Deliverable D4.4

Student Model Tools R2

Identifier: NEXT-TELL-D4.4-BHAM-Student_Model_Tools_R2_v06.doc
Deliverable number: D4.4
Author(s) and company: Susan Bull, Matthew Johnson, Pavithra Ganesan, Will Byrne, Luis Hernandez Munoz, Dean Petters, Tom Duffy (BHAM)
Peter Reimann (MTO)
Ravi Vatrapu, Usman Tanveer (CBS)
Michael Kickheimer Rust (TUG)
Barbara Wasson (UniRes)
Internal reviewers: Gerhilde Meissl-Egghart (TALK)
Work package / task: WP4
Document status: Final
Confidentiality: Public
Version 2012-08-31

© NEXT-TELL consortium: all rights reserved
History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2012-08-22</td>
<td>Version for internal review</td>
</tr>
<tr>
<td>2</td>
<td>2012-08-24</td>
<td>With internal review comments</td>
</tr>
<tr>
<td>3</td>
<td>2012-08-27</td>
<td>With completed CoNeTo and RepGrid sections</td>
</tr>
<tr>
<td>4</td>
<td>2012-08-28</td>
<td>Incorporating review comments (non-CoNeTo); further review comments</td>
</tr>
<tr>
<td>5</td>
<td>2012-08-30</td>
<td>Incorporating review comments (CoNeTo)</td>
</tr>
<tr>
<td>6</td>
<td>2012-08-31</td>
<td>Final version submitted to EC</td>
</tr>
</tbody>
</table>

Impressum

Full project title: Next Generation Teaching, Education and Learning for Life
Grant Agreement No: 285114
Workpackage Leader: Susan Bull, BHAM
Project Co-ordinator: Harald Mayer, JRS
Scientific Project Leader: Peter Reimann, MTO

Acknowledgement: The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 258114.

Disclaimer: This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

This document contains material, which is the copyright of certain NEXT-TELL consortium parties, and may not be reproduced or copied without permission. All NEXT-TELL consortium parties have agreed to full publication of this document. The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the NEXT-TELL consortium as a whole, nor a certain party of the NEXT-TELL consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk, and does not accept any liability for loss or damage suffered by any person using this information.
# Table of Contents

1 Executive Summary .................................................................................................................. 1

2 Introduction ................................................................................................................................. 2  
   2.1 Purpose of this Document ..................................................................................................... 2  
   2.2 Scope of this Document ....................................................................................................... 2  
   2.3 Status of this Document ...................................................................................................... 2  
   2.4 Related Documents ........................................................................................................... 2

3 Learner Model Data Sources ...................................................................................................... 3  
   3.1 Data Sources: State of the Art ............................................................................................ 3  
   3.2 Data Sources: NEXT-TELL ............................................................................................. 4  
   3.3 Data Sources: Examples .................................................................................................... 7  
       3.3.1 Example 1: manual input of data from online or face-to-face meetings ......................... 7  
       3.3.2 Example 2: teachers transforming data using ProNIFA .............................................. 12  
       3.3.3 Example 3: rubric-based assessment ........................................................................... 13

4 Learner Model Visualisations .................................................................................................... 16  
   4.1 Visualisations: State of the art .......................................................................................... 16  
   4.2 Visualisations: NEXT-TELL ........................................................................................... 17

5 Communication and Negotiation Tools ................................................................................... 20  
   5.1 Negotiated Learner Modelling ......................................................................................... 20  
       5.1.1 Fully negotiated learner models .................................................................................. 20  
       5.1.2 Partially negotiated learner models .......................................................................... 20  
       5.1.3 Other types of learner model discussion .................................................................... 21  
   5.2 CSCL and Socio-Cultural Approach: State of the art ....................................................... 21  
   5.3 Applying Negotiated Learner Modelling in NEXT-TELL: CoNeTo ................................. 22  
       5.3.1 CoNeTo technical description ................................................................................. 25  
       5.3.2 Using CoNeTo .......................................................................................................... 28

6 Repertory Grids for Formative Assessment ............................................................................. 33  
   6.1 RGFA Technical Description .......................................................................................... 33  
   6.2 RGFA New Features ....................................................................................................... 35

7 Shared Infrastructure for OLM and Activity Capture ............................................................... 41

8 Work in Progress ....................................................................................................................... 44  
   8.1 OLM ........................................................................................................................... 44  
   8.2 RGFA .......................................................................................................................... 46

9 Summary .................................................................................................................................. 47

10 References ............................................................................................................................... 48

11 Glossary ................................................................................................................................... 52
1 Executive Summary

This document accompanies the release of the second versions of the Open Learner Model, Communication and Negotiation, and Repertory Grid for Formative Assessment tools, originally released and described in D4.2. The D4.4 tools take into account the revisions and extensions to the tools as described in D4.3; results from teacher workshops; and some of the state-of-the-art research that has been recently developing, that was not all publicly available at the time of writing D4.3. Our second release also includes experiences from the relevant NEXT-TELL partners during development, and feedback from the reviewers at the first annual review.

The open learner model’s capacity to link to other project tools is facilitated through APIs, and its role within the overall NEXT-TELL architecture more clearly formulated. Much of the data coming through the NEXT-TELL components may ultimately end up contributing to the open learner model (demonstrated more fully in the next release), while additionally, and a focus of this deliverable, can come directly from users - which includes input based on automated activities (e.g. Open Sim data transformed by the teacher using ProNIFA). Thus, while the e-portfolio has come further to the forefront (see D3.4), the open learner model remains central particularly as a tool for teachers.

The link between the requirements for negotiating the learner model and the communication and negotiation tool has been clarified, and these are now linked through an API. The communication and negotiation tool can be used throughout the NEXT-TELL infrastructure, but we focus here on examples for negotiating the learner model.

