing tools, and from multiple data sources. (Section 6)
- Facilities to allow students to negotiate the content of their learner model, to increase its accuracy and strengthen their relationship with the information it contains. (Section 4)
- Further control for teachers regarding what data is permitted to enter the model, whether students can negotiate it, and what can be negotiated as is pedagogically appropriate at the time. (Section 4)
- The need for students and teachers to access a reliable and robust portal. (D2.5)
- Facilities for self-assessment. (D2.5)
- A clear and concise corpus of introductory materials for initiating students and teachers in the use of the LEA's BOX OLM. (Appendix 1, Appendix 2)

2.4 Materials and Resources

To support usage of the OLM within evaluation, a user manual is provided (Appendix 1), together with a summary presentation for introducing the tool (Appendix 2).

3 Data and Visualisations

Working forward from our use cases and a specification of the evidence layer from which the learner model is built (Section 3.1), we also consider the implementation of requirements for the open learner model to visualise multiple dimensions of the data at once (Section 3.3). We also report on how combining the visualisations with non-competency based analytics might be done, in order to increase the utility and flexibility of the information presented (Section 3.4). We conclude this section with a comparison of the properties of the current visualisation set (Section 3.5), and briefly consider how this might be extended effectively for Release 3, pending feedback from end users.

3.1 Attributes of the Underlying Evidence

The LEA's BOX learner model uses an active learner modelling approach; that is to say in this context that the learner modelling algorithm executes across the corpus of evidence, at the point at which a visual representation of the model is desired. The underlying evidence base now contains information about the following, for each inference that it holds (Table 3).

Table 3: attributes of each item of evidence.

Attribute(s)	Description
Time	Date and time at which the data was added
Student	Student to which the competency inference refers
Contributor	The identity of the person who caused the information to be added (e.g. may be student, teacher or peer)
Contributor type	Teacher, student etc.
Data/Evidence Source	Where the data has originated from (e.g. the name of the software tool)
Group/Class	As part of which group or class the student is working
Activity	What task the student is undertaking to cause the inference to occur
Subject	The wider subject, to which the activity and competency belong
Competency	Competency that the inference will update in the framework
Activity Influence	How important this activity is, compared to others (inference significance)
Competency Influence	How important this competency is, compared to others (inference significance)
Inference Value	The value of the competency inference in the range 0 (no competence) to 1 (competent)
Approved	Whether this data should contribute to the model (1) or whether it is obsolete, removed or superseded (0).

This permits for a wide range of learning-based analytics to be generated from this dataset. Those that are most relevant to displaying the *current* state of student competency see the learner model being opened from the perspectives of the *competency framework*, a list of *students*, a list of *groups/classes* and a list of *datasources. A*dditionally, the learner model may now be opened from the perspective of *activities* (and *subjects*) undertaken. This brings the OLM into alignment with other LEA's BOX tools, such as myClass, that contain a curriculum component, and also with the use cases and for planned studies. The display of activities is the largest architectural change made to the functionality of the system (in addition to the way in which configuration takes place – covered in Section 5.2).

3.2 Filtering the Data to be Visualised

In addition to the existing options to filter the data in the open learner model, amendments have been made to both the interface and learner modelling routines to allow multiple items to be selected from the filter at once (e.g. Figure 5). This means that the evidence base may be queried with custom sub-groupings of any combination of attributes from the data, increasing the flexibility to control precisely what is visualised (in terms of students, groups, activities, competencies, and datasources). Use of the active learner modelling approach has required a limited number of compatibility changes to the algorithms of the back end.

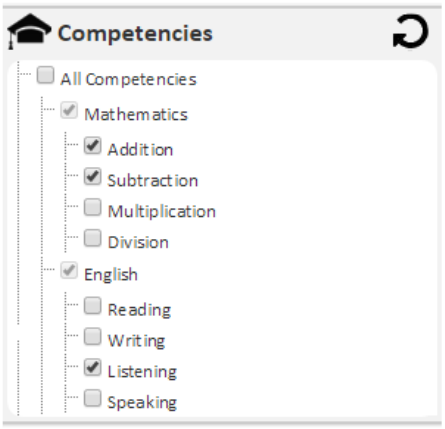


Figure 5: select any combination of items at once.

3.3 Visualising Multiple OLM Dimensions Together

The LEA's BOX learner model has previously been opened as a of a competency framework, a list of students, a list of groups and a list of evidence sources. These can be considered as different dimensions of the learner model, each presenting the same information, but from a different combinatorial perspective; different windows into the same content.

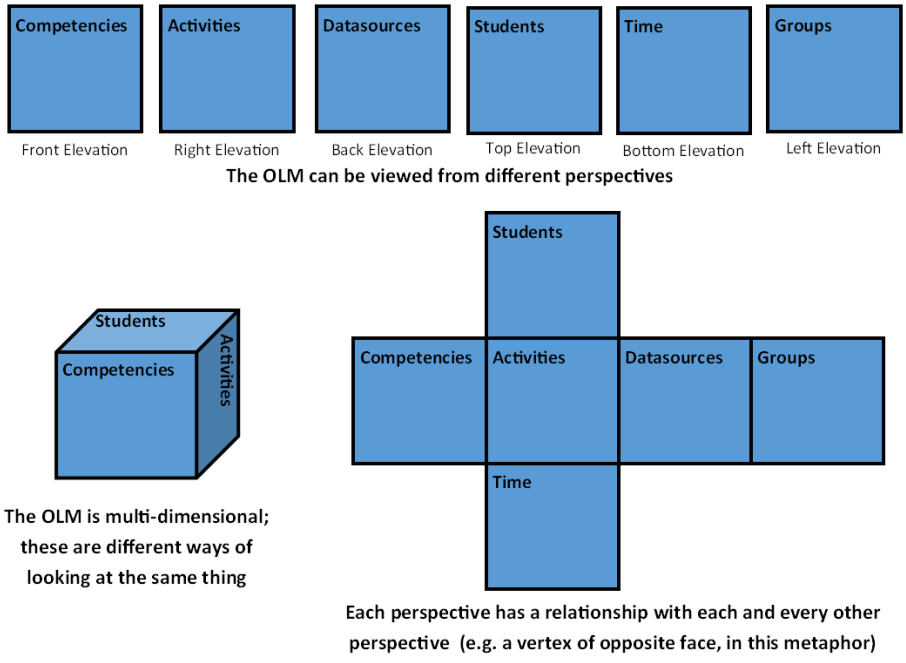


Figure 6: the OLM is multi-dimensional: a cube metaphor.

However, as alluded to in the cube metaphor in Figure 6, these can be considered as different windows into the same content. Each different dimension has a relationship with every other dimension, and it is possible to look at more than one dimension at once. This has the potential to allow for more concise, data rich visualisations and for different viewpoints and insights to be obtained. With added dimensions comes added complexity, but arguably this is an intuitive way to build a relationship with the data, and visualise noisy data [Keim, 2002]. We should also consider the scalability factor with large datasets, with which cross-referencing can be difficult.

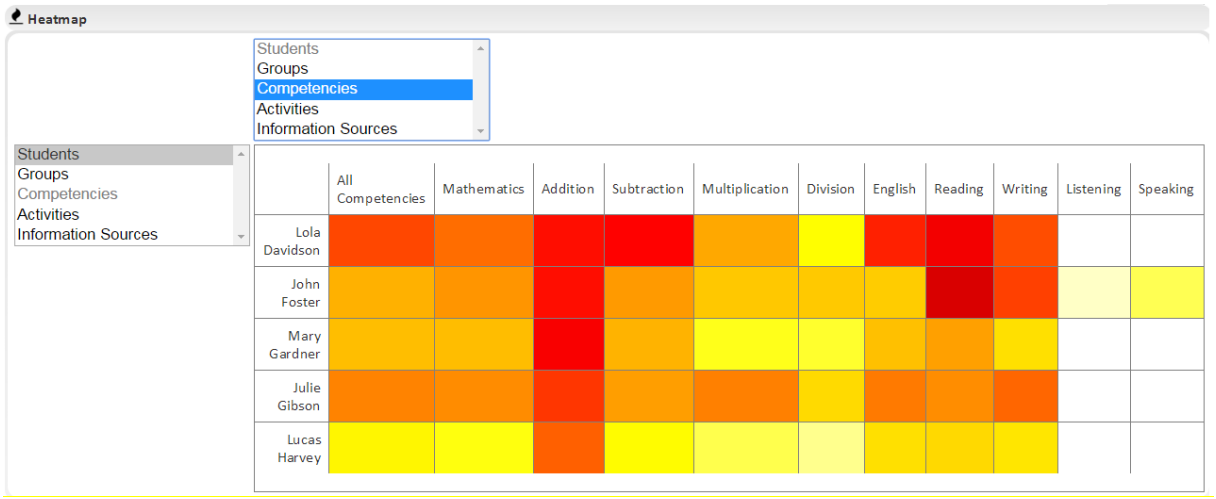


Figure 7: displaying multi-dimensional data: heat-map example.

As part of Release 2, an additional interface is included to allow any two dimensions of the learner model to be compared, using discrete axes (Figure 7). The teacher/student is required to select which type of data is to be placed on each axis, and a visual representation of the data is rendered in the centre. A heat-map matrix is used as the sample visual method for this, where the intensity of red-pigment in colour is used to represent competency in each of the cells. This is seen as an intuitive and simple way to show multivariate data [Few, 2006].

Further visualisation work with multi-dimensional visualisations is planned, pending feedback from initial evaluation with end users, with two dimensions. There is potential here for a range of parallel co-ordinate-based visual methods. Multiple dimensions displayed in the same visualisation may also better display the structural relationships that exist within the educational set, for example which competencies may be linked to which activities.

Scalability is also an issue with large datasets, and this is one area identified for improvement. Visualisation options such as using a fish-eye lens could be one possible solution. There is additionally the overhead of interpreting the visualisation, with the increased level of information available at a single view, and this usability factor needs to be considered further.

3.4 Combining OLM Visualisations with Non-Competency Based Analytics

Notably the attribute list included in Table 3 also contains information regarding *when* the evidence was added (temporal component) and by *whom*. This allows for the learner model to additionally display how it has changed over time – a key requirement resulting from focus groups with end

users. This is one example of how introducing an extra non-competency-based element or dimension to the visualisation extends its utility.

This particular visualisation type currently has an area graph as its implementation: chosen for its high contrast in the plot of two continuous dimensions (competency and time). See Figure 8. Each time the learner model algorithm executes, this is completed, and so a 'trace' has been added to record this information. Key to the display of this is scalability, and so, consistent with the other visualisations, a separate graph is drawn for each item in a list (whilst superimposition of lines is possible, this only works for small datasets). The scales are the same between all graphs, and all graphs are in alignment, for ease of comparison. Zoomability, and the option to add extra fields of data, such as identifying episodes of negotiation, could further increase the utility of this type of visualisation, in situations where the viewer clearly understands the nature of the information that is being presented.

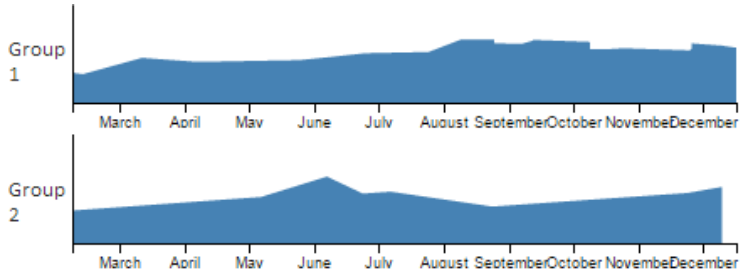


Figure 8: visualising the learner model across time: an example.

Arguably, including more general analytical content further describes the inferences from which the model is constructed, and also has the potential to increase the number of points at which the learner model can be cross-referenced with other aspects of learning and learning-based activity. Combining further, more general analytic components may have the potential to increase the power and informational entropy of OLM visualisations. Table 4 makes some suggestions as to how this may be generated from the data specification laid out in Table 3, and these are noted as 'work-in-progress' as at Month 22, pending further feedback from end users.

Table 4: activity-based analytics that can be generated from the learner model evidence layer.

Metric	Built From	Detail
Level of activity	Time	When information arrives over time; temporal component
Level of information	Row count	How much evidence exists
Intense periods of activity	Time	When information arrives close together, above a threshold. Similar to level of activity, except identifies time during which information arrives in quick succession
Last updates	Time	When a specific item was last updated
Last updates	Time	Which parts of the model was updated last
Contributors	Contributor	Identities of those who have contributed information to a specific part of the model (rather than who/what is affected). Maybe also the level of contributions from each.
Changes in recent activity level / delta	Time	Who/what is more/less active than previously. This is based on recent activity and is a trend indicator. Also identifies what is currently being worked on.

Metric	Built From	Detail
Network of contributors	Contributor	Who is contributing information to the model of another. Potentially also width of the arc can also represent magnitude of information, and the presentation of a node may show the learner model.
Negotiations	Evidencesource	Number of negotiations that have taken place, and when these have occurred
Redundant evidence	Time, evidencesource	Identification of evidence that no longer contributes to the learner model because a negotiation has taken place. Report the level of excluded evidence with potentially a link to the items of evidence
Process	Competency, time, activity etc.	The order in which information is provided for competency frameworks. Where to start, where to go next. Sequence information, showing activity patterns and corresponding changes in the learner model
Lack of information	Competency	Identification of entries/competencies for which there is no information.
In need of more recent information	Time, competency	The most recent information is older than a certain threshold date. Also able to identify easily who has contributed and who has not, if filters are set in such a way.
Concurrency	Time, student	Identification of students who are working on the same thing at the same time. I.e. the same competencies are being updated for the same students
Interaction burst times	Time	Burst times of evidence being added. For evidence added in quick succession, how long are these? Another way of identifying intense period of activity.
Time difference from a target	Time	Dateline is specified. The difference between when an item was last updated
Evidence granularity	Competencyid	A which level of a hierarchical competency structure is evidence being added.

3.5 Properties of the Current Visualisation Set

The current visualisation set is diverse in nature, as elaborated in Table 5. A series of frameworks may be used to describe the attributes of each, and in the analysis of this section we have considered, and combined, elements of [Bertin, 1983], [Pfitzner et al., 2002] and [Mabbott, 2007]. The twelve visualisation options included in this release are collated to cater for:

- individual differences
- varied styles of interactions
- multiple purposes of use
- user preferences
- different visual densities of information
- different levels of complexity
- varied perquisite ability or familiarity resulting from previous experience.

Uses and users of OLM are varied, by their very nature, and this is important to cater for in visualisation design [Bull and Kay, 2010]. Many of the visualisation options are isomorphic, being understandably generated from the same underlying model. Some also display additional aspects of

the information, such as the structure between components, and how components have changed over time, whilst others are deliberately simplified. All visualisations are available to both the student and the teacher; no prescribed use for any is given. Both stakeholder types may configure which visualisations they wish to use.

Table 5: LEA's BOX OLM visualisation set.

Attribute	Skill Meter	Table	Smiley Faces	Stars	Gauge	Histogram	Word Cloud	Radar Plot	Treemap	Network	Across Time	Heatmap
OLM Perspectives	1	1	1	1	1	1	1	1	1	1	1	2
Graphical	●		●	●	●	●		●	●	●	●	●
Textual		●					●					
Quantised		●	●	●		●				●		
Continuous	●				●			●	●		●	●
Structured	●	●	●	●	●				●	●		●
Interactive								○	●	●		
Text Labels	○	○	○	○	○	○	●	○	○	○	○	○
Shape								○		○	○	
Colour							●			●		●
Size									●	●		
Area	●							○			●	
Pattern												
Position					●	●				○		
Proximity										○		
Line thickness												
Quantity				●			○					
Image			●	●	●							
Animation									○			
Hyperlinking									●			
Feel												
Historical information											●	
Current information	●	●	●	●	●	●	●	●	●	●	●	●
Multi-dimensional												●
Value		●		●	●							
Orientation					●							
Texture												
Depth												
Hierarchical	○	○	○	○	○				●	○	○	
Network/arcs										●		

Key: ● attribute ○ present to some extent in the visualisation, but not a core element

Pending feedback from end users in ongoing school engagement and surveys, the visualisation set will be revised and added to during Year 3. Of particular interest are those which display multi-dimensional information, can instigate and support learner model negotiation, and those that can include aspects of more general learning analytics, to increase their utility and the number of cross-referencing points with learning in general. Issues of scalability, usability, and interpretability are core to this.