The repertory grid for formative assessment tool has been updated based on the results of participatory design workshops (reported in D2.8), and its role as a potential source of information for teachers to add further data to the learner model, explained.
2 Introduction

2.1 Purpose of this Document
This document reports on the ‘Student Model Tools R2’

2.2 Scope of this Document
This document covers the following aspects of the workpackage 4, month 24 deliverable:
- Learner model data sources
- Learner model visualisation
- Communication and negotiation tool
- Repertory grid for formative assessment

2.3 Status of this Document
Final version.

2.4 Related Documents
Before reading this document it is recommended to be familiar with the following documents:
- D4.1 Methods and Specifications for the Student Model V1
- D4.2 Student Model Tools R2
- D4.3 Methods and Specifications for the Student Model V2

Documents which complement this document include:
- D2.4 ECAAD Tools R2
- D2.8 Report on Classroom Research with STEM and TESL Assessment
- D3.4 Activity Capturing Tools R2
3 Learner Model Data Sources

This deliverable focuses on the open learner model (OLM) component of Figure 1 (lower right), which is based on data running through the various other NEXT-TELL components, to the OLM; or data coming directly into the OLM through some other automated or manual route.

3.1 Data Sources: State of the Art

Russell (2010) highlights the extent to which the amount, detail and speed with which learner data can currently be collected and summarised, enables greater opportunities for timely teacher interventions and data-rich evidence-based decision-making. This relates to cognitive density as described by Crawford et al. (2008) in terms of student engagement and teacher decision-making, applicable in an OLM context (see D4.3; Bull et al., 2012). Open learner models are now being designed to take into account potentially diverse data (Mazzola & Mazza, 2010; Morales et al., 2009). Recent work proposes combining e-portfolios and independent open learner models to provide data for other learner models (e.g. in standalone adaptive military training), thereby involving learner modelling across multiple applications (Raybourn & Regan, 2011). Cruces et al. (2010) provide a tool to integrate and edit models, supplemented with a separate open learner model to interact with other learner models based on different learning resources. Their approach is therefore a generic one. Dolog and Schaefer (2005) introduce a framework for exchanging learner profiles between various sources, including the evidence for the learner model data to allow another system to interpret its meaning appropriately. Although the NEXT-TELL approach does not provide information for, or transfer learner model information across external systems in this way, issues such as those in related methods are also important here (e.g. evidence for incoming data, potential inconsistencies and priority of information from different sources). Thus, these are amongst our starting points.
3.2 Data Sources: NEXT-TELL

Data sources for the NEXT-TELL OLM currently include direct teacher input (e.g. following a discussion with a student, after marking work, or resulting from negotiation with a student). By ‘direct input’ we mean that the data may have come straight from a person into the OLM, or may have come via a teacher, in an artifact or process appraisal as shown in Figure 1, between the e-portfolio and OLM.

Automated data from technologies are also shown in Figure 1. These include direct assessments such as Moodle quizzes, activity in OpenSim, Google docs, spreadsheets, social networks, OpenSim, e-portfolio, self-assessment/evaluation forms, and other computer-assisted learning or adaptive environments. Data might also be transformed by the teacher using the NEXT-TELL ProNIFA tool. Combinations of input to the learner model will depend on teachers’ choices of activities for their students. The learner model can accept a range of automatically generated inferences from data sources in xml format, as part of its API. For example:

```xml
<?xml version="1.0"?>
<add_to_model>
  <userid>tmaki</userid>
  <usertype>student</usertype>
  <contributorid></contributorid>
  <contributorname></contributorname>
  <evidencesource>OLMlets_inference</evidencesource>
  <evidenceurl></evidenceurl>
  <tags></tags>
  <activityid>1027</activityid>
  <competencyid>645</competencyid>
  <knowledgelevel>0.7</knowledgelevel>
  <strength></strength>
  <guidence></guidence>
</add_to_model>
```

An example from the current version of the software, of learner model data from teacher input, is shown in Figure 2. Assessments of student understanding or competencies may be tendered to the model, associated with defined specific student activities (which may be electronic or paper-based), and associated with competencies. This is soon to be extended to student self-assessment, peer-assessment and parent input (e.g. according to activities undertaken outside school), with the teacher having an optional administrative role to approve what information should be allowed to enter the learner model.
In addition to numeric feedback that also contributes to the learner model (see star input in Figure 2), input may be as textual feedback (student strengths, and guidance or areas for improvement). Textual feedback is not transformed by the modelling process, but is available for inspection, supporting the outcomes of the modelling process. It is designed also to help in any student-teacher negotiation of the model (perhaps using CoNeTo - see Section 3). Model data may also be supported by the inclusion of a hyperlink to evidence supporting a specific assessment, for example, to a learning-based artefact stored in the e-portfolio, Moodle, or a Google document. These are links to the artefacts in the evidence layer. Drilling down through the model information will ultimately arrive at these artefacts. An API is under development to also allow automatic linking of items of data from other pieces of technology, such as OLMlets (Bull et al., 2010) and Moodle.

Currently, by default, the most recent data from the various sources has higher weighting, according to the following algorithm:

\[
\text{new\_value} = \text{new\_data} \times \text{depreciation\_factor} + \text{old\_value} \times (1 - \text{depreciation\_factor})
\]

This is a fundamental aspect of the modelling process. Clicking on an item in the OLM interface displays a narrative of information qualifying how the model value has been derived (shown in Figure 3). This adds a level
D4.4
Student Model Tools R2

Figure 3: Evidence for how learner model information was derived

of transparency for the user to see the modelling process. At present this is numeric, demonstrating proof-of-concept - the data is available and can be easily accessed. Work is underway to makethis interface more graphical and readily interpretable by users.

It is possible for teachers to prioritise data from different sources according to their own requirements in or outside the classroom. Parameters are available, set initially to defaults (equal weighting). An interface is currently under development to allow teachers to manipulate these parameters. When activities, units of work, subjects and competencies are combined, these influence which parameters are taken into account. When information from multiple students is combined, all students receive an equal weighting; no one student can be set to be more influential than any other.

This allows, for example, an English teacher in Norway to specify that an OpenSim activity indicating that beginner level learners using an avatar can follow simple spatial instructions has greater weighting in the learner model than information from related questions in a multiple choice quiz. Similarly, in a 21st Century skills and collaboration context, a student facilitating a meeting might be expected to have more frequent interactions during the meeting (e.g. to move between agenda items), than some other meeting participants. This role is consequently likely to require different competency weightings than other roles. On-going work throughout the project is considering the extent to which the various data sources should contribute to representations of specific competencies in our example subject areas, to provide recommendations for teachers should they wish to have advice on overriding the default in any case(s).
3.3 Data Sources: Examples

We here provide examples to show how data can enter the learner model from different sources, in addition to that given as our illustrative example in Figure 2.

3.3.1 Example 1: manual input of data from online or face-to-face meetings

Because it is a challenging task for a teacher to micromanage group work (with or without ICT) in a classroom with many students, software such as LAMS (http://www.lamsfoundation.org/index.htm) has been developed to mitigate teachers’ cognitive load. In our approach, we aim to go beyond activity tracing (e.g. as implemented with LAMS) by providing the teacher with visual information on students’ on-going learning on the competency level. Our approach also makes intentionally more use of the students themselves as a resource for managing collaboration: by putting a few students in the role of meeting facilitators, the teacher has fewer activity management tasks to deal with and can concentrate on the overall classroom process, rather than individual groups. We consider this to be a contribution to the challenges of ‘classroom orchestration’ (Dillenbourg & Fischer, 2007). At the same time, students are provided with authentic opportunities to practice preparing and running on-line and face-to-face meetings, a competency that is valuable both inside and outside schools; especially since meetings can take up substantial time in the workplace (Romano & Nunamaker, 2001). We describe below, aspects of the meeting process: planning, facilitation, and documentation and communication of outcomes in relation to OLM competencies. (We propose that, to validly provide authentic opportunities to students, our competencies match those described for good “real world” meetings generally. Because of the OLM-competency relationship, the detail is presented in this deliverable. However, it applies across several deliverables.

Planning

The meeting process begins when an individual determines that a meeting is required. Francisco (2007) created a short (8 question) yes/no ‘Should You Meet?’ checklist that can be used for the identification of this. For example, “Can you state the purpose of your meeting?”; “Do you have the information you need to meet productively?” (Francisco, 2007). Seibold (1979) further emphasises the importance of identifying the specific purpose(s) of the meeting and delineating a range of goals, to create the basic structure of the meeting. The next stage is to decide on the group composition for the meeting, to ensure that all those affected, are present (Seibold, 1979). The tasks involved in this – including allocation of roles and responsibilities - can be seen to form their own competency below. In addition, all members should be briefed on the points above through an agenda, allowing individual feedback prior to the meeting (Seibold, 1979). Weston (2009) also emphasises the importance of group participation before the meeting starts, with a series of steps to improve preparations before a meeting. This includes introducing complex issues at one meeting and deferring discussion and questions for the next meeting; and the importance of supplying written materials to participants in sufficient time prior to the meeting. Participants should then read the material in advance, and perhaps have informal discussions beforehand. Updating the agenda may result before a meeting (Macaulay & Alabdulkarim, 2005).

Kaner (2007) suggests that an agenda should comprise three basic points: (i) the topics to be discussed; (ii) the desired outcomes for each topic; and (iii) processes needed to achieve the desired outcomes. A variety of activities may be used in a meeting (Kaner, 2007) which can be teamed with relevant level of involvement and time estimates in order to specify the processes needed to achieve the desired outcomes. For clarity, desired outcomes should be split into: the overall goal for the topic (what final result do we want to achieve in order to be finished with this topic); and the meeting goal (what narrowly defined, specific objective do we want to achieve for this topic at an upcoming meeting?). Kaner (2007) states that these goals do not necessarily have to be written in the agenda, but should be explicitly stated during the course of the meeting - thus there is overlap with the next section (meetings and facilitation).

Once the composition of the meeting has been decided, all appropriate group roles should be delineated, responsibilities assigned and authority delegated where necessary (Seibold, 1979).

Jonker et al (2007) suggest that at least two key roles of chairman and secretary are necessary when inspecting individuals in a meeting scenario.
Meetings and Facilitation

A meeting facilitator may effectively set the frame by describing: the task, the outcome, the process, the rationale for the process, and the expected amount of time required (Kaner, 2007). Data resulting from group discussion or brainstorming may need to be sorted using pre-defined criteria or creating categories, and the list may need to be reduced through prioritising items.

To conduct a meeting in a meaningful way, it is necessary to balance creative and critical thinking to productively support discussion and decisions (Francisco, 2007). A facilitator might use different types of intervention strategy if problems develop during a meeting, for example: interpretation (shifting focus to the process, describing, inviting discussion); direct action (interrupting meeting flow e.g. preventing interruption, encouraging an individual) (Viller, 1991).

A meeting exit survey may be used to evaluate a meeting, for example: "How well did we use the time allotted?", "How well thought-out were our decisions?" (Francisco, 2007). Exit survey questions can also address the skills of individual participants, such as "How effective was the facilitator?" Indeed, questions similar to these have been used in consultancy, in relation to evaluating the cost-effectiveness of meetings, such as: “The meeting leader...runs meetings effectively”, “...listens carefully and actively”, “…creates an environment where people are comfortable disagreeing” (Rogelberg et al., 2012).

Documentation and Communication of Outcomes

While a meeting must be documented by capturing minutes and noting actions (Francisco, 2007), this need for a record is often overlooked by students and, indeed, some textbooks do not adequately cover all types of minutes in different settings (Wolfe, 2006). Thus, it is particularly important for teachers to ensure students’ awareness of this requirement. This explains our distinction of competencies related to these issues, as a third category, despite their apparent relative simplicity.