4 Learner Model Negotiation

The LEA's Box OLM provides learners with a persuasion feature that allow them to obtain evidence for their learner model data and try to persuade the system to make changes to their model by challenging evidence or providing justifications. This persuasion feature aims at making the learner model more accurate, support learner reflection on their learner model contents, as well as their learning more generally, and also facilitate planning and self-monitoring. The negotiation is initiated by the learner and parametrised by teacher. As the negotiation can for now only be initiated by the learner, it can be qualified as persuadable open learner model. However, in the interface and herafter, it is qualified as Negotiation because it is intended to become a "full" negotiation, with interaction symmetry [Baker, 1990].

4.1 Negotiation moves

The possible moves for the system and the learner are presented and illustrated in Table 6. These moves are mainly based on [Baker, 1990] and cover the moves that can be found in the literature ([Bull and Pain, 1995], [Dimitrova, 2003], [Kerly and Bull, 2008], [Van Labeke et al., 2007], [Thomson and Mitrovic, 2010]). As it is a persuasion feature rather than a full negotiation, we can observe two main differences between the moves available for the learners and the system. First, the self-assessment is only available for the student that initiate the negotiation, if the self-assessment is different from the model, the student can try to persuade the system to amend it. Secondly, as the negotiation is initiated by the learner, only they have the possibility to challenge the evidence used by the system to calculate the value in his model. The statement, only available for the system, is not exactly a move but a step between two moves to sum up the current state of the negotiation, like reminding the student's current level, his self-assessment and the justification that he already provided to persuade the system to change his model.

Table 6: Negotiation moves for each stakeholder, with examples.

	Student	System
Accept/agree	Accept a compromise Agree with the system's evidence	Accept a compromise Agree with the student's justifications
Decline	Decline a compromise proposed by the system	"Your last negotiation for this competency is too recent"
Compromise	Propose a compromise between the system's compromise and the student's self-assessment	Propose a compromise between the current level and the student's self-assessment
Request evidence or justifications	Request evidence for the current level	Request justifications for a self-assessment
Provide evidence or justifications	"I did some homework" "I had a class"	"Your level in Writing is 72 and it is a sub-competency of English"
Self-assess	"I think my level should be 80"	×
Challenge evidence	"I disagree with the result of this quiz"	×
Statement	×	"Your level in the competency English is 75 and you think it should be 80"

4.2 Negotiation workflow

The negotiation workflow is given in Figure 9. When a negotiation of a given competency is initiated by the student, the system displays the student's current level for this competency as a statement.

Then, the student can either request evidence or self-assess. The move "request evidence" is available for the student during all the negotiation. The evidence explain how is calculated the current level of a student for the competency being negotiated. It takes into account all pieces of evidence directly associated with this competency and the student's current level in its sub-competencies if any (Figure 10). A direct piece of evidence can for instance be a score to an exam or a quiz, a teacher assessment or the result of a past negotiation of this competency. Each piece of evidence has a weight: the more a piece of evidence is recent the bigger is its weight.

The student's self-assessment is followed by a statement of the system that reminds the student's current level and his/her self-assessment. Then, the system requires justifications in order to increase or decrease the student's level to fit the student's self-assessment. Depending on the student's justification current level and self-assessment, the system uses the negotiation parameters defined by the teacher in order to either accept or decline the student's self-assessment. The system can also propose a compromise between the student self-assessment and his/her current level. If a self-assessment or a compromise is accepted, the negotiation ends and the model is updated with a level that both the student and the system agreed. It will lead to the generation of a new piece of evidence. All evidence that is older than it will no longer contribute to the modelling process for the negotiated competency, but it will remain in the system. If new piece of evidence are added after successful negotiation, the outcome of the negotiation is taken into account like any other piece of evidence in the modelling process (see example of Figure 10). If a self-assessment or a compromise is declined, the negotiation ends but the model is not updated as the system, parameterised by teacher, ultimately retains the control. In both cases, the negotiation is recorded.

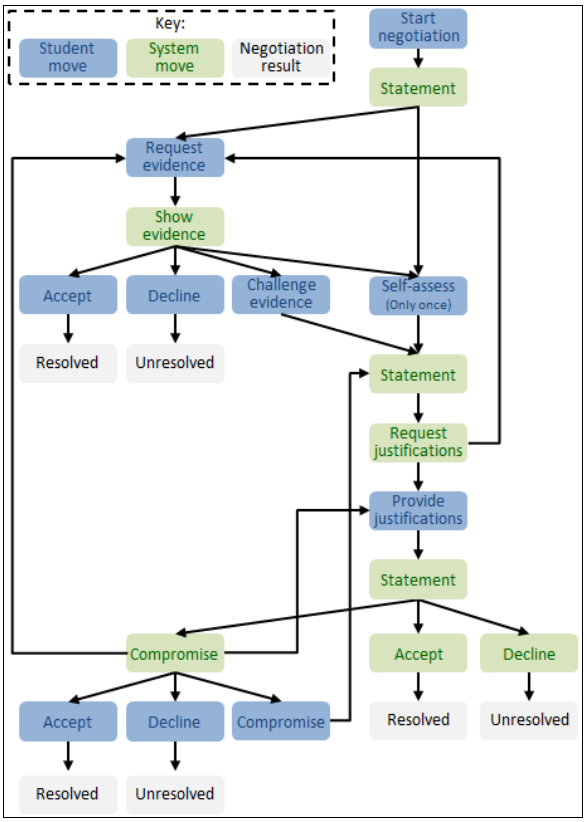


Figure 9: negotiation workflow.

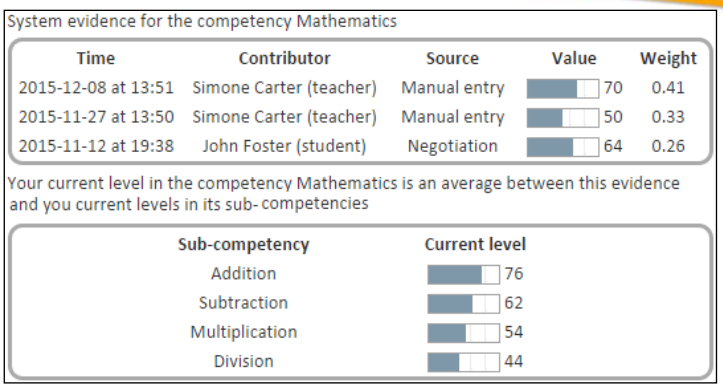


Figure 10: example of system evidence for the competency "Mathematics".

4.3 Negotiation parameters

The negotiation parameters are defined in the teacher's preferences (Figure 11). Thus, the teacher can define a **minimum time between two negotiations**, like "no minimum time", "30 minutes" or "1 week". For instance, if the teacher defines the minimum time between to negotiation as "1 week", it means that if a student negotiate his/her level for a competency for a negotiation has already resulted in an update of the model during the week, the system will decline the student's self-assessment for this reason. The teacher can also define a **minimum number of pieces of evidence with a source other than negotiation between two negotiations**. The other sources of evidence could be for instance a teacher assessment or the result of a pedagogical activity like a quiz. The teacher can define a **maximum threshold to increase** a level and a **maximum threshold to decrease** it. For instance with a maximum threshold to increase a level of 10, with a level defined between 0 and 100, if a student has a level of 65 and self-assess with more than 75, then the system will offer a compromise between 65 and 75. Finally, the teacher can defined the **justifications** that the student can provide during the negotiation, each associated with a maximum weight. When a student self-assessed with a level superior to his current level, he will be able to provide the system with one or several justifications with a positive weight. In this case, if the student's self-assessment is superior to his current level plus the sum of his justifications weights, then the system will offer a compromise between the student's current level and his current level plus the sum of his justifications weights. To the contrary, when a student self-assessed with a level inferior to his current level, he will be able to provide the system with one or several justifications with a negative weight. It is possible to set the model editable, by setting the parameters maximum thresholds to increase or decrease to 100. If these thresholds are set to 0, then no negotiation will be accepted by the system.

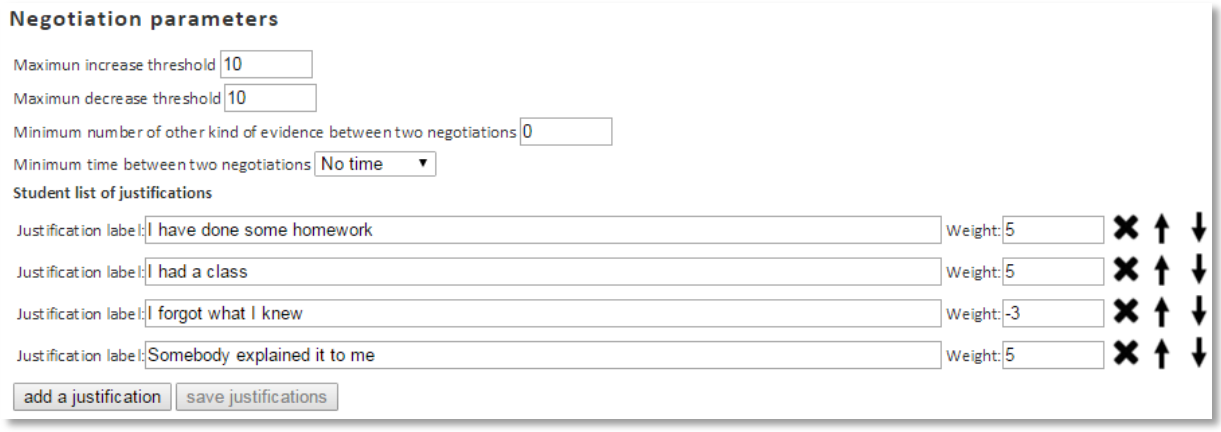


Figure 11: negotiation parameters.

```
if(lastNegoTooRecent || notEnoughOtherEvidence)
{
    Decline();
}
else if(selfAssessement > currentLevel)
{
    offerCompromise = false;
    if (selfAssessement > max) offerCompromise = true;
    if (selfAssessement > calculatedValue) offerCompromise = true;
   if(offerCompromise) Compromise(min(max, calculatedValue));
   else Accept(selfAssessement);
}
else
{
    offerCompromise = false;
    if (selfAssessement < min) offerCompromise = true;
    if(selfAssessement < calculatedValue) offerCompromise = true;
   if(offerCompromise) Compromise(max(min, calculatedValue);
   else Accept(selfAssessement);
}
```

Figure 12: negotiation algorithm.

The system's decision algorithm using these parameters is given in Figure 12, where *lastNegoTooRecent* is a Boolean that is true if the time since the last negotiation of the same competency is inferior to the teacher's parameter; *notEnoughOtherEvidence* is a Boolean that is true is the number of pieces of evidence with a source other than negotiation since the last negotiation is inferior to the teacher's parameter; *max* is an integer equal to the student's current level plus the maximum threshold to increase a level defined by the teacher and *min* is an integer equal to the student's current level minus the maximum threshold to decrease a level defined by the teacher.

Let's take an example: a student is negotiated a competency where his current level is 62 and his self-assessment is 70. The student provided two justifications to persuade the system, one with a weight of +2 and one with a weight of +3. The maximum threshold to increase a level defined by the teacher is 15. In this case, the system will offer a compromise with the value 67 as 62+2+3=67 and 62+2+3 < 62+15.

4.4 Student negotiation

The negotiation starts when the student selects a competency (Figure 13). The student can request evidence in order to understand the system's understanding of his level. If the student wants, he can try to persuade the OLM to change his level. For this purpose, he first has to tell the OLM what level he thinks he should have, then click on the button "Negotiate".

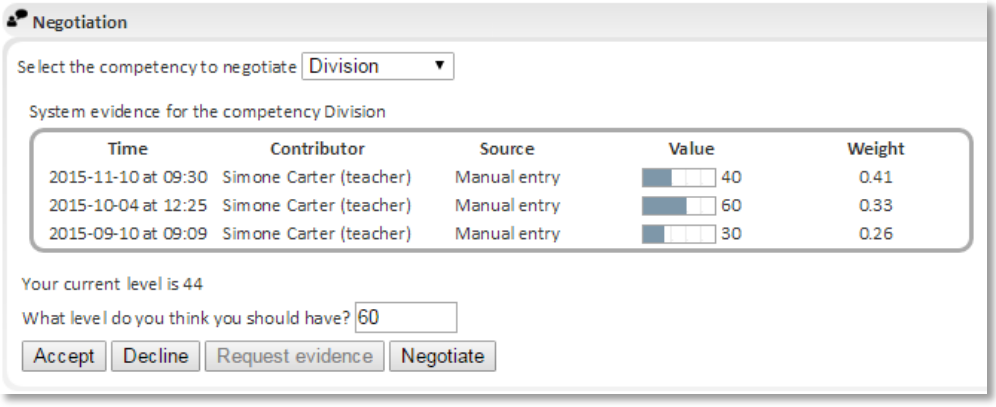


Figure 13: starting of a negotiation for the competency division.

During the negotiation, the OLM will ask the student to justify the change to your level. The justifications available are defined in the negotiation parameters. The student can also challenge a piece of evidence. Once this has been done, the student has to click on the button “Continue”.

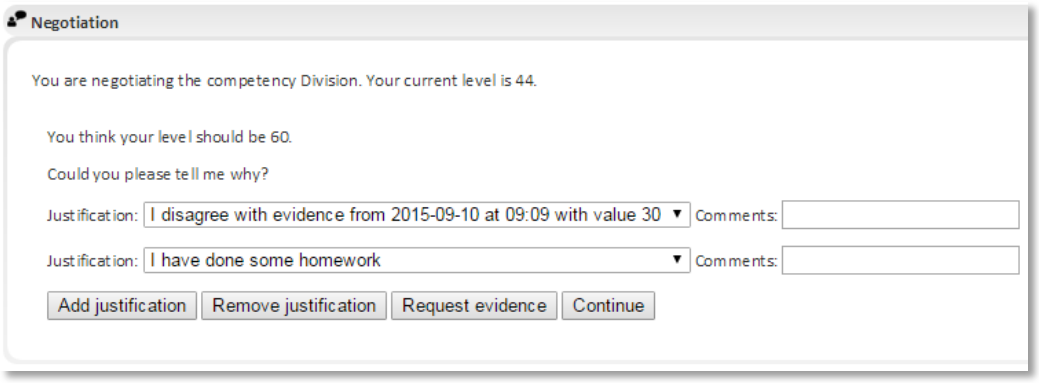


Figure 14: example of negotiation for the competency division.

Depending on the negotiation parameters, on the student’s justifications and on the difference between his current level and the level he thinks he should have, the OLM can accept his proposition of change or offer a compromise. The OLM can also decline a negotiation if the last one is too recent or if there is not enough piece of evidence since the last negotiation on this competency. If the OLM offers a compromise (Figure 15), the student can either accept it, decline it (that put an end the negotiation), try to persuade the OLM that he is right by adding more justification, or offer another compromise.

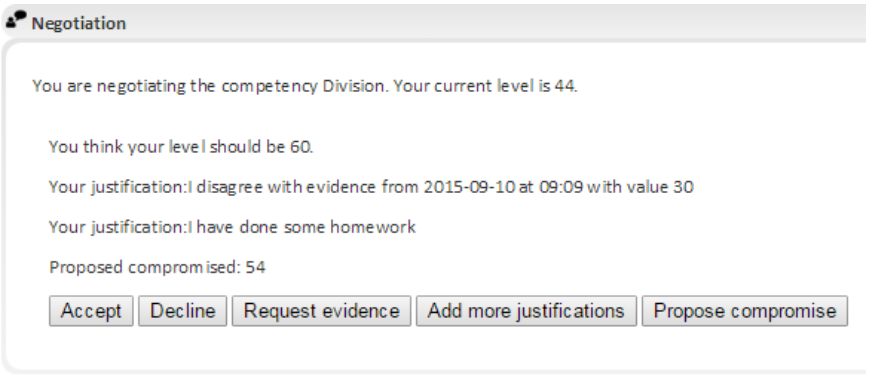


Figure 15: example of a negotiation with a compromise proposed by the OLM for the competency division.

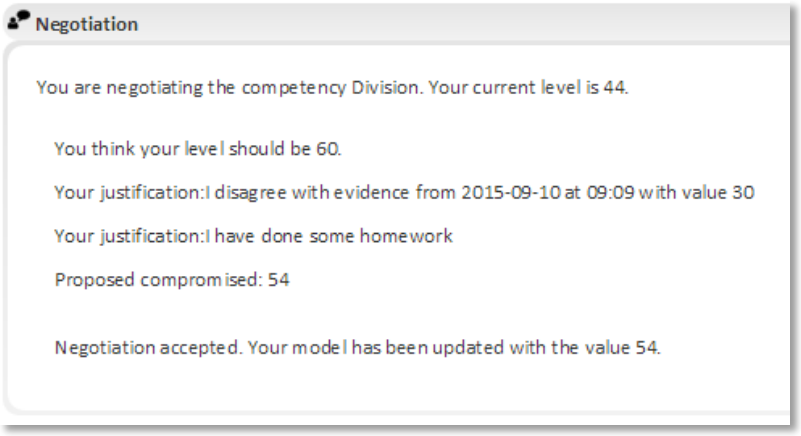


Figure 16: example of a negotiation resolved for the competency division.