Technology Support for NEXT-TELL

Different technologies can provide different levels of system support to meeting facilitation, e.g. from complete automation with no facilitator function, to simply providing support for recording and reporting information (Macaulay & Alabdulkarim, 2005). As NEXT-TELL is designed for use in a range of subjects and settings, we concentrate on generic tasks.

In order to provide teachers with information on students’ competency development in an automated manner, students may plan their meeting using the Planner (middle left of Figure 1). This allows comparison of the meeting as planned with the meeting as conducted. Planning a meeting consists of sequencing a number of group activities, which are represented as meetlets. Meetlets combine the description of a series of steps (e.g. the steps necessary to have a group perform a brainstorming activity) with a specification of the tools and artifacts with which to conduct the activity. For instance, for a brainstorming activity this could be a (collaboratively edited) Google Spreadsheet document. Meetlets also contain information about how to evaluate the success of the activity; e.g. in a brainstorming meetlet, this could be the number of ideas generated (further explanation of meetlets is given in D3.4). This information is used to update the competency model of the facilitator and/or group members.
Figure 4: Meetlet structure

Figure 4 depicts how information in the meetlet structure is used to drive a specific meeting activity and to appraise/assess an activity. For the case considered here, the meeting is conducted online. The descriptions of the sub-steps of an activity (for instance, for a brainstorming activity this may include eliciting and combining individual ideas) are interpreted by the Activity Stepper (lower left of Figure 1) that guides the team members through these steps, and then rules describing how the resulting artefacts (e.g., individual and collective idea lists) are to be appraised, are applied to the artefacts. This artifact appraisal information is then used to update the learner model.

We are considering simple appraisal rules that build on information directly available in the artefacts. For example, for brainstorming, the number of individual ideas, collective ideas, and the ratio between them, can be used to formulate appraisal rules. More advanced rules could calculate the semantic overlap between ideas generated individually and those proposed as a group solution, but are at present not implemented. Our focus is currently on how to represent knowledge that is typically formulated by teachers in rubrics in a way that can (in principle) be interpreted by machines.

Based on the above literature about meeting structure and facilitation, the following competencies (and sub-competencies) to be modelled and displayed in the open learner model have been defined, for (i) planning; (ii) facilitation; and (iii) documentation and communication of outcomes of meetings.

### Planning Meetings

1. Determine whether a meeting is necessary
   - identification of clear purpose
   - identification of set of goals
   - setting sufficient duration

2. Determine group roles and responsibilities
   - inclusion of all relevant stakeholders
   - identification of roles of secretary and chair
   - correct allocation of roles and responsibilities
3. Creation of agenda
- agenda content: topics, desired outcomes, processes to achieve outcomes;
- circulation of agenda prior to meeting
- sufficient time allowed for participant feedback

Meeting Facilitation
1. Setting the frame
- task introduction
- statement of outcome
- process
- rationale
- duration

2. Intervention strategies (as appropriate)
- interpretation (shifting focus to the process), describing, inviting discussion)
- direct action (preventing interruption, encouraging an individual)

3. Sorting data (as appropriate)
- pre-defined criteria
- creating categories
- prioritising items

Documentation and Communication of Meeting Outcomes
1. Documentation
- capturing minutes
- noting actions and deadlines

2. Communicating outcomes
- distributing minutes
- passing on information to/discussion with relevant others (identified in actions)

Kaner (2007) distinguishes a number of meeting purposes, from information distribution to creating commitment for decisions, and a different engagement levels for group activities. Both relate to agenda planning in the sense that the meeting activities should match the meeting purpose with respect to activity level. Hence, looking at an agenda, one can make judgements as to how appropriate the planned activities are vis-a-vis the meeting purpose in terms of the (average) engagement level. In order to perform this kind of appraisal, one needs to have each meeting activity in the agenda indexed by its engagement level. This will be our upcoming focus for further development of the meeting competencies in the project (and hence, to be represented in the learner model).

Manual input to competencies in the learner model may come from peers (ongoing) and the students themselves, provided as in Figure 5, where numerical assessment is provided, and additional (non-modelled) comments are possible.
These competencies and others form part of a default set in the learner model (which can be added to by teachers), including support for competencies in German (for IT skills) and Norwegian (for English as a Second Language) - see Figure 6. Further information about the English and IT skills competencies is given in D4.3.

Figure 6: Teacher view of/access to competencies database

OLM
Available at http://eeevle.bham.ac.uk/nexttell-cas/
Example Login (teacher): user : “bbrown”; password “12345”
Example Login (student): user : “aadams”; password “12345”
3.3.2 Example 2: teachers transforming data using ProNIFA

Similar to Example 1 in terms of people inputting information directly into the learner model is the case of teachers using ProNIFA to transform data to make it more suitable for the learner model, as much data is not readily available in competence form.

We illustrate with a set of activities aimed at 11-12 year-olds, including an electronic reading and listening test and interactions in a virtual world (OpenSim). Assessment methods (automatic and manual) are applied to data from these activities to determine achievement level for relevant competencies. Each activity provides an example of classroom data. The first, the online CLIL listening and reading test, has a mix of item types: multiple choice, click item, click text, click name, click word, move paragraph. The test is described in more detail in D2.8. Each item is weighted according to difficulty by professional test developers and these weights, along with student answers and other test item information, is used by ProNIFA (see D2.4), an automatic assessment method, to generate competence levels for students taking the English reading and listening tests before data is passed to the OLM.

The second data set derives from activity within OpenSim and includes chat logs and video recordings of activity in 3D space. For example, from OpenSim we get (i) a simple chat log file (time stamp, chatting person/entity, chat text); (ii) a set of competencies (CEFR skills shown here), specified in a text file (number, id, initial probability that students have that skill, short description); and educator-defined (scripted) rules, which vary from very simple such as checking whether a certain entity writes a certain text; to more complicated, such as computing distances travelled in the virtual environment.

(i) [07:21 UTC] <b><i>Teacher</i></b>Well done, Karen.<br>

(ii) 001 CEFR#094 0.5 Listening A1

(iii) [Rule1]
Who=Teacher
What=Well done, <NAME>.
ASkills=1,2
AUpdate=0.2
LSkills=3
LUpdate=0.1 (If the teacher says "Well done" and a name, the probabilities of skills 1 and 2 for learner <NAME> are increased by 0.2; and for skill 3, decreased by 0.1.)

ProNIFA parses the log files, checks whether the rules apply and updates the probabilities of the competencies (and the probability distribution over the competence states). Further detail about data format is given in D2.8. The result can be displayed to teachers (e.g. in bar chart or table form), as illustrated in Figure 7, for them to transfer to the OLM (as in Figure 2). (The next tool release will also allow some level of automatic input from ProNIFA to the OLM.)
Figure 7: Displaying probability distribution to teachers

**ProNIFA**
The ProNIFA installation package is available from the NEXT-TELL tools website:
http://sandbox.next-tell.eu

### 3.3.3 Example 3: rubric-based assessment

This section focuses on the kind of assessment that is frequently performed in classrooms, but does not refer to problem solving steps, and does not rely on a trace of single steps (as they would be captured in a log file, for instance). Typical examples are teachers’ appraisal of artefacts (essays, sketches, designs, multi-media products) and complex performances (such as giving a presentation, playing an instrument). Such types of assessment are not covered by methods such as ProNIFA, that require a sequence of answers to individual test item solutions or problem solving steps on a fine-grained level of detail.

Teachers often use checklists and rubrics for these cases. Both can be represented as tables. A checklist is usually applied to an artefact (such as a document) or to an observation to establish that either of these has certain features. The result of a applying a checklist is usually a Pass/Fail decision (has/does not have all features required.) But it can also be an incremental score, which brings it closer to a rubric. In terms of competencies, a single checklist usually would inform only one competency, if any. But this does not necessarily need to be the case. A checklist can be applied to a complex artefact (e.g., a meeting agenda document), and can inform multiple competencies (for instance, agenda planning skills and word processing skills).

#### CHECKLISTS

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Value if present</th>
<th>Value if not present</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>(usually 1)</td>
<td>(usually 0, but can be negative when not-compensable)</td>
</tr>
<tr>
<td>A2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A(l)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Score: (x out of max)</td>
<td>Score-Part1:</td>
<td>Score-Part2:</td>
</tr>
</tbody>
</table>
For more differentiated and possible multi-dimensional assessment, rubrics are used. If a rubric has just one row, it is kind of a checklist with a finer granularity of judgement than just Fail/Pass. But usually, a rubric has multiple rows (dimensions of judgement). A rubric typically has the following general form:

**RUBRICS**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Lowest level (e.g., 1)</th>
<th>...</th>
<th>...</th>
<th>Highest level (e.g., 5)</th>
<th>Optional Weighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>(Description of evidence for cell value)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A(l)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Students can also benefit from rubrics, as they are likely to more readily understand the reasons for their assessment scores when they are also given the rubrics (Barney et al., 2012). This is particularly important in formative assessment contexts such as ours.

In addition to descriptive suggestions for rubric design (e.g. Mertler, 2001), there are many tools that help to create checklists and rubrics (such as rubistar.4teachers.com) as well as sites that provide ready-made rubrics (eg., http://www.lauriefowler.com/rubrics.html). However, for a teacher it would be unusual to think about how the appraisal of an artefact or of a performance relates to a competency. Either the artefact/performance is appraised, and maybe a mark is given for it, or in some cases the appraisal may be related to a standard. Standards are similar to competencies, but they are not necessarily psychologically motivated; many standards are pure content standards (“students masters x”).

Since we want to model assessment rather than provide a specific instance of a rubric (or rubric generator), and since we want to be able to support the assessment automatically where possible, the rubric generator type of software is of limited use to us (note that rubric generators are just that: they help with the generation of the rubric (table), but do not support the application of the rubric to the student’s artefacts). Existing rubric sites are of great use, however, because they can help us, and end-users, to decide which dimensions should go into an appraisal/assessment. But for (partially) automated assessment, a more foundational solution is needed.

To express rubrics in rule form, we would have in each cell a rule that says, logically: “If x and/or y and/or z (…) then classify artefact/performance into this cell”, where x, y and z correspond to aspects of the student-created artefact or performance. Hence, a fairly natural way to represent rubrics in NEXT-TELL could be decision tables. They can express rule logic and their tabular format make them a “natural” representation for end users, in particular teachers who use rubrics. (As exhibited above, checklists and rubrics are readily depicted as tables.) Decision tables have the added advantage that they can serve as the basis for declarative programming: If well-formed, they can be automatically transformed into executable code. A more abstract, and more general notion that has been derived from decision table is decision model, with foundations and applications described in Von Halle & Goldberg (2010).

Formulating decision tables is essentially all that the teacher and/or the assessment expert (which may be a teacher in some cases) need to do to describe an assessment. In order to turn this assessment model into an executable method, a modelling/programming expert would need to come in to relate the variables mentioned in the decision tables to actual data.
In conclusion, using decision models/tables as the tool to express the logic for artefact assessment, we can distribute the design of the assessment model/method over three roles:


2. Assessment expert (amongst them, specially trained teachers): Relation to and updating of learner (competency) model; also formulated in decision tables (in cases where teaching/learning standards are relevant for teachers, this may be expressed as an update of the extent to which a student masters a standard; in this case, the teacher would also be involved in formulating the Learner Model updating rules, since the teacher will be knowledgeable about the standards.)