If a negotiation is accepted by the student and the OLM, the model is updated (Figure 16). The evidence will be replaced by a new piece of evidence with source negotiation and the value that the student and the OLM agreed (Figure 17).

Figure 17: outcome of a negotiation resolved for the competency division.

5 Revised Back End Services and Configuration

In order to allow for tighter integration, and revisions to the OLM visualisations, driven by use cases, Release 2 contains updates to the back end of the system. In this section we report on the key changes to the architectures of information flow (Section 5.1) and of informational relationships (Section 5.2), in addition to revised APIs (Section 5.3), and data format of the visualisation service (Section 5.4).

5.1 Modular Architecture

The overall architectural structure of the flow of information between the modular components of the learner model is retained and extended to include facilities for learner model negotiation (Figure 18). The additional modular component of the negotiation logic uses information directly from the learner model algorithm, teacher and system preferences, and from the visualisations, to present the user with a dialogue that facilitates negotiation (also see Section 4).

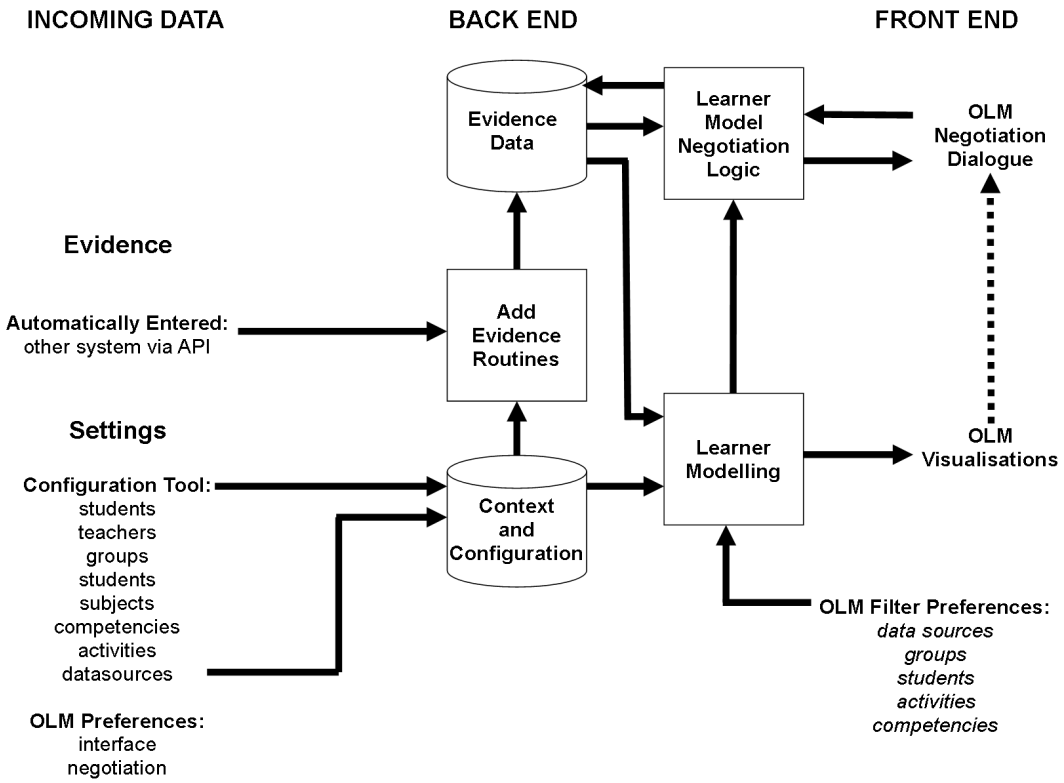


Figure 18: revised information flow architecture.

These additional algorithms function as a further layer, allowing information present in the visualisations to be negotiated, and the outcomes of negotiation to contribute as a special form of evidence that the learner modelling process. (See D4.6 for the specification and technical description for this.) This creates an additional layer in the information stack (as shown in Figure 19), together with an action that the OLM system is itself able to amend the data in its evidence layer, and to also be able to identify evidence that should not have any influence in the modelling layer.

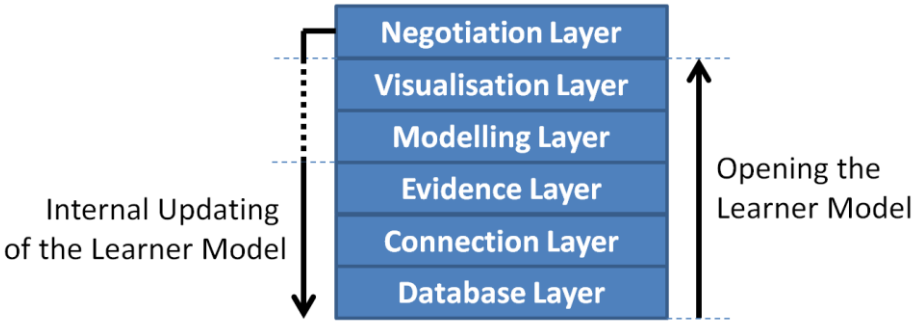


Figure 19: OLM information stack.

5.2 Informational Architecture

Improvements in the integration work between the Configuration Tool and the OLM also mean that pre-configuration can take place in terms of what should be present in the OLM. All information is available for presentation when it is pedagogically relevant, rather than relying on the data as the link between these different educational entities. (This is a revision from the initial specifications.)

The architecture, shown in Figure 20, is now implemented. The addition of subjects and activities allows for a more curriculum-based focus. Competencies are assigned to one subject only, although may be associated with multiple activities within this. Within this architecture, a subject must thus exist in order for activities and competencies to be created. The relationships between students/teachers, groups, and subjects are such that these may be added in any order (and as with all elements, amended at any stage). Where links are not present directly between entities, these are derived using the other existing relationships. Notably items of data (competency inferences) are principally linked to only one activity, and whilst the source of data needs to exist prior to the data being added, an explicit link between the data and activity is not needed in anticipation of this. A full definition of the prerequisites for adding data is given in Section 6.1. See D2.5 for more information about the configuration. The numbers in Figure 20 refer to API events, see Section 5.3.

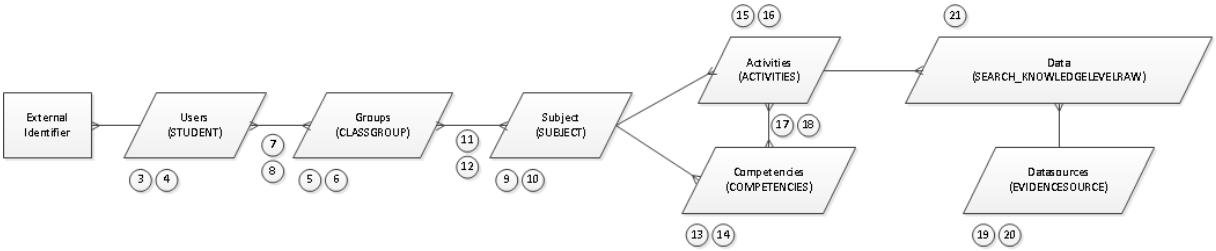


Figure 20: informational architecture.

Furthermore, improvements in the API now mean that the configuration tool is able to configure modelling parameters, required by the modelling process. Namely these are a “competency influence” and an “activity influence”. These are to be used when a teacher decides that a given competency or activity should contribute to a greater or lesser extent to the model. This is to say, the activity’s/competency’s importance relative to others - this is reflected in the way in which evidence is combined. This is part of API functions 13 and 15 in Table 7.

A transfer to the new informational architecture has mandated a revised data structure. This is shown in Figure 21, together with each of the primary keys (PK), foreign keys (FK), and attributes that are denormalised (*den) for reasons of performance.

Database tables are also present for preferences and the configuration of the negotiation layer algorithms. The attributes persisted here support users in their interaction with the open form of the learner model, and, as such, the interface. These are linked by user credentials only.

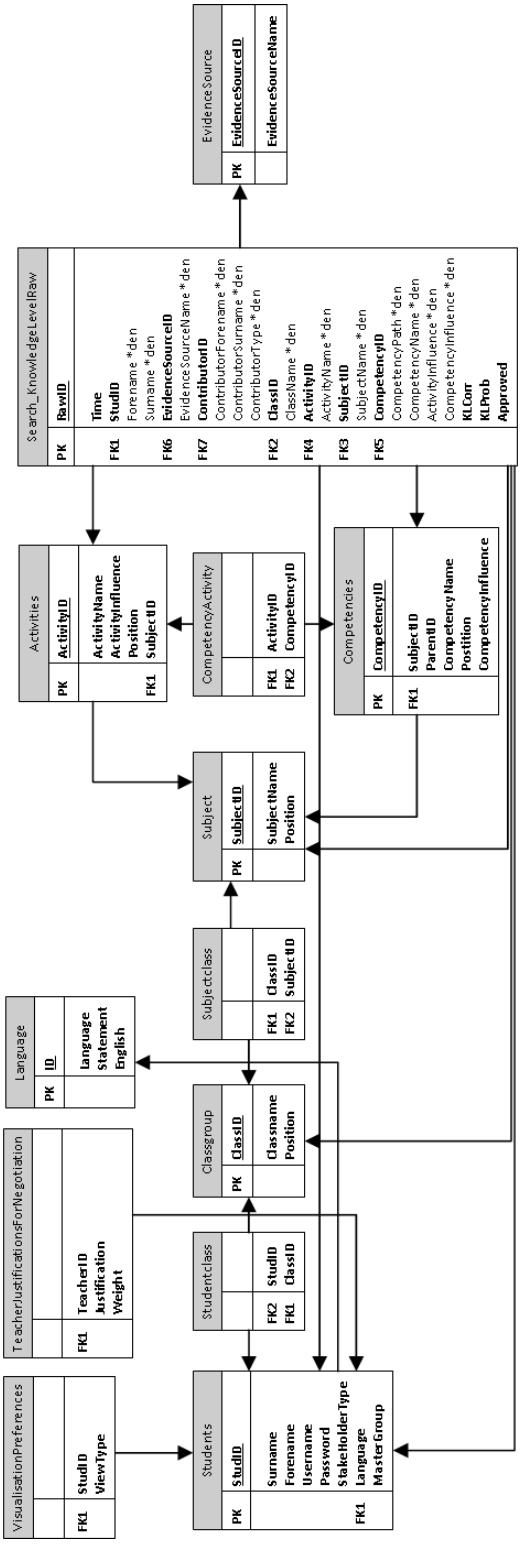


Figure 21: revised database structure.

5.3 Revised API Specification

Each element and relationship in the architecture of Figure 18 has a corresponding API method, to which the numbers in the figure refer. These are summarised in Table 7, and a full specification is given in Appendix 3. Each method's function is categorised with create, read, update or delete functionality. Methods 1 and 2 relate to the authentication mechanism, whilst those categorised as a 'definition' relate mainly to the boxes in the architecture of Figure 18, and those defined as 'relationships' relate to the lines between entities (with the exception of the 1:∞ relationships with regard to subjects and items of evidence/competency inferences).

Table 7: API functions.

ID	Function	Mode	Type
1	Log In	Create	Authentication
2	Log Out	Delete	Authentication
3	Add / Update User	Create, Update	Definition
4	Delete User	Delete	Definition
5	Add / Update Group	Create, Update	Definition
6	Delete Group	Delete	Definition
7	Add User to Group	Create	Relationship
8	Delete User from Group	Delete	Relationship
9	Add / Update Subject	Create, Update	Definition
10	Delete Subject	Delete	Definition
11	Add Subject to Group	Create	Relationship
12	Delete Subject from Group	Delete	Relationship
13	Add / Update Competency	Create, Update	Definition / Relationship
14	Delete Competency	Delete	Definition
15	Add / Update Activity	Create, Update	Definition / Relationship
16	Delete Activity	Delete	Definition
17	Add Competency to Activity	Create	Relationship
18	Delete Competency from Activity	Delete	Relationship
19	Add / Update Data Source	Create, Update	Definition
20	Delete Datasource	Delete	Definition
21	Add Data and Competency Information	Create	Definition / Relationship

5.4 Visualisation Service

The visualisation service (see also Figure 18) is used to render datasets that result from opening the learner model, to make them interpretable. This reusable service has been further extended from Release 1 to allow for the incorporation of temporal data in its JSON data format, the revised structure of which is shown in Figure 22. Further methods are also added to provide more support

for D3 (Data Driven Document) libraries, including the ability to read Tab Separated Values (TSV) files. Moving forward into the final year of development, this service is better equipped to support development of more state-of-the art and interactive visualisations. Backwards compatibility is maintained, together with its lightweight and flexible PHP implementation. A more complete description is given in D4.2.


```
{
  "child0": {
          "name": "Robert Brown",
          "data": {
                  "item0": {
                          "value": 0.85467,
                          "time": "2015-11-04 14:32:58",
                          "negotiation": false
                        }
                  item1: {
                      "value": 0.249,
                          "time": "2015-11-10 15:49:36",
                          "negotiation": false
                        }
                  item2: {
                      "value": 0.55,
                          "time": "2015-11-22 11:17:27",
                          "negotiation": true
                        }
               },
          "title": "0.55",
          "id": 1578,
          "source": "students"
     }
  "child1": {
          "name": "Alison Kline",
          "data": {
                  "item0": {
                          "value": 0.149875,
                          "time": "2015-11-04 14:49:15",
                          "negotiation": false
                    }
                  "item1": {
                          "value": 0.348967,
                          "time": "2015-11-04 19:09:11",
                          "negotiation": false
                    }
                  "item2": {
                          "value": 0.78594,
                          "time": "2015-11-10 15:35:48",
                          "negotiation": false
                    }
                  "item3": {
                          "value": 0.648975,
                          "time": "2015-11-22 11:12:45",
                          "negotiation": false
                    }
                  "item4": {
                          "value": 0.789824,
                          "time": "2015-11-22 11:32:07",
                          "negotiation": false
                    }
            },
          "title": "0.789824",
          "id": 1594,
          "source": "students"
     }
}
```


Figure 22: JSON structure of data for the visualisation service.

6 Import of Data

The LEA's BOX OLM is designed to take data from multiple sources and contains no facilities to generate educationally related data itself; the system takes existing competency-based inferences relating to different students, groups and activities, and models these to give a prediction of student competency across diverse educational datasets. For this it relies on pre-configuration and data entered via API calls. In this section we look in greater detail at API call 21 (Section 6.1), and give an example of a tool added to the LEA's BOX Portal to import inferences to the learner model's evidence base, via CSV file import from client tools (Section 6.2).

6.1 Importing Data via the API

API call 21 ("addinformation") is the method used to add competency-based inferences to the evidence layer of the OLM (shown in Table 8). A tightening of overall system architecture means that prerequisite relationships are now required, to provide the full context of each inference added. (See also Section 5.2.)

Table 8: API Function 21: addinformation.

Arguments	Description	Returns
sharedsecret leasid method competencyid groupid userid datasourceid value activityid	access password id number of the user logged in "addinformation" id of competency id of the group id of user to be updated id of the datasource inference value (range: 0 to 1) id of the activity	On success of adding information: 'information added: "<time of addition>" value:"<value>"'. Else 'competency does not exist in the database', 'group does not exist in the database', 'user does not exist in the database', 'datasource does not exist in the database', 'activity does not exist in the database', 'user is not a member of the group', 'user is a teacher', 'activity not assigned to class', 'competency not assigned to class', 'competency not part of activity', 'value is not a number', 'value should be in a range of 0 to 1', 'adding information failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=addinformation&competencyid=198&groupid=72&userid=1001&datasourceid=20&value=0.756&activityid=911		

Meaningful XML error messages are sent should one of the following prerequisite conditions not be met:

- competency must exist
- group/class must exist
- student must exist
- evidence/datasource must exist
- activity must exist
- student must be a member of the class/group

- student must be a student
- activity must be assigned to the group/class
- competency must be assigned to class/group
- competency must form part of the activity
- value for the inference must be in the range of 0 (no competence) to 1 (competent)

In the event where the same inference is to update multiple entities (e.g. multiple students, multiple competencies) then the API call must be made individually for each combination of these.