3. Technical expert: Relating the decision models (tables) to data sources & channelling the decision results to the right receivers (learner model, human users).
4 Learner Model Visualisations

This section updates the state-of-the-art in learner model visualisation including work published since the last deliverable, and shows how NEXT-TELL is drawing on this in the choice of visualisations.

4.1 Visualisations: State of the art

Various OLM presentation examples have been described in the literature for university students (see Bull & Kay 2010, for a more detailed overview). The most common visualisations used in courses include skill meters (Bull et al, 2010a; Mitrovic & Martin, 2007; Weber & Brusilovsky, 2001); concept maps (Mabbott & Bull, 2006; Perez-Marin et al., 2007); and hierarchical tree structures (Conejo et al., 2011; Kay, 1997; Mabbott & Bull, 2006). Recently, tree map overview-zoom-filter approaches to open learner modelling have also started to appear (Bakalov et al., 2011; Kump et al., 2012). The range of potential (human and technology-gathered) data sources demands methods of learner model externalisation that can either be adapted according to the specific data sources, or methods that are sufficiently generic to be applicable in the range of cases. As previously argued for generic open learner model contexts (Albert et al., 2010), we take the latter approach in NEXT-TELL.

The student OLM in NEXT-TELL may be used with learners having a range of ages. At this stage at least, we provide a single set of visualisation options across all students; and a single (partly overlapping) set for teachers. This is in part to enable a NEXT-TELL OLM that can be taken up quickly, according to the classes available and wishing to use the OLM; and in part to accord with studies suggesting preferences for simpler learner model views: there does not (yet) appear to be a general need for more detailed views for more advanced (and older) users – though much research still remains to be undertaken. This will be part of our research contribution in later stages of the project.

To illustrate the above: a recent eye-tracking study of university students trying to understand four learner model presentations - kiviat chart, concept tag cloud, tree map and concept hierarchy) - found the kiviat graph and concept hierarchy to be more efficient (for understanding the representation), than the tree map and tag cloud (Mathews et al., 2012). Furthermore, the kiviat chart was considered to be the best format through which to gain a quick overview of knowledge; although other views were useful if further detail was required. Based on these results, Mathews et al (2012) conclude that the most useful visualisation is likely to depend on the context for which it is being used. Another OLM eye-tracking study found that, when concept map, pre-requisite map, tree structure of concepts, tree structure following lecture topics and sub-topics, and alphabetical index of the same underlying model were compared, visual attention in a view depended on whether the view was amongst the user’s preferred views, rather than on the structure of the view itself (Bull et al., 2007). Albert et al. (2010) state that university students thought simpler views were understandable and suitable for gaining an overview of their learning; while a more complex activity visualisation was less popular, perhaps because the complexity of the information was difficult to understand. Duan et al (2010) found preferences towards skill meters over more complex visualisations; however, users with more complex views of the learner model perceived it to be more accurate. Also, although a majority, the preference for the simpler skill meters was still only 53%. Ahmad and Bull (2008) also found that students perceived more detailed views of the learner model to be more accurate than skill meters. However, users still reported a higher level of trust in the more simple skill meter overviews. University students with experience with a range of different OLM systems have indicated their preference for having both overviews and detailed learner model information available for viewing (Bull, 2012).

Girard and Johnson (2008) aim to extend considerations about the presentation of OLMs such as described above, to school level. To date, at that level most attempts have been very simple in presentation, a common example being smiles (e.g. Bull & McKay, 2004; Kerly & Bull, 2008; Tsinakos, 2010), or proficiency indicated by colour on the zoomable Khan Academy exercise dashboard (Khan Academy, 2012), similar to link annotation in adaptive navigation support in adaptive educational hypermedia (Brusilovsky, 2001). More complex structures have also been explored, for instance the externalisation of a Bayesian network, in the context of supported
interaction with human or artificial partners (Zapata-Rivera & Greer, 2004). The NEXT-TELL OLM is broader and for more flexible use, and so a less complex approach is considered more appropriate.

Much HCI visualisation work follows Shneiderman’s (1996) recommendation for the overview, zoom, filter and details-on-demand approach. This allows drilling down to further information, as required. As seen above, some OLMs are now trying tree map approaches (building on Shneiderman (1992)), to achieve this (Bakalov et al., 2011; Kump et al., 2012). However, Mathews et al’s (2012) results suggest that tree maps may not be as easily understood – at least until a learner is more familiar with the representation. Thus, as well as taking direction from the current state-of-the-art, in the next phase of the project NEXT-TELL aims to add to the state-of-the-art for use of OLMs by students and teachers.

4.2 Visualisations: NEXT-TELL

In the absence of a range of studies of OLM views at school level, coupled with the evidence to date for simple views in some contexts, we follow the suggestions from the literature at university level that simpler views may be generally understandable. At this stage it is not clear which will be the most appropriate approach in our setting, but overview first, detail second, is a starting point for further investigations - the requirement for overviews has been met; the potential need for details included, but proposed in a more clearly structured form than typical tree maps, with which some users may struggle. A reduced set of the initial OLM views (reported in D4.2) are included in the present system. These are summarised in Figure 8. The smiley metaphor, skill meter, table and histogram views have been retained as the most readily interpretable of the set of OLM representations. Holding the mouse over items displays their numeric value, and clicking on them presents a more detailed breakdown of their sub-components that contribute to the stated value.

Figure 8: OLM visualisations

Further to these implemented views we propose the use of a tree map and word clouds to complement this set. The tree map approach (shown in Figure 9) shows all competencies within a selection at their top level, and clicking on an item in the map breaks the section down into its subcomponents for example (Meeting Facilitation, broken down into Setting the Frame, which is broken down into Statement of Outcomes, Task Introduction etc.) The magnitude of the tree map section is representative of the level of knowledge gained in the area, and the colour indicates the recency of the information. While not necessarily the most useful way of...
viewing information for quick response, this view is especially suited to displaying larger amounts of information in a small space, through the drill-down mechanism.

Figure 9: Treemap

In contrast, the left of Figure 10 shows a word cloud (blue text); and right – the reverse (i.e. competencies not yet demonstrated) (red text). These displays are currently under development, for use on-the-spot by teachers to provide feedback to students, and to offer feedforward (guidance) as to which meeting facilitation competencies to develop further.

Figure 10: Word clouds

In contrast, the left of Figure 10 shows a word cloud (blue text); and right – the reverse (i.e. competencies not yet demonstrated) (red text). These displays are currently under development, for use on-the-spot by teachers to provide feedback to students, and to offer feedforward (guidance) as to which meeting facilitation competencies to develop further.

The granularity of display can be determined by the teacher (e.g. in skill meters, “creating an agenda” could be further split into Kaner’s (2007) three points: topics to be discussed; desired outcomes for each topic; processes necessary to achieve the outcomes. Teachers can use the visualisation(s) that best fit their purpose or preferences at the time. For example, for a quick, on-the-spot decision about where a group needs help, “setting the frame” clearly stands out as needing improvement on the right of the word cloud visualisation in Figure 10. In contrast, the left of Figure 10 indicates that a meeting is probably already well-planned, and perhaps the next phase should now be considered.

We retain multiple views for each user based on previous findings of different views suiting different users (Bull et al., 2010b), also in line with the approach of current systems using multiple views displaying similar learner model information (e.g. Bull et al., 2010a; Conejo et al., 2011; Duan et al, 2010; Mazzola & Mazza, 2010). Each view is optionally available to the user (student or teacher), allowing them to choose a custom combination of the visual representation to have in juxtaposition. As stated above, because of the clarity of overview information shown in simple views, we place these in the foreground for users – both students and teachers, but also provide additional detail about the composition of competencies, as well as evidence for the competencies (and sub-competencies) shown. Currently this is in text form as in Figure 3. We aim to use the results of other research now ongoing in this area (as described above), to help inform our own more detailed visualisations, alongside our own studies of web-based prototype versions of the more complex, or expanded/drilled down OLM displays. The results of this research will be included in tool release 3.

A revised version of the OLM browser facility is shown in Figure 11. It is possible to change the scope of any of the information parameters using the filter mechanism at the head of the page, to support enquiry of different scopes and levels of granularity. Items in the OLM, feedback or interaction tabs below will automatically update. The ability to search by time period will be the next feature to be introduced.
Figure 11: OLM browser

OLM
Available at [http://eeevle.bham.ac.uk/nexttell-cas/](http://eeevle.bham.ac.uk/nexttell-cas/)
Example Login (teacher): user: “bbrown”; password “12345”
Example Login (student): user: “aadams”; password “12345”
5 Communication and Negotiation Tools

The CoNeTo communication and negotiation tool is available for negotiation of several areas of NEXT-TELL (see Figure 1). The basic processes are similar in each case, but the actual uses differ. We illustrate with the case of negotiating the learner model.

5.1 Negotiated Learner Modelling

Key features of negotiated learner models are not only that the presentation of the learner model must be understandable by the user, but also that the aim of the interactive learner modelling should be an agreed model. Most negotiated learner models are negotiated between the student and tutoring system. However, other stakeholders can also be involved, and the notion of “the system” can be broadened to include a range of technologies such as used in NEXT-TELL. In this section we consider (i) fully negotiated learner models; (ii) partially negotiated learner models; and (iii) other types of learner model discussion. These are all relevant to our notion of negotiating about the learner model, or negotiating its content, performed with CoNeTo.

5.1.1 Fully negotiated learner models

Mr Collins was the first learner model designed for negotiation with the student (Bull & Pain, 1995). Its focus was on increasing model accuracy by student-system discussion of the model, while at the same time promoting learner reflection through discussion. The model contained separate belief measures: the system’s inferences about the student’s understanding, and the student’s own confidence in their knowledge (that they input along with their responses to questions). Mr Collins used menu-based discussion to allow students to challenge and respond to the system at any time, and allowed the system to initiate discussion of the learner model if there were discrepancies between the system’s inferences and the student’s stated confidence in their knowledge. This follows the notion of interaction symmetry – i.e. the student and system have the same negotiation moves available (Baker, 1990). These included initiating discussion, maintaining discussion and ending discussion; and allowed each party (student and system) to request explanations, challenge beliefs, justify viewpoints, amend beliefs, accept compromise, maintain beliefs (i.e. if student and system cannot agree, both inconsistent belief measures would be retained). Adult learners did challenge Mr Collins when they disagreed with their learner model, and did suggest changes to it. This indicated the potential for further research on the implementation of negotiated learner models.

STyLE-OLM (Dimitrova, 2003) used a dialogue game based model in negotiation, with the following dialogue moves (adapted from Pilkington (1999)), similar to the above: inform, inquire, challenge, disagree, justify, agree, suggest, skip. It could change dialogue tactics flexibly using its dialogue model. Initial findings provided additional support for the potential to promote learner reflection in university students using negotiated learner modelling.

Based on the negotiation options of Mr Collins (Bull & Pain, 1995), CALMsystem further developed the notion of negotiating the learner model using natural language with a chatbot (Kerly & Bull, 2008). An evaluation with children aged 10-11 revealed significant improvements in self-assessment and reduction of discrepancies in the learner model, amongst the negotiated learner model condition.