6.2 Importing Data from Other Systems, using CSV

There are frequent cases where competency-based inferences exist and need to be imported in batch, but real time automatic transmission of data is not a feasible option due to required development work, or permissions to edit client software. A CSV import tool is now included within the main LEA's BOX portal, for adding data directly into the learner model evidence layer. This is an example that may be extended to incorporate data from different systems in different formats.

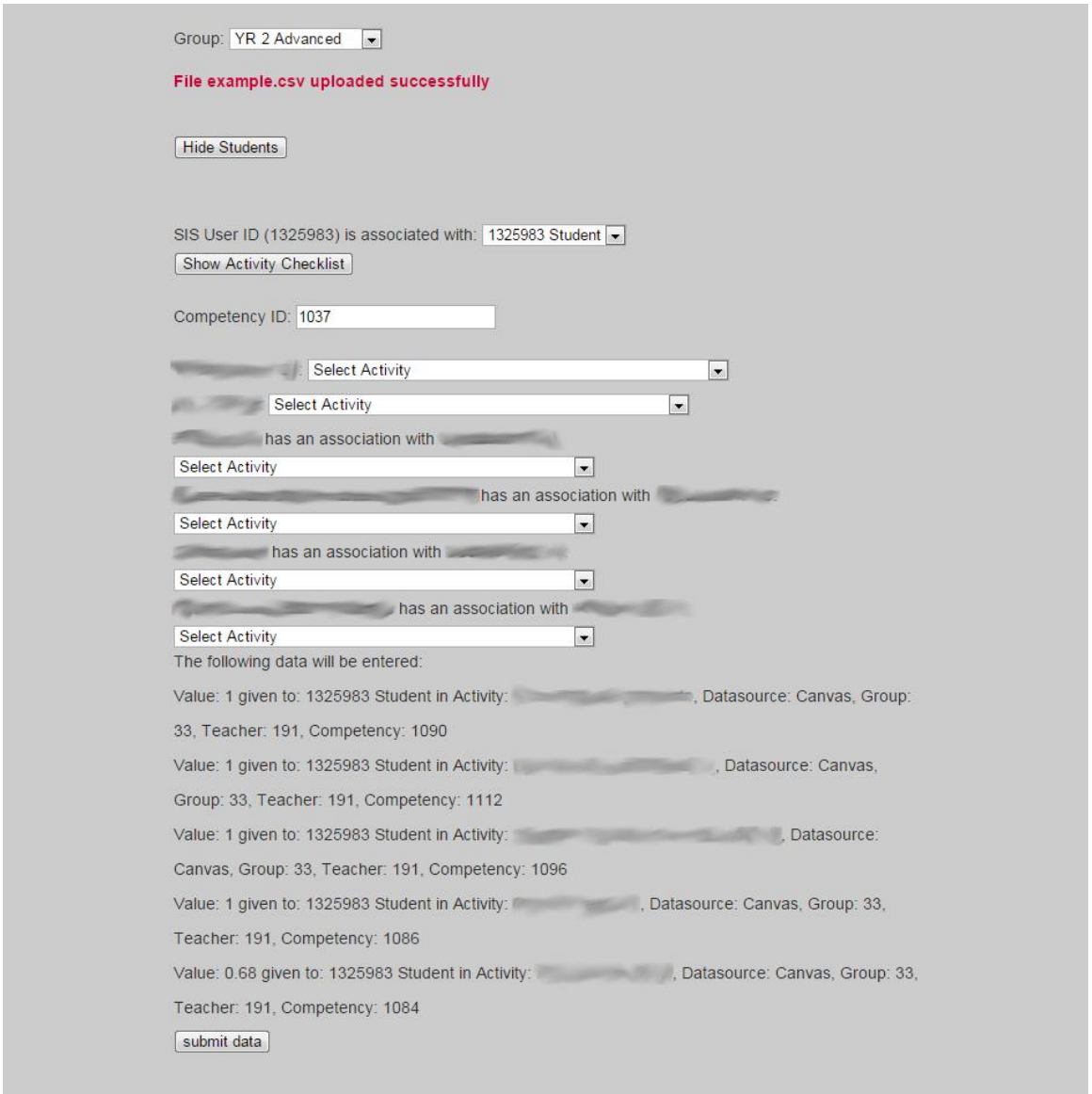


Figure 23: spreadsheet import tool.

Integrating with external systems and patching between data formats has the potential for great complexity, not least of all when some of the mapping information is missing. Synchronisation with the LEA's BOX global configuration, so that all context information is present, is a further challenge (i.e. the prerequisite list stated in Section 6.1). We present an example solution for patching data from quizzes in the content management system Canvas through to the learner model via CSV, which detects missing information. A GUI is provided (Figure 23) for the use to complete this. The missing information is entered once only, and then is remembered for subsequent imports by the same user from the same source of information.

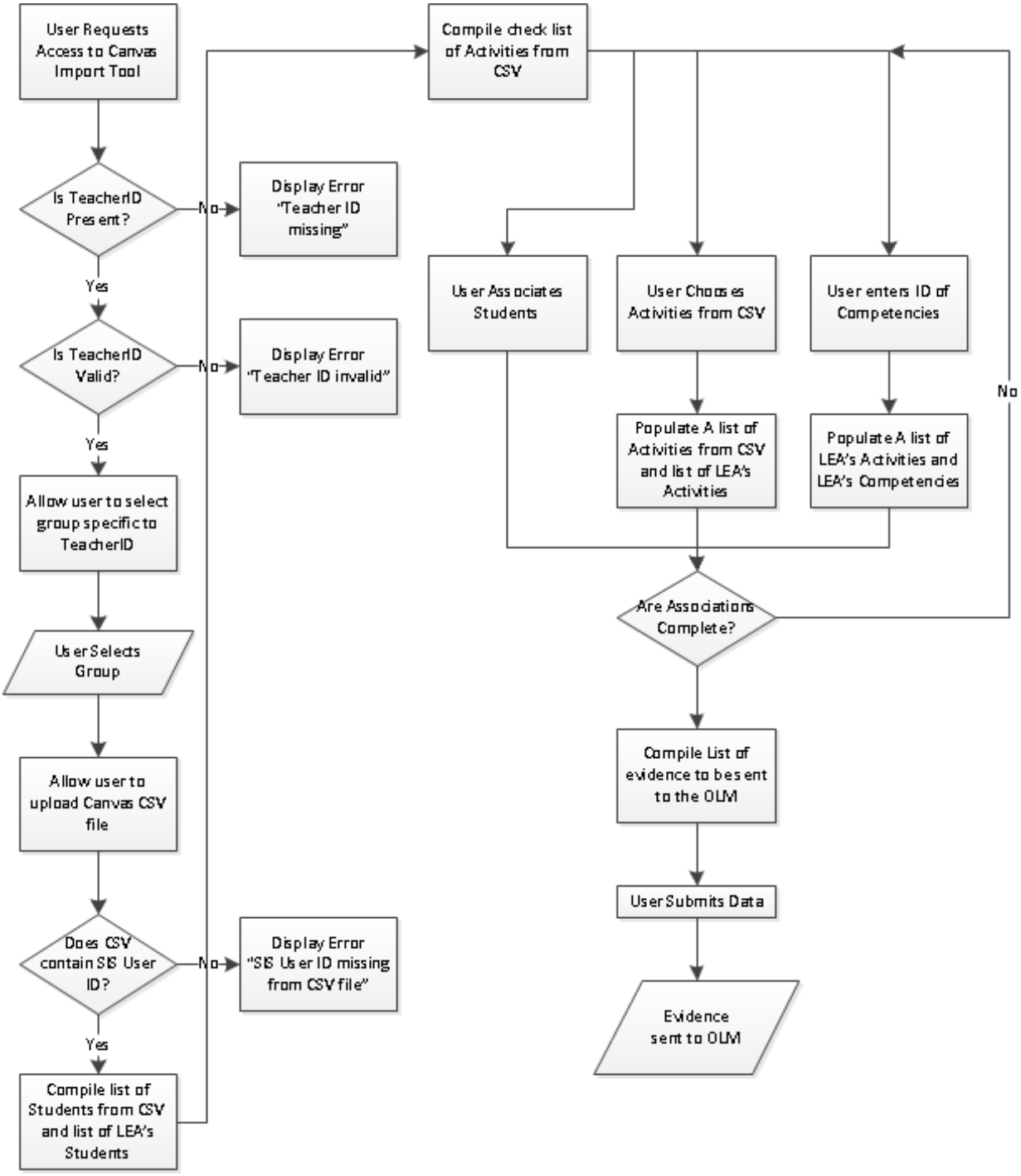
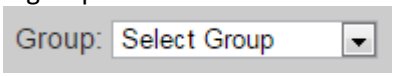


Figure 24: workflow for importing data via the spreadsheet import tool.

The overall workflow for an import is shown in Figure 24. This includes error checking clauses. The user is required to be logged in as a teacher to complete the import. For the user this implied the following sequence of steps:

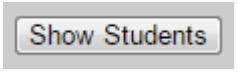
1. Accessing the resource via the portal will redirect the user to an embedded page help at: http://css-kmi.tugraz.at/mkrwww/leas-portal/CanvasImport/upload.php
2. A group should be selected with which the information should be associate

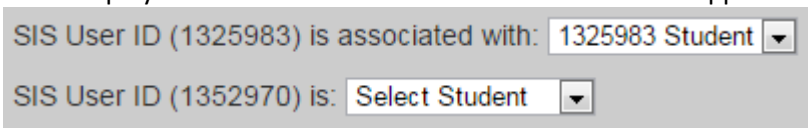
3. A Canvas CSV file is selected and uploaded to the system and a feedback message is give

4. Upon receipt, the system will look for the "SIS User ID" field and will generate a list of students that is hidden by default. Click the "Show Students" to reveal them

5. Once displayed the students from the Canvas CSV file will appear as follows:

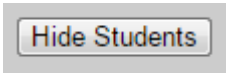
   

   Students that already have their SIS User ID associated with a student inside LEA's Box will appear as the first line. Students that have no association will appear as the second line.

   Associations are made automatically once set (there are no submit buttons for this step). To remove a student's association within the system, set the option within the dropdown list to "Select Student".

6. After completing the associations with the students in the CSV and LEA's BOX, click the "Hide Students" button.

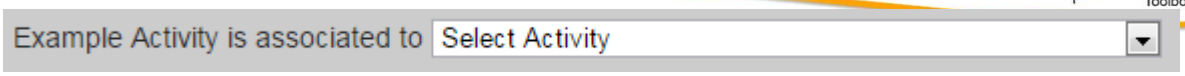
7. The next step is to complete the association between the activities in the import file and in LEA's BOX. If the associations between the Activities within the Canvas CSV file and LEA's Box don't already exist then they need to be created. First reveal the list of Activities found within the CSV file by pressing "Show Activity Checklist"

8. Upon revealing the Activity checklist, every column in the Canvas CSV will be shown in a list. Any column from the CSV you wish to associate with an Activity inside LEA's Box must be selected by the check list

9. When an Activity from the CSV is selected, the ability to associate the Activity from the CSV to the LEA's BOX is instantly available.



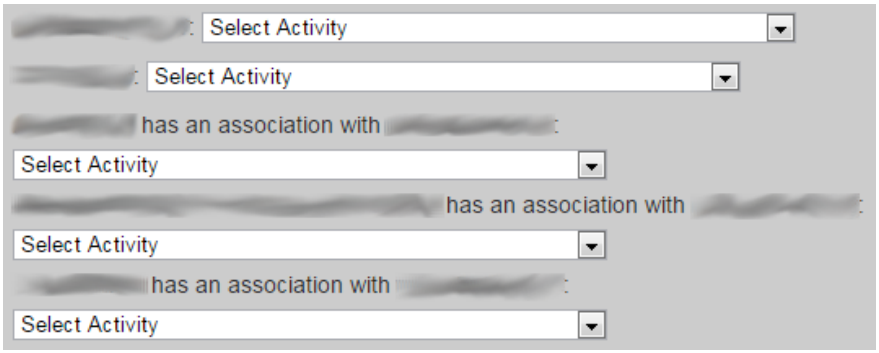
Selecting an association from the drop down box will create association is made immediately (there are no submit buttons for this stage). Similarly, in order to remove an association previously created, set the associated activity to "Select Activity".

10. Once the Activities from the CSV file have been associated with the Activities from the LEA's Box, the user can now associate those activities to competencies. At the present time this this requires the aid of the portal's configuration tool to obtain the ID numbers of the competencies you wish to work with. (This is an area identified for future improvement.)

ID
1037
1038

11. Once the ID the parent competency is known, this should be added to "Competency ID"

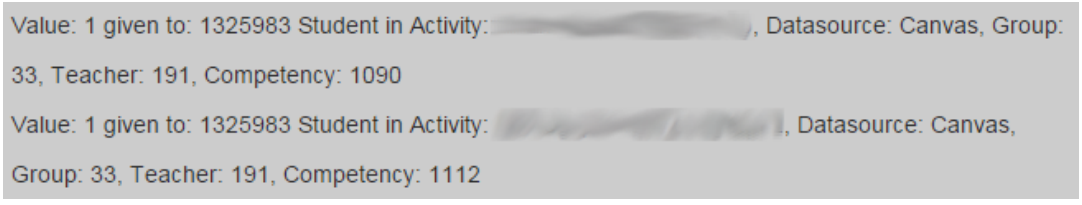
A competency hierarchy is then created from the LEA's BOX



Multiple activities can be associated to a competency, simply select another activity and the system will automatically associate that activity to the competency.

Currently associations between Activities and Competencies cannot be deleted. (This is an area identified for future improvement.)

12. When associations have been established between an activity, datasource, competency, group, student, and the values to be imported, then the data is collated and presented for inspection as per the following example. (The addition of names for the ID numbers is identified as an area needed for improvement.)



13. Once the user has reviewed all the data the user can then click on the "Submit Data" button at the bottom of the page. After the user sends the data the "Submit Data" button will change to read "Data Processed" and will prevent the user from submitting the data again. The import is complete.

7 References

[Baker, 1990] Negotiated Tutoring., an Approach to Interaction in Intelligent Tutoring Systems, unpublished PhD thesis, Open University, Milton Keynes, UK.

[Bertin, 1983] Bertin, J. (1983) The semiology of graphics. Univ. Wisconsin Press: Madison, Wisc.

[Bull and Kay, 2010] Bull, S. and Kay, J. (2010) Open learner models, in Advances in intelligent tutoring systems. Springer Berlin Heidelberg, pp. 301-322.

[Bull and Pain, 1995] Bull, S. & Pain, H. (1995). 'Did I Say What I Think I Said, And Do You Agree With Me?': Inspecting and Questioning the Student Model, in J. Greer (ed), Proceedings of World Conference on Artificial Intelligence and Education, AACE, Charlottesville VA, 501-508.

[Dimitrova, 2003] Dimitrova, V. (2003). StyLE-OLM: Interactive Open Learner Modelling. Int. Journal of Artificial Intelligence in Education 13(1), 35-78.

[Elias, 2011] Elias, T. (2011), Learning Analytics: Definitions, Processes and Potential. . Retrieved from http://learninganalytics.net/LearningAnalyticsDefinitionsProcessesPotential.pdf

[Few, 2006] Few, S. (2006) Multivariate Analysis Using Heatmaps. Perceptual Edge, 10th Oct 2006.

[Keim, 2002] Keim, D. (2002) Information Visualization and Visual Data Mining, in IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 1, January-March 2002.

[Kerly and Bull, 2008] Kerly, A. & Bull, S. (2008). Children's Interactions with Inspectable and Negotiated Learner Models, in B.P. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (eds), Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg, 132-141.

[Mabbott, 2007] Mabbott, A. (2007) User Choice in Viewing and Interacting with Open learner Models. PhD Thesis, University of Birmingham, December 2007.

[Pfitzner et al., 2002] Pfitzner, D., Hobbs, V. and Powers, D. (2003). “A unified taxonomy framework for information visualization”, Proceedings of Australian Symposium on Information Visualisation (InVis, Au’03), 57-66.

[Thomson and Mitrovic, 2010] Thomson, D. & Mitrovic, A. (2010). Preliminary Evaluation of a Negotiable Student Model in a Constraint-Based ITS, Research and Practice in Technology Enhanced Learning 5(1), 19-33.

[Van labeke et al., 2007] Van Labeke, N., Brna, P. & Morales, R. (2007). Opening Up the Interpretation Process in an Open Learner Model, Int. Journal of Artifial Intelligence in Education 17(3), 305-338.

APPENDIX 1: OLM User Manual

Interface Structure and Components

The interface constitutes one primary webpage, which acts as a browser for the open representation of the learner model (Figure 25). 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 25 to Figure 38 use test data to show where information will appear on the screen. Visualisations are covered in Section 0.

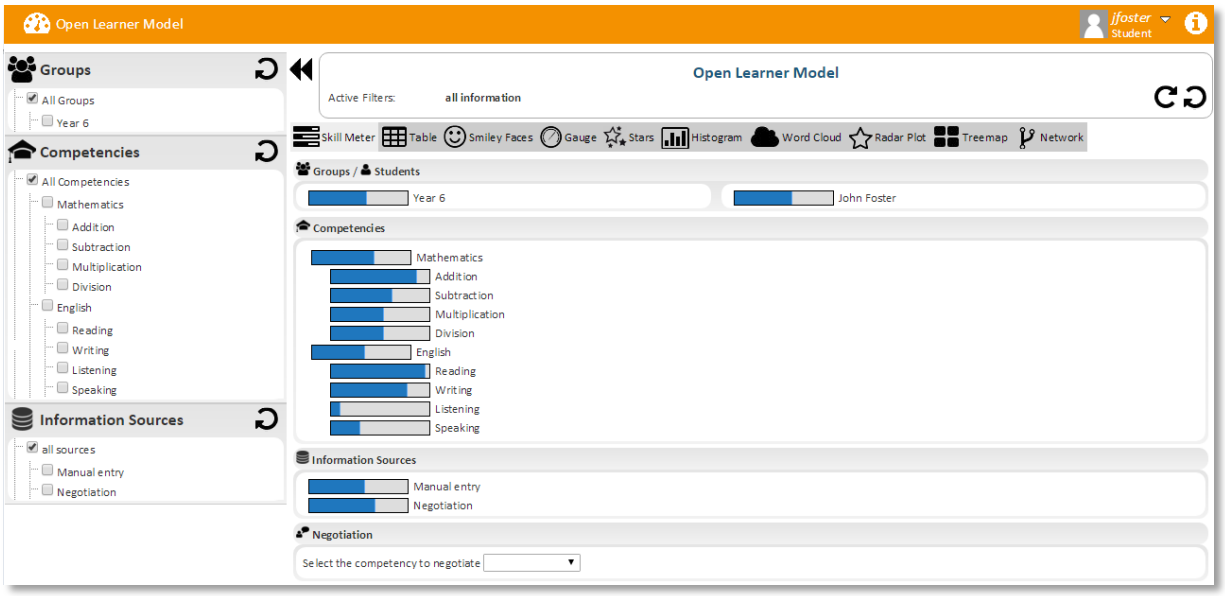


Figure 25: OLM browser interface.

The following key facilities are included:

- **Information filters** (left of Figure 25). 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 25). 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 and displaying the returned graphic or HTML content.
- **Breadcrumb and functions** (top centre and right of Figure 25). The filters currently applied, and the 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 25). 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, Turkish 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 25). This will display basic guidance on how to operate the browser.
- **Negotiation** (bottom of Figure 25). This facility, described in section, is only available for students.

Iconography and Localisation

Icons are used consistently throughout the browser. (Figure 26).

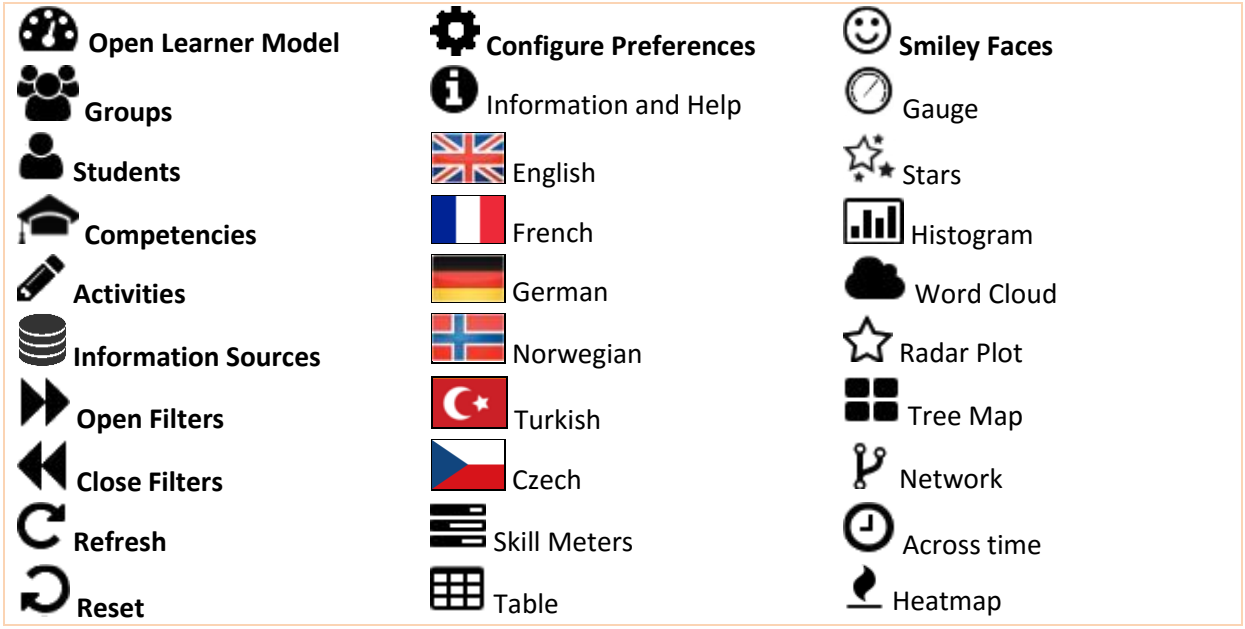


Figure 26: iconography.

The interface is localised into six languages: English, French, German, Czech, Turkish and Norwegian.

Visual Methods

The OLM set of visualisations consists of twelve 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 9).

Table 9: LEA's Box OLM visualisation set.

Visualisation	Graphical	Textual	Quantised scale	Continuous scale	Structure	Unstructured	Interactive
Skill Meter	✓			✓	✓		
Table		✓	✓		✓		
Smiley Face	✓		✓		✓		
Stars	✓		✓		✓		
Gauge	✓			✓	✓		
Word Cloud		✓	✓			✓	
Histogram	✓		✓			✓	
Radar Plot	✓			✓		✓	
Treemap	✓			✓	✓		✓
Network	✓		✓		✓		✓
Across time	✓			✓	✓		
Heatmap	✓			✓		✓	✓

The visualisations coloured on Table 9 have been added since the last release. Since the last release, four visualisations have been added: stars, gauge, across time and heatmap. The two first are very simple but answer to a need of customisation. The stars have characteristics equivalent to smiley faces, but are less "children connoted". The gauge have characteristics equivalent to skill meters that is one of the most popular visualisation, but with a very different design. The 'across time' visualisation answers to a need to represent the model evolution across time. It is not possible with the other LEA's Box OLM visualisations even if this kind of visualisation is frequent in OLM. The heatmap visualisation is a multidimensional visualisation, it answers to a need to represent on a same graphic two kinds of information. For instance a heatmap can represent on a same graphic the data coming each information sources for each competency.

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 27.)

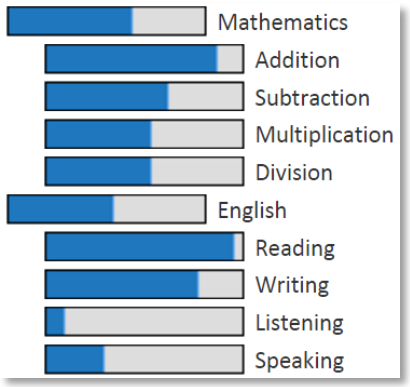


Figure 27: skill meter visualisation.

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 28**.)

	Very Weak	Weak	OK	Strong	Very Strong
All Competencies			●		
Mathematics				●	
Addition					●
Subtraction				●	
Multiplication			●		
Division			●		
English			●		
Reading					●
Writing				●	
Listening	●				
Speaking		●			

Figure 28: table visualisation.

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 29.)

Stars

Similar to the smiley faces visualisation, the stars have a scale is quantised into 5 levels, from *one star out of five* indicating little/no competence, through to *five starts out of five* indicating competence. Again, indentation is used to show hierarchical structure.

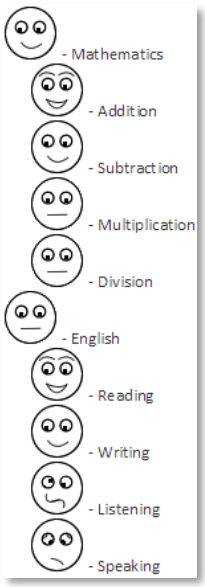


Figure 29: smiley face visualisation.

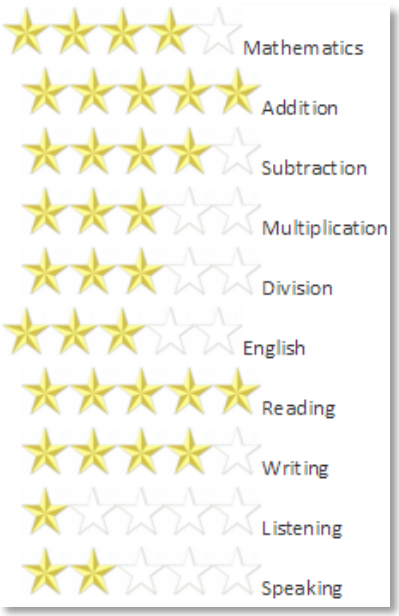


Figure 30: stars visualisation.

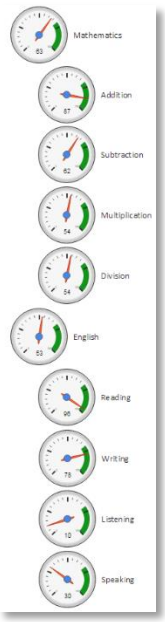


Figure 31: gauge visualisation.

Gauge

Similar to the skill meters visualisation has a continuous scale. The position of the arrow in the gauge indicates the student's competency in the area.

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 32.)

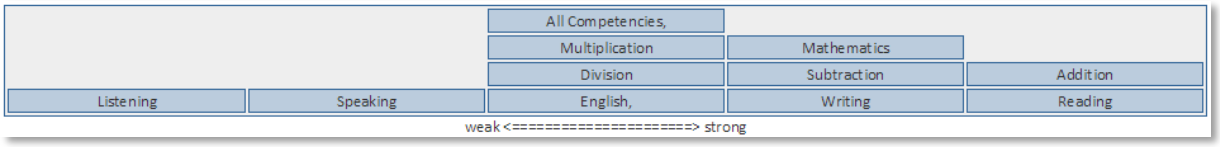


Figure 32: histogram visualisation.

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 33.)

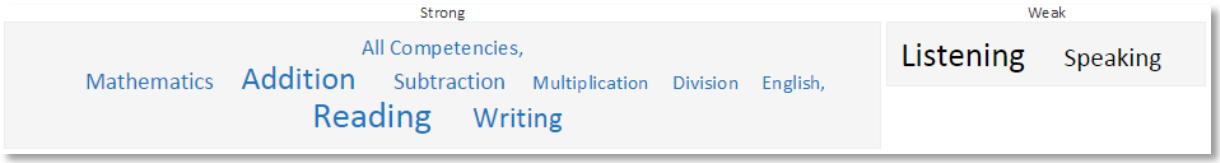


Figure 33: word cloud visualisation.

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 34.)

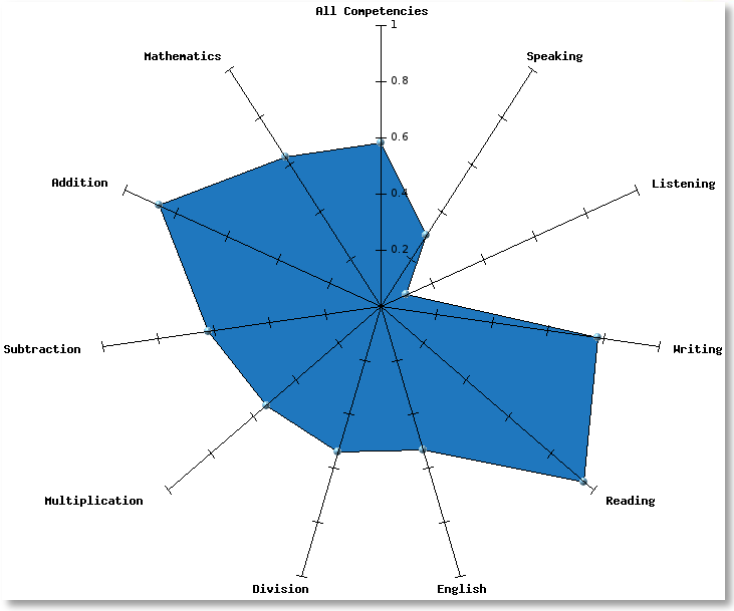


Figure 34: radar plot visualisation.

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 35.)

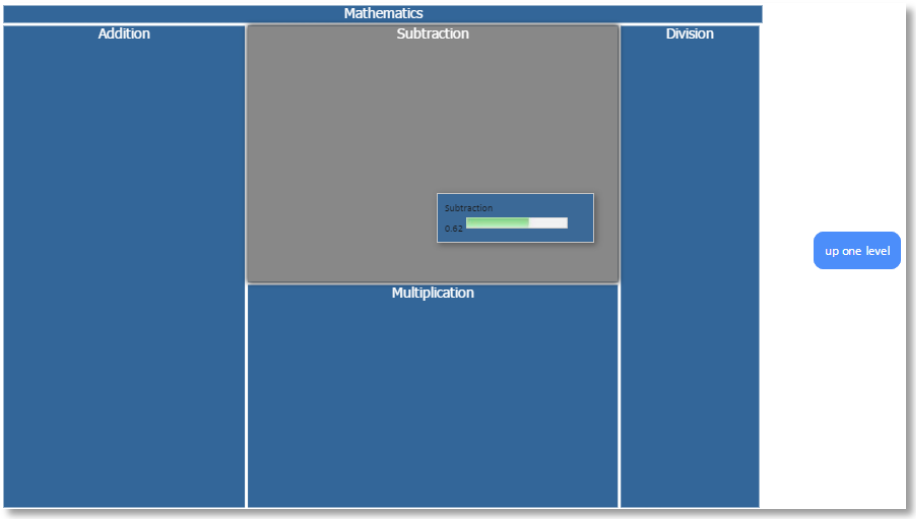


Figure 35: treemap visualisation.

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 36.)

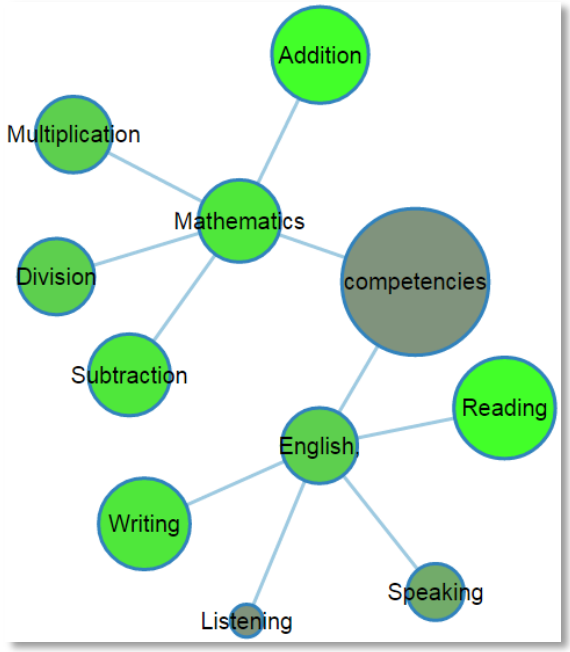


Figure 36: network visualisation.

Across time

The across time visualisation presents an area graph for each item to be visualised in the OLM. This is the state of the model across time. Competency is shown on the y axis, and time on the x axis. All scales are the same between graphs, and graphs are shown in alignment for ease of comparison.

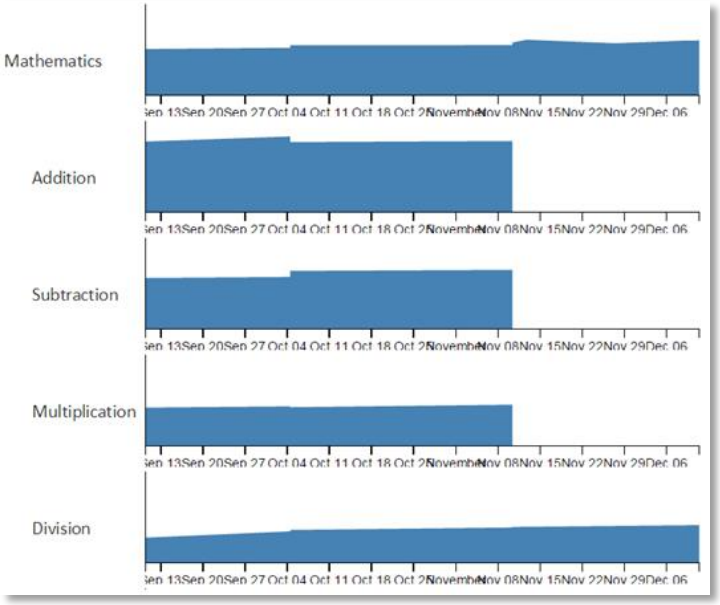


Figure 37: across time visualisation.

Heatmap

The heatmap visualisation allows any two information types within the OLM to be compared. Select a data type for the x axis and y axis and the heatmap matrix will be displayed. This visualisation is able to display more data at once than the others, and allows different relationships to be compared. For example, in Figure 38, the open learner model shows the different levels of competency for

information coming from each datasource. The intensity of the (red) pigmentation shows the extent of competency.

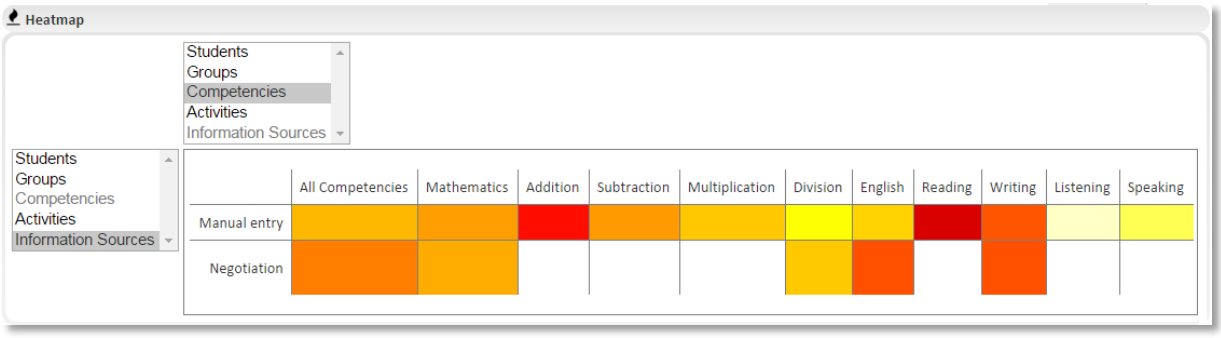
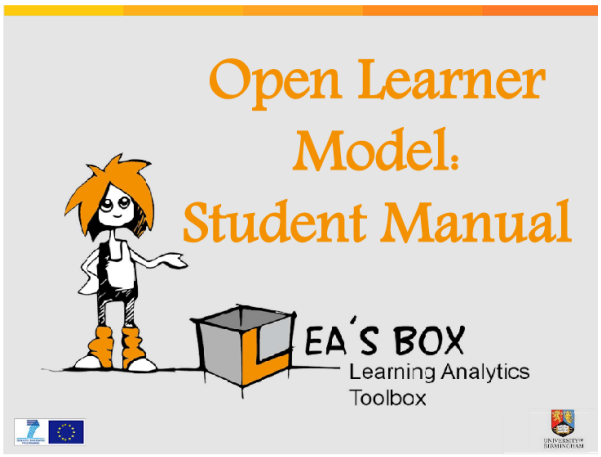
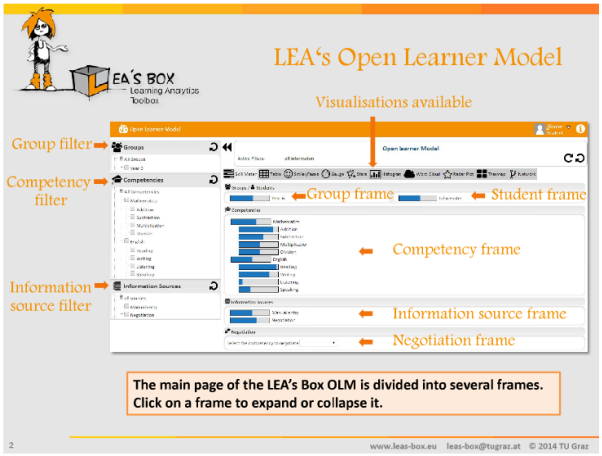


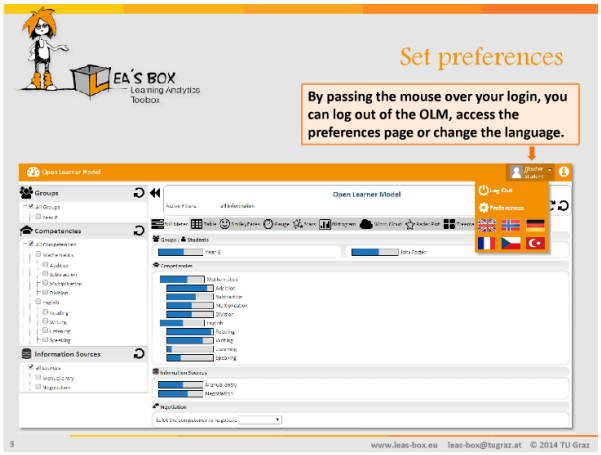
Figure 38: heatmap visualisation.

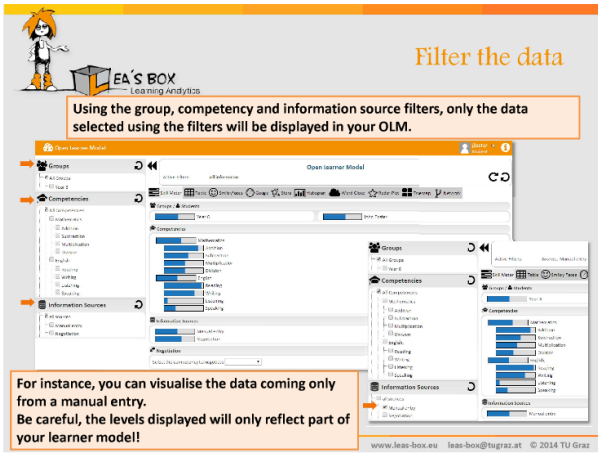
APPENDIX 2: Presentations for End Users

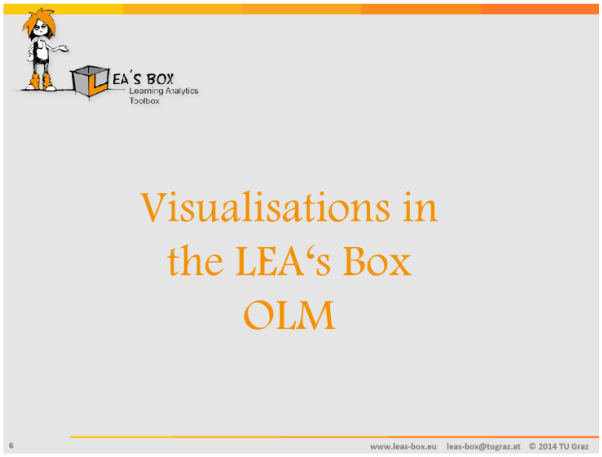
Students



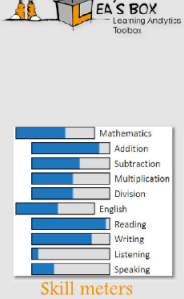








Skill meters and gauges visualisations



Skill meters

Gauges

In the skill meters and gauges visualisations, each competency is represented using a continuous value.

A competency represented by a full skill meter or a gauge with the indicator at the end of the green area, means that this competency is considered to be fully held by the student.

An empty skill meter or gauge means that there is no evidence for this competency.

Smileys and Stars visualisations

Smileys

Stars

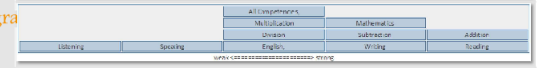
In the smileys and stars visualisations, the competencies are shown on a 5 points scale:
from very weak (a confused smiley or 1 star out of 5)
to very strong (a very happy smiley or 5 stars out of 5)

Table and histogram visualisations

Table

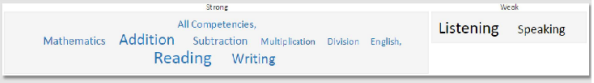
	Very Weak	Weak	OK	Strong	Very Strong
All Competencies			•		
Mathematics				•	
Addition					•
Subtraction				•	
Multiplication			•		
Division			•		
English			•		
Reading					•
Writing				•	
Listening	•				
Speaking		•			

Histogram



In the table and histogram visualisations, the competencies are evaluated on a 5 points scale from very weak (left) to very strong (right).

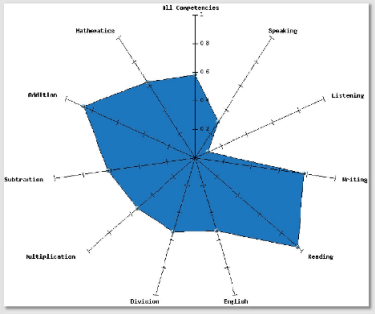
Word cloud visualisation



In the word cloud visualisations, the competencies are divided into 2 groups: the competencies with a strong level (in blue on the left) and the competencies with a weak level (in black on the right)

On the strong side, the larger a competency name, the stronger the competency.
On the weak side, the larger a competency name, the weaker the competency.

Radar plot visualisation



In the radar plot visualisation, each competency is represented on an axis of the chart.

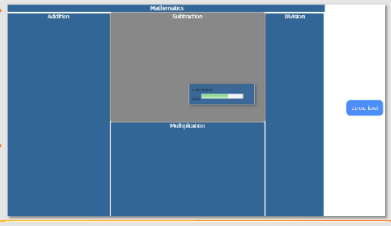
The competencies are joined to form an area representing the levels of the competencies.

Treemap visualisation

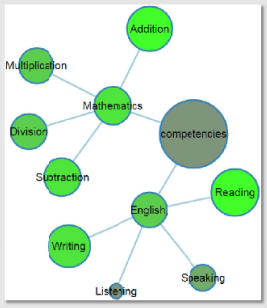
The treemap visualisation shows all competencies at a single level.
To see the sub-competencies of a competency, click on it.
To see the parent competency, click on the button "up one level" or on the parent competency.
The larger the size of the area for a competency, the stronger the competency in comparison with the other competencies at the same level.
You can also see the value of a competency by passing the mouse over it.

Parent competency "Mathematics"

Sub-competencies "Addition", "Subtraction", "Multiplication", "Division"



Network visualisation



In the network visualisation, each competency is represented by a bubble.

The larger and greener a bubble, the stronger the competency.

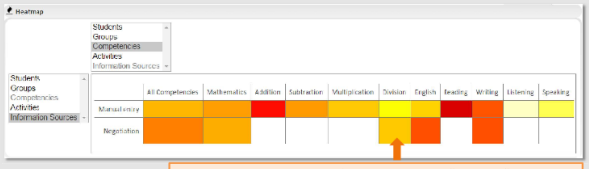
You can click on a competency to expand or collapse its sub-competencies

"Across time" visualisation

The "across time" visualisation represents the evolution of a competency across the time.
In this example, on the 27th October, the level for the competency "Addition" decreased, and the level of the competency "Subtraction" increased.

Heatmap visualisation

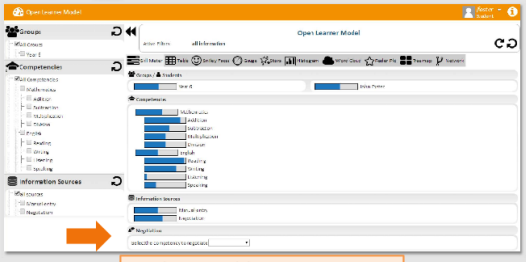
In the heatmap visualisation, you have to select the information that you want to see.
The heat map is a double entry table where a cell represent the data relative to both the column and the row headers.
The darker a cell, the stronger the data represented.
Pass the mouse over a cell to know its exact value



In this example, the level of the competency "Division" coming only from the information source "Negotiation" is 54

Negotiation in the LEA's Box OLM

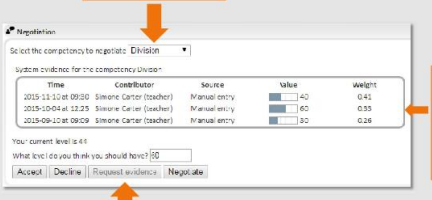
Negotiation



To make your model more accurate, you can try to persuade the system to amend your model using the negotiation feature.

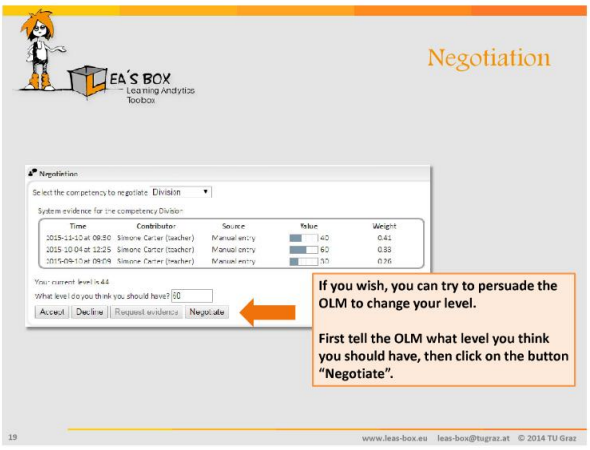
Negotiation

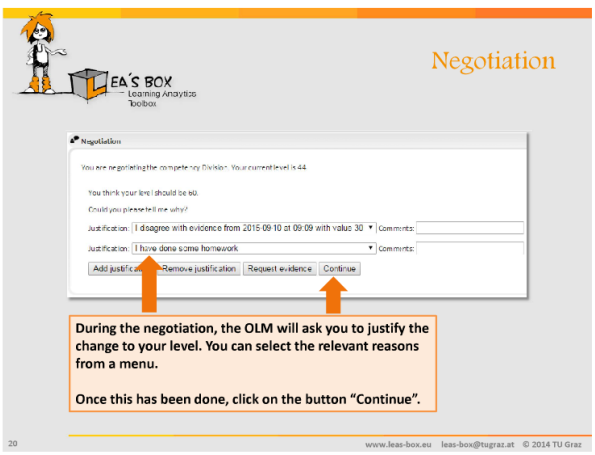
Select a competency.



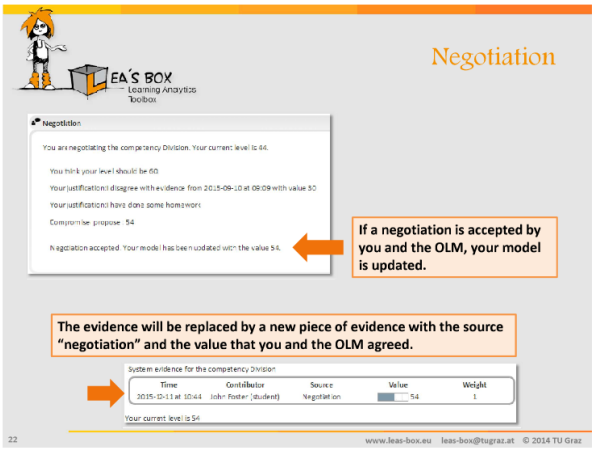
You can see the evidence for values, and how your level for this competency is calculated.

You can request evidence to see the evidence for your level.


Negotiation
If you wish, you can try to persuade the OLM to change your level.
First tell the OLM what level you think you should have, then click on the button "Negotiate".

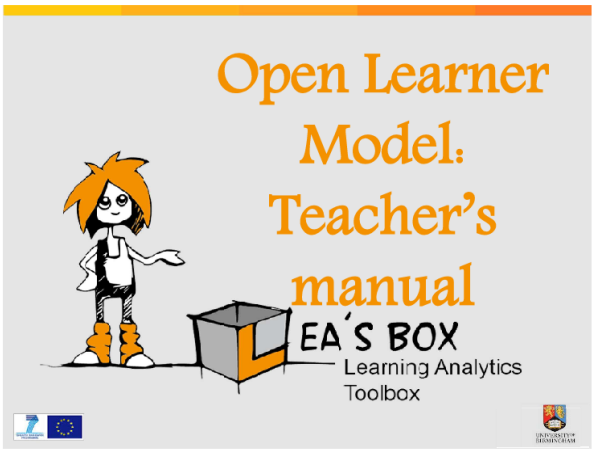

Negotiation
During the negotiation, the OLM will ask you to justify the change to your level. You can select the relevant reasons from a menu.
Once this has been done, click on the button "Continue".

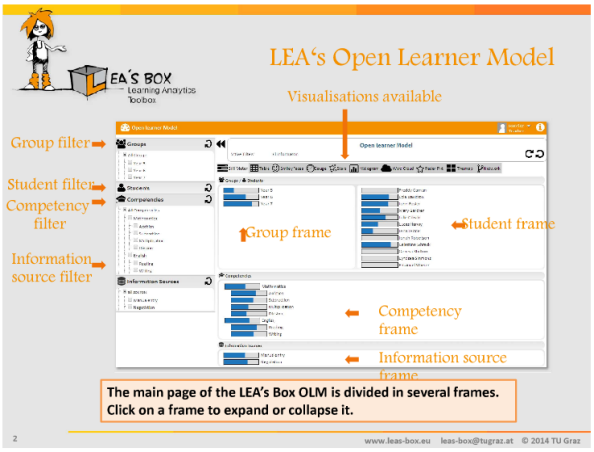

Negotiation
Depending on your justifications and on the difference between your current level and the level you think you should have, the OLM may accept your proposition of change or offer a compromise.
The OLM can also decline a negotiation if the last negotiation was too recent or if there is not enough evidence since the last negotiation of this competency.
If the OLM offers you a compromise, you can either accept it, decline it (this ends the negotiation), try to persuade the OLM of your viewpoint by adding more justifications, or offer another compromise.


Negotiation
If a negotiation is accepted by you and the OLM, your model is updated.
The evidence will be replaced by a new piece of evidence with the source "negotiation" and the value that you and the OLM agreed.

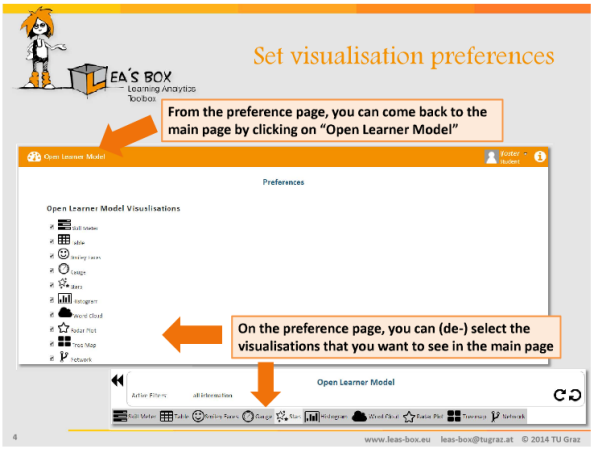

Visit me at
www.leas-box.eu

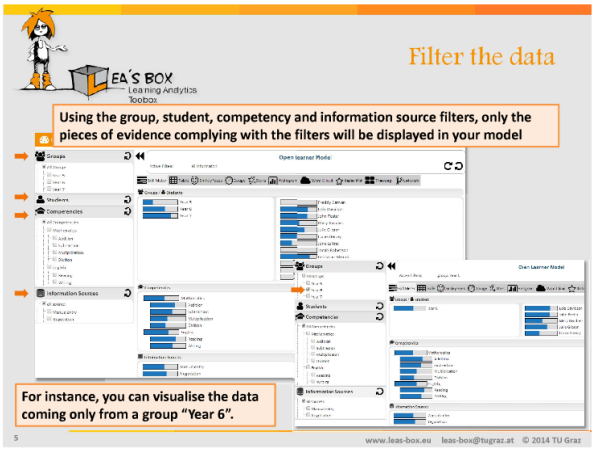
Teachers

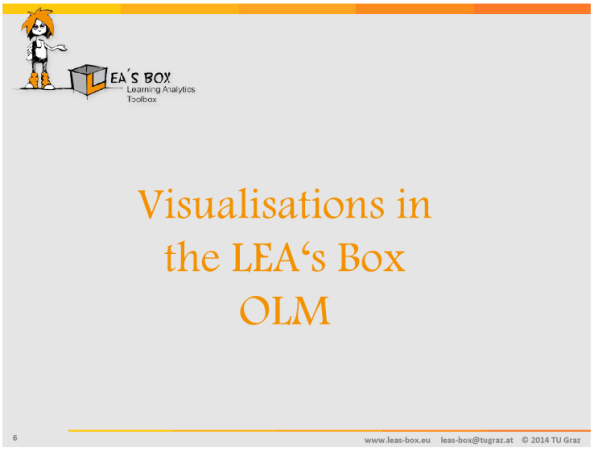

Open Learner Model: Teacher's manual
EA'S BOX
Learning Analytics Toolbox


LEA's Open Learner Model
Visualisations available
Group filter
Student filter
Competency filter
Information source filter
Group frame
Student frame
Competency frame
Information source frame
The main page of the LEA's Box OLM is divided in several frames.
Click on a frame to expand or collapse it.

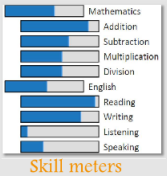

Set preferences
By passing mouse over your login, you can log out from the OLM, access to the preference page or change the language


Set visualisation preferences
From the preference page, you can come back to the main page by clicking on "Open Learner Model"
On the preference page, you can (de-) select the visualisations that you want to see in the main page


Filter the data
Using the group, student, competency and information source filters, only the pieces of evidence complying with the filters will be displayed in your model
For instance, you can visualise the data coming only from a group "Year 6".


Visualisations in the LEA's Box OLM

Skill meters and gauges visualisations



Skill meters

Gauges

In the skill meters and gauges visualisations, each competency is represented using a continuous value.

A competency represented by a full skill meter or a gauge with the indicator at the end of the green area, means that this competency is considered to be fully held by the student.

An empty skill meter or gauge means that there is no evidence for this competency.

Smileys and Stars visualisations

Smileys

Stars

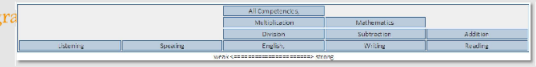
In the smileys and stars visualisations, the competencies are shown on a 5 points scale:
from very weak (a confused smiley or 1 star out of 5)
to very strong (a very happy smiley or 5 stars out of 5)

Table and histogram visualisations

Table

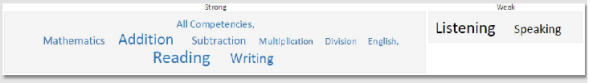
	Very Weak	Weak	Ok	Strong	Very Strong
All Competencies			•		
Mathematics				•	
Addition					•
Subtraction				•	
Multiplication			•		
Division			•		
English			•		
Reading					•
Writing				•	
Listening	•				
Speaking		•			

Histogram



In the table and histogram visualisations, the competencies are evaluated on a 5 points scale from very weak (left) to very strong (right).

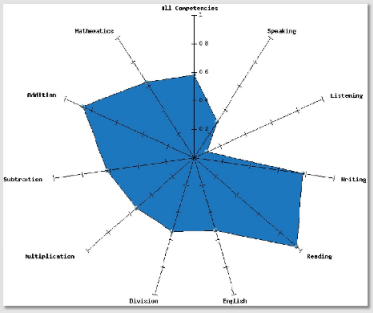
Word cloud visualisation



In the word cloud visualisations, the competencies are divided into 2 groups: the competencies with a strong level (in blue on the left) and the competencies with a weak level (in black on the right)

On the strong side, the larger a competency name, the stronger the competency.
On the weak side, the larger a competency name, the weaker the competency.

Radar plot visualisation



In the radar plot visualisation, each competency is represented on an axis of the chart.

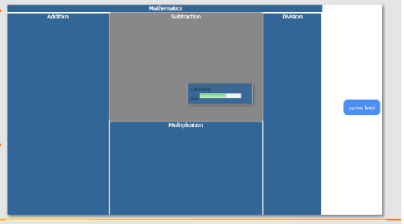
The competencies are joined to form an area representing the levels of the competencies.

Treemap visualisation

The treemap visualisation shows all competencies at a single level.
To see the sub-competencies of a competency, click on it.
To see the parent competency, click on the button "up one level" or on the parent competency.
The larger the size of the area for a competency, the stronger the competency in comparison with the other competencies at the same level.
You can also see the value of a competency by passing the mouse over it.

Parent competency "Mathematics" ➡

Sub-competencies "Addition", "Subtraction", "Multiplication", "Division" ➡



Network visualisation

In the network visualisation, each competency is represented by a bubble.

The larger and greener a bubble, the stronger the competency.

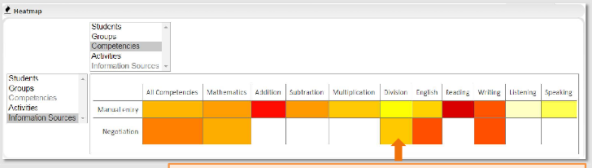
You can click on a competency to expand or collapse its sub-competencies

"Across time" visualisation

The "across time" visualisation represents the evolution of a competency across the time.
In this example, on the 27th October, the level for the competency "Addition" decreased, and the level of the competency "Subtraction" increased.

Heatmap visualisation

In the heatmap visualisation, you have to select the information that you want to see.
The heat map is a double entry table where a cell represent the data relative to both the column and the row headers.
The darker a cell, the stronger the data represented.
Pass the mouse over a cell to know its exact value



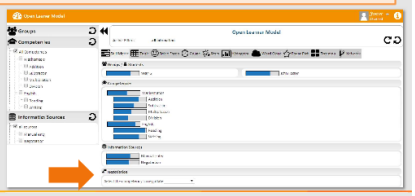
In this example, the level of the competency "Division" coming only from the information source "Negotiation" is 54

Negotiation in the LEA's Box OLM

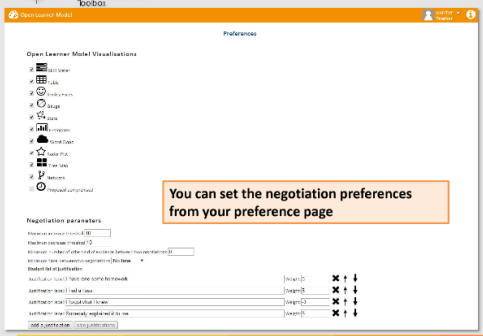
Student negotiation

The LEA's Box OLM provides learners with a negotiation feature that allow them to obtain evidence for their learner model data and try to persuade the system to make changes to their model by challenging evidence or providing justifications.
This negotiation feature aim at making the learner model more accurate, support learner reflection on their learner model contents, as well as their learning more generally, and also facilitate planning and self-monitoring.
The negotiation is initiated by the learner and parametrized by teacher

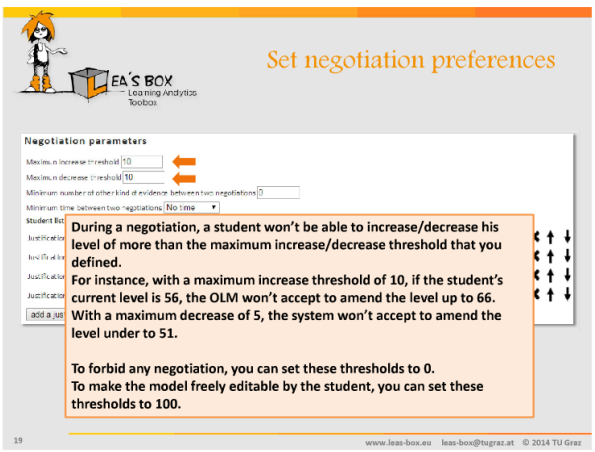
Student's main page

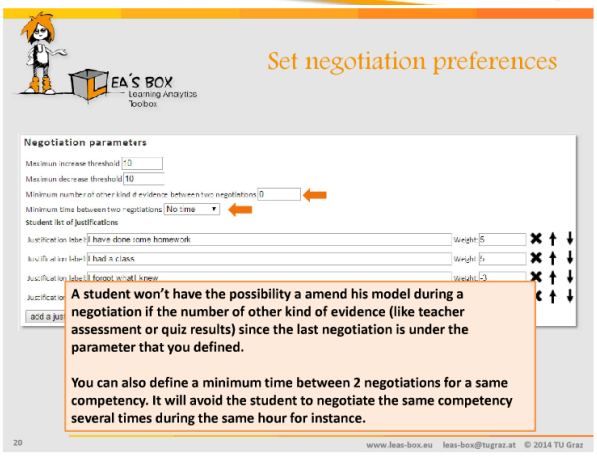


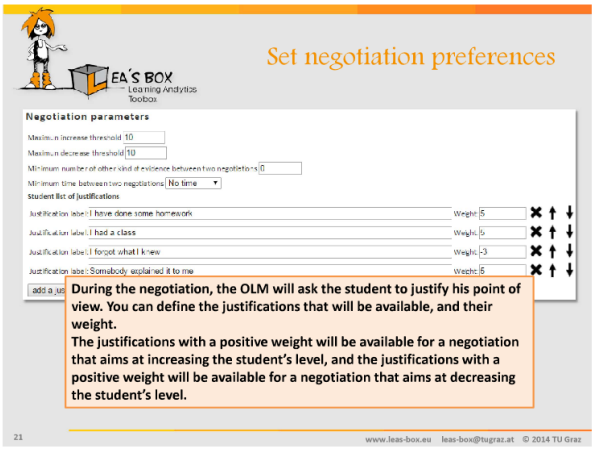
Set negotiation preferences

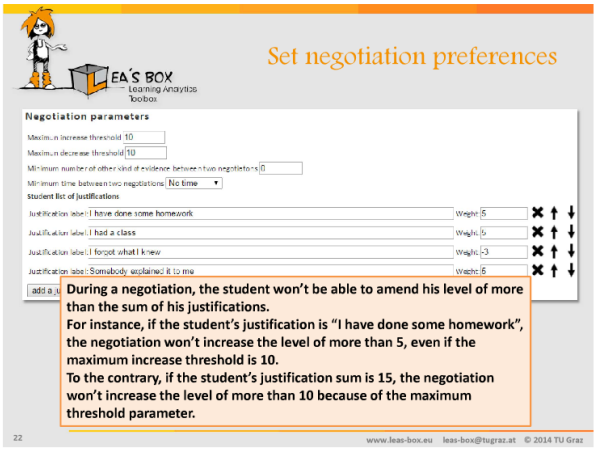


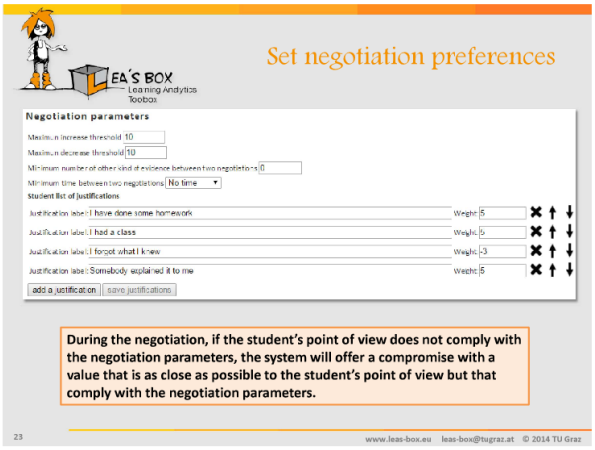
You can set the negotiation preferences from your preference page

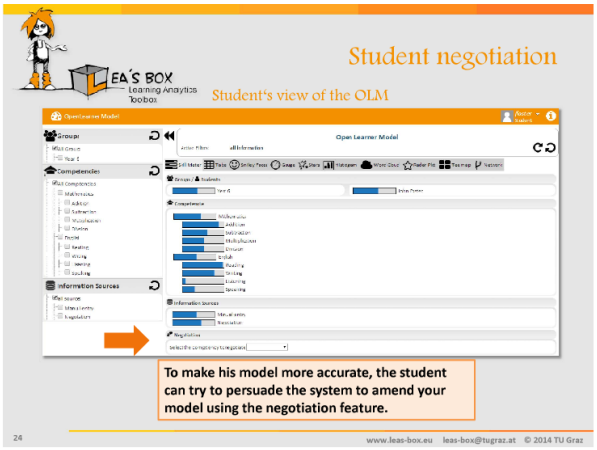

Set negotiation preferences
Negotiation parameters
During a negotiation, a student won't be able to increase/decrease his level of more than the maximum increase/decrease threshold that you defined.
For instance, with a maximum increase threshold of 10, if the student's current level is 56, the OLM won't accept to amend the level up to 66.
With a maximum decrease of 5, the system won't accept to amend the level under to 51.
To forbid any negotiation, you can set these thresholds to 0.
To make the model freely editable by the student, you can set these thresholds to 100.

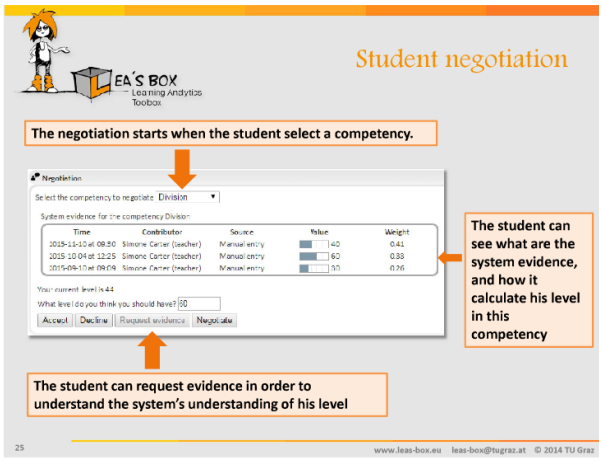

Set negotiation preferences
Negotiation parameters
A student won't have the possibility a amend his model during a negotiation if the number of other kind of evidence (like teacher assessment or quiz results) since the last negotiation is under the parameter that you defined.
You can also define a minimum time between 2 negotiations for a same competency. It will avoid the student to negotiate the same competency several times during the same hour for instance.

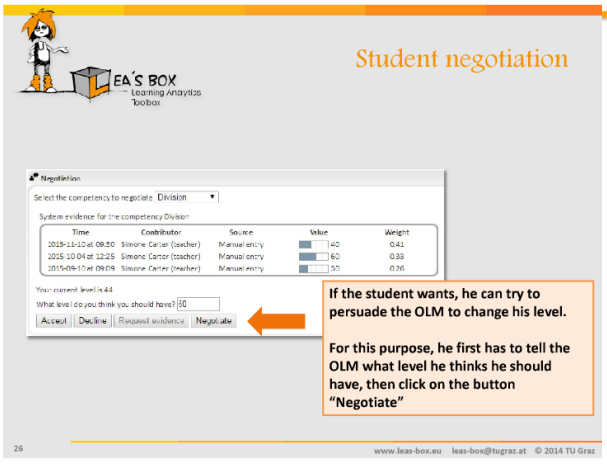

Set negotiation preferences
Negotiation parameters
During the negotiation, the OLM will ask the student to justify his point of view. You can define the justifications that will be available, and their weight.
The justifications with a positive weight will be available for a negotiation that aims at increasing the student's level, and the justifications with a positive weight will be available for a negotiation that aims at decreasing the student's level.

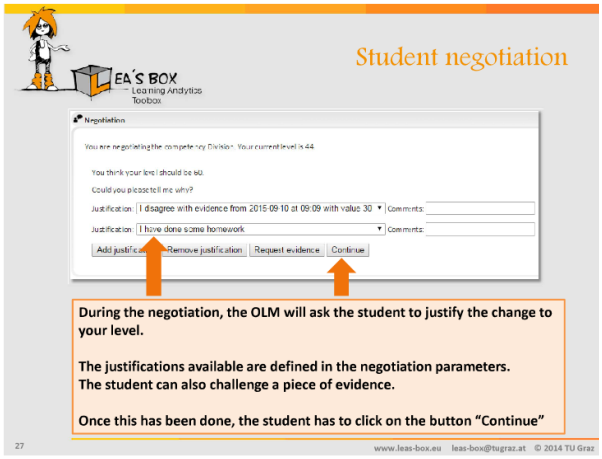

Set negotiation preferences
Negotiation parameters
During a negotiation, the student won't be able to amend his level of more than the sum of his justifications.
For instance, if the student's justification is "I have done some homework", the negotiation won't increase the level of more than 5, even if the maximum increase threshold is 10.
To the contrary, if the student's justification sum is 15, the negotiation won't increase the level of more than 10 because of the maximum threshold parameter.

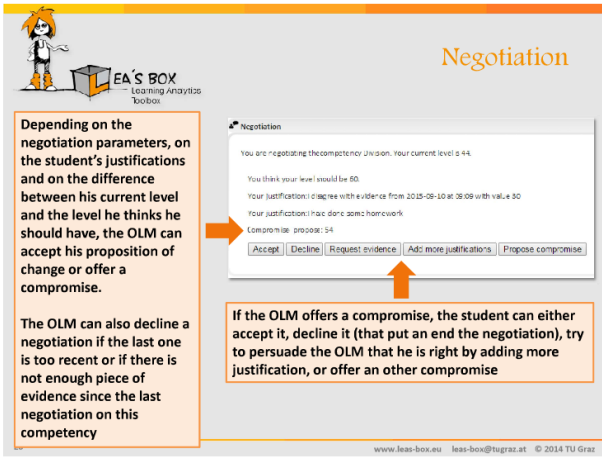

Set negotiation preferences
Negotiation parameters
During the negotiation, if the student's point of view does not comply with the negotiation parameters, the system will offer a compromise with a value that is as close as possible to the student's point of view but that comply with the negotiation parameters.

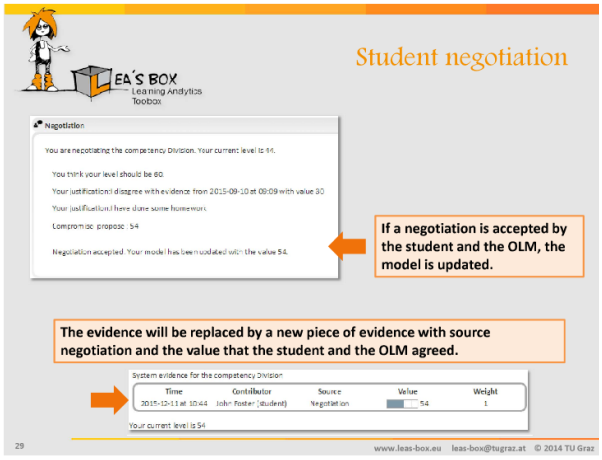

Student negotiation
Student's view of the OLM
To make his model more accurate, the student can try to persuade the system to amend your model using the negotiation feature.


Student negotiation
The negotiation starts when the student select a competency.
The student can see what are the system evidence, and how it calculate his level in this competency
The student can request evidence in order to understand the system's understanding of his level


Student negotiation
If the student wants, he can try to persuade the OLM to change his level.
For this purpose, he first has to tell the OLM what level he thinks he should have, then click on the button "Negotiate"


Student negotiation
During the negotiation, the OLM will ask the student to justify the change to your level.
The justifications available are defined in the negotiation parameters.
The student can also challenge a piece of evidence.
Once this has been done, the student has to click on the button "Continue"


Negotiation
Depending on the negotiation parameters, on the student's justifications and on the difference between his current level and the level he thinks he should have, the OLM can accept his proposition of change or offer a compromise.
The OLM can also decline a negotiation if the last one is too recent or if there is not enough piece of evidence since the last negotiation on this competency
If the OLM offers a compromise, the student can either accept it, decline it (that put an end the negotiation), try to persuade the OLM that he is right by adding more justification, or offer an other compromise


Student negotiation
If a negotiation is accepted by the student and the OLM, the model is updated.
The evidence will be replaced by a new piece of evidence with source negotiation and the value that the student and the OLM agreed.


Visit me at
www.leas-box.eu

APPENDIX 3: Full API Specification

1. LOG IN ("login")

Arguments	Description	Returns
sharedsecret leasid password	access password id number of the user user's password	On successful log in: 'LOGIN SUCCESSFUL FOR "<user name>".' Else: 'user could not be found', 'password encryption was not successful' or 'password is incorrect'.
Example http://.../leas-olm/api/login?sharedsecret=********leasid=1011&password=12345		

2. LOG OUT ("logout")

Arguments	Description	Returns
sharedsecret	access password	On successful log out: 'log out was successful.' Else: 'user is not logged in', 'log out failed'.
Example http://.../leas-olm/api/logout?sharedsecret=********		

3. ADD/UPDATE USER ("updateuser")

Arguments	Description	Returns
sharedsecret leasid password username forename surname school type	access password id number of the user user's password username of the user first name of the user last name of the user id of the user's school "student" or "teacher"	On success of adding user: 'user was added to the database: "<user name>" (id:"<user id>")'. On success of updating user: 'user was update: "<username>" (id: "<leasid>")'. Else 'Please enter the parameter "sharedsecret/leasid/password/username/forename/surname/school/type"', 'Lea's ID number ("<leasid>") must be an integer', '"<type>" was not recognised. This should either be "teacher" or "student"'.
Example http://.../leas-olm/api/updateuser?sharedsecret=********&leasid=1011&username=bbrown&forename=Bob&surname=Brown&password=12345&school=masterGroup&type=teacher		

4. DELETE USER ("deleteuser")

Arguments	Description	Returns
sharedsecret leasid method userid [override]	access password id number of the user logged in "deleteuser" Id of the user to be deleted (Optional) true	On success of adding user: 'user deleted'. Else 'user cannot be found', 'user cannot be deleted as there is data associated. Override needed', 'deleting user failed'.
Example		
http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deleteuser&userid=1		

5. ADD/UPDATE GROUP ("updategroup")

Arguments	Description	Returns
sharedsecret leasid method groupid groupname position	access password id number of the user "updategroup" id of the group name of the group the number in the order of sequence	On success of adding group: 'group created'. On success of updating group: 'group updated'. Else 'creating group failed', 'updating group failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=updategroup&groupid=75&groupname=the%20api%20updated%20this&position=1		

6. DELETE GROUP (“deletegroup”)

Arguments	Description	Returns
sharedsecret leasid method groupid [override]	access password id number of the user “deletegroup” id of the group (Optional) true	On success of adding group: ‘group deleted’. Else ‘deleting group failed’, ‘group cannot be deleted as there is data associated. Override needed.’
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deletegroup&groupid=1		

7. ADD USER TO GROUP (“addusertogroup”)

Arguments	Description	Returns
sharedsecret leasid method userid groupid	access password id number of the user logged in “addusertogroup” Id of the user to be added to the group id of the group	On success of adding user to group: ‘user added to group’. Else ‘adding user to group failed’, ‘user is already a member of the group’.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=addusertogroup&userid=1011&groupid=75		

8. Delete User from Group (“deleteuserfromgroup”)

Arguments	Description	Returns
sharedsecret leasid method userid groupid [override]	access password id number of the user logged in “deleteuserfromgroup” Id of the user to be deleted from the group id of the group (Optional) true	On success of deleting user from group: ‘user removed from group’. Else ‘user cannot be found’, ‘group cannot be found’, ‘user is not a member of this group’, ‘removing user from group failed’.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deleteuserfromgroup&userid=2&groupid=1		

9. ADD/UPDATE SUBJECT (“updatesubject”)

Arguments	Description	Returns
sharedsecret leasid method subjectid subjectname position	access password id number of the user who is logged in “updatesubject” id of the subject name of the subject the number in the sequence that the subjects are ordered by	On success of adding group: ‘subject created’. On success of updating group: ‘subject updated’. Else ‘creating subject failed’, ‘updating subject failed’.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=updatesubject&subjectid=75&subjectname=the%20api%20updated%20this&position=1		

10. DELETE SUBJECT ("deletesubject")

Arguments	Description	Returns
sharedsecret leasid method subjectid [override]	access password id number of the user "deletesubject" id of the subject (Optional) true	On success of adding subject: 'subject deleted'. Else 'deleting subject failed', 'subject cannot be deleted as there is data associated. Override needed.'
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deletesubject&subjectid=1		

11. ADD SUBJECT TO GROUP ("addsubjecttogroup")

Arguments	Description	Returns
sharedsecret leasid method subjectid groupid	access password id number of the user logged in "addsubjecttogroup" Id of the subject to be added to the group id of the group	On success of adding subject to group: 'subject added to group'. Else 'adding subject to group failed', 'subject is already in the group'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=addsubjecttogroup&subjectid=1011&groupid=75		

12. DELETE SUBJECT FROM GROUP ("deletesubjectfromgroup")

Arguments	Description	Returns
sharedsecret leasid method subjectid groupid [override]	access password id number of the user logged in "deleteuserfromgroup" Id of the subject to be deleted from the group id of the group (Optional) true	On success of deleting subject from group: 'subject removed from group'. Else 'subject cannot be found', 'group cannot be found', 'subject is not associated with this group', 'removing subject from group failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deletesubjectfromgroup&subjectid=2&groupid=1		

13. Add/Update Competency ("updatecompetency")

Arguments	Description	Returns
sharedsecret leasid method competencyid competencyname position competencyparentid subjectid competencyinfluence	access password id number of the user logged in "updatecompetency" id of the competency name of the competency the number in the order of sequence the competencyid of the parent competency the subject that the competency comes under influence of the competency (value 0 to 1)	On success of adding competency: 'competency created'. On success of updating competency: 'competency updated'. Else 'updating competency failed', 'creating competency failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=updatecompetency&competencyid=75&competencyname=the%20api%20updated%20this&position=1&competencyparentid=0&subjectid=911&competencyinfluence=5		

14. Delete Competency ("deletecompetency")

Arguments	Description	Returns
sharedsecret leasid method competencyid [override]	access password id number of the user logged in "deletecompetency" id of the competency (Optional) true	On success of deleting competency: 'competency deleted'. Else 'competency cannot be found', 'deleting competency failed', 'competency cannot be deleted as there is data associated. Override needed', 'competency cannot has sub-competencies that are associated. Override needed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deletecompetency&competencyid=1		

15. ADD/UPDATE ACTIVITY ("updateactivity")

Arguments	Description	Returns
sharedsecret leasid method activityid activityname position activityinfluence subjectid	access password id number of the user who is logged in "updateactivity" id of the activity name of the activity the number in order of sequence for the modelling procedure. (range 0-1) id of the subject the activity belongs to	On success of adding activity: 'activity created'. On success of updating activity: 'activity updated'. Else 'updating activity failed', 'creating activity failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=updateactivity&activityid=75&activityname=the%20api%20updated%20this&position=1&activityinfluence=5&subjectid=911		

16. DELETE ACTIVITY ("deleteactivity")

Arguments	Description	Returns
sharedsecret leasid method activityid [override]	access password id number of the user "deletesubject" id of the activity (Optional) true	On success of deleting activity: 'activity deleted'. Else 'activity cannot be found', 'deleting activity failed', 'activity cannot be deleted as there is data associated. Override needed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deleteactivity&activityid=1		

17. ADD COMPETENCY TO ACTIVITY ("addcompetencytoactivity")

Arguments	Description	Returns
sharedsecret leasid method competencyid activityid	access password id number of the user logged in "addcompetencytoactivity" Id of the competency id of the activity	On success of adding competency to activity: 'competency added to activity'. Else 'adding competency to activity failed', 'competency is already in the activity'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=addcompetencytoactivity&competencyid=1011&activityid=75		

18. DELETE COMPETENCY FROM ACTIVITY ("deletecompetencyfromactivity")

Arguments	Description	Returns
sharedsecret leasid method competencyid activityid [override]	access password id number of the user logged in "deletecompetencyfromactivity" Id of the competency id of the activity (Optional) true	On success of deleting competency from activity: 'competency removed from activity'. Else 'competency cannot be found', 'activity cannot be found', 'competency is not associated with activity', 'removing competency from activity failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deletecompetencyfromactivity&competencyid=2&activityid=1		

19. ADD/UPDATE DATA SOURCE ("updatedatasource")

Arguments	Description	Returns
sharedsecret leasid method datasourceid datasourcename	access password id number of the user logged in "updatedatasource" id of the datasource name of the datasource	On success of adding competency: 'datasource created'. On success of updating competency: 'datasource updated'. Else 'updating datasource failed', 'creating datasource failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=updatedatasource&datasourceid=20&datasourcename=api%20test		

20. Delete Datasource ("deletedatasource")

Arguments	Description	Returns
sharedsecret leasid method datasourceid [override]	access password id number of the user logged in "deletedatasource" id of the datasource (Optional) true	On success of deleting datasource: 'datasource deleted'. Else 'datasource cannot be found', 'deleting datasource failed', 'datasource cannot be deleted as there is data associated. Override needed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1&method=deletedatasource&datasourceid=1		

21. ADD DATA AND COMPETENCY INFORMATION ("addinformation")

Arguments	Description	Returns
sharedsecret leasid method competencyid groupid userid datasourceid value activityid	access password id number of the user logged in "addinformation" id of competency id of the group id of user to be updated id of the datasource inference value (range: 0 to 1) id of the activity	On success of adding information: 'information added: "<time of addition>" value:"<value>"'. Else 'competency does not exist in the database', 'group does not exist in the database', 'user does not exist in the database', 'datasource does not exist in the database', 'user is not a member of the group', 'user is a teacher', 'value is not a number', 'value should be in a range of 0 to 1', 'adding information failed'.
Example http://.../leas-olm/api/masterapi?sharedsecret=********&leasid=1011&method=addinformation&competencyid=198&groupid=72&userid=1001&datasourceid=20&value=0.756&activityid=911		