5.1.2 Partially negotiated learner models

Close to the above definition of negotiated learner models is xOLM (Van Labeke et al., 2007). Based on Toulmin’s argumentation model (Toulmin, 1959, cited in Van Labeke et al., 2007), xOLM uses: data (actual belief); claims (summary belief - e.g. level I, level II); warrants (evidence for beliefs); and backings (qualitative and quantitative attributes supporting warrants). However, xOLM relies on the student to initiate discussion of the model. For example, students can challenge claims, warrants and backings, and receive justifications from the system. Learners can choose to agree, disagree, or move on (without resolution). New evidence is added to the learner model based on this, which could subsequently be explored by the learner. The system allows the learner’s challenge to succeed where there is unresolved disagreement (Van Labeke et al., 2007).
In contrast, EI-OSM defers the overriding decision to the (human) teacher if interaction between a student and teacher cannot resolve discrepancies using the system’s evidence-based argument approach (Zapata-Rivera et al., 2007), also based on Toulmin. In addition to the use of data, claims, warrants and backings, EI-OSM also incorporates rebuttal and rebuttal data. There were mixed reactions from teachers as to whether they would consider assessment claims from students without the availability of relevant evidence, but they believed that these could form a useful starting point for formative dialogue (Zapata-Rivera et al., 2007).

Also relevant here are OLMs that are persuadable. The main difference between these and fully negotiated learner models are that, as with xOLM (Van Labeke et al., 2007) and EI-OSM (Zapata-Rivera et al., 2007), models that can be persuaded do not follow the principle of each partner having the same moves available, or matching roles in diagnosis. A system will have to agree before any changes can be effected in the learner model, and this situation occurs if the challenge comes from the student. For example, EER-Tutor is an intelligent tutoring system with a separate component allowing students to challenge concepts in the learner model (Thomson & Mitrovic, 2010). A student can choose to initiate a dialogue with the system at any time. The system offers a question, and the model is updated according to the correctness of the answer given (multiple choice or short answer). Flexi-OLM also has multiple choice and short answer questions, and allows users to challenge their learner model (Mabbot & Bull, 2006). Flexi-OLM provides evidence for the model contents in the form of students’ responses that led to the inferences about their understanding, to provide a resource for reflection. Students can proceed to persuasion after considering the evidence, and similar to EER-Tutor, can demonstrate their knowledge (or lack thereof) by answering additional questions posed by the system.

### 5.1.3 Other types of learner model discussion

While not a negotiated learner model, OLMlets (Bull et al., 2010b) was used with a Facebook group to allow university students to discuss their models with each other (Alotaibi & Bull, 2012), indicating a willingness to critically consider understanding in an open-ended way. This is crucial for methods of model negotiation between human partners where open discussion is encouraged. A similar approach in the sense that the model itself is not negotiated, is allowing children to provide their assessments of their knowledge to the system if they disagree with it, quantitatively and explained in text comments which can be seen by the teacher. Such input can become a focus for subsequent (human) teacher-child discussion (Zapata-Rivera & Greer, 2004).

Research into design of student-system collaborative assessment found that university-level participants acting as (human) teacher-student pairs would challenge an assessment and resolve disagreements in a manner resembling that used in negotiated learner modelling (Pain et al., 1996). While originally intended as a study on the feasibility of collaborative assessment between student and system, it also provides data to support the idea of student-teacher negotiation of the learner model. Indeed, subsequent work addressed issues in negotiated collaborative assessment focused on negotiating the model between assessee (student) and assessor (system or teacher) (Brna et al., 1999), raising questions such as: assessment criteria; reasons for the criteria; degree to which the student can challenge criteria; evidence to collect during interaction; sources of material available for consultation; ground rules of negotiation; how ground rules are chosen and communicated; extent to which a student can influence the outcome of negotiated assessment; degree to which a student learns during negotiated assessment (Brna et al., 1999). These questions are still relevant, and perhaps need even more careful consideration when the learner model is based on multiple sources of data, as the requirements may not be consistent across cases. For example, sources of data may differ not only in relevance, but also their accessibility, granularity, validity and reliability (Zapata-Rivera et al., 2007).

### 5.2 CSCL and Socio-Cultural Approach: State of the art

As mentioned previously in the DoW and the D4.2 Annex, the Communication and Negotiation Layer is informed by the Socio-Cultural approach and implements the current state-of-the-art in argumentation systems in Computer Supported Collaborative Learning (CSCL). CSCL research subscribes to an “intersubjective” epistemology (Suthers, 2006) that emphasizes the importance of designing efficient and effective socio-cultural processes between the collaborating learners. The emphasis on the intersubjective processes is justified in terms of the conceptual work and empirical findings in the socio-cultural tradition in the social sciences.
epitomized by the ongoing work in Soviet Psychology (Wertsch, 1985). Constructionism (Harel & Papert, 1991), Communities of Practice (Lave & Wenger, 1991), Cultural Historical Activity Theory (Roth & Lee, 2007), and recent Ethnomethodology and Conservation Analysis inspired work on distributed interaction (Suthers et al., 2007) and group cognition (Stahl, 2006).

The theoretical debate on the relationship between interactional processes and individual agencies (Sawyer, 2002) in the Socio-Cultural approach remains a source of productive tension when it comes to CSCL in particular and the design of argumentation systems in particular. The tension is productive as it opens up a design space in terms of designing computational support for both knowledge construction and knowledge discourse. One example of this productive tension in the design of CSCL argumentation systems is the “representational guidance” line of work (Tooth et al., 2002; Suthers, 2008) that explores the research and development space at the intersection of ontologies for argumentation (hypothesis, data, for, against etc), the notations for the chosen ontology, and the level of integration of collaborative discourse. The design rationale for the Communication and Negotiation Tool (CoNeTO) is informed by the extant literature on CSCL argumentation research (Andriessen et al., 2003, the “Collaborative Representations” project that extends the Representational Guidance research program (Suthers, 2008; Vatrapu & Suthers, 2009), and a recent review of computer supported argumentation (Scheuer et al., 2010).

5.3 Applying Negotiated Learner Modelling in NEXT-TELL: CoNeTo

As stated previously, a primary benefit of negotiated or persuaded learner models is considered to be their ability to increase the accuracy of the model data while also prompting learner reflection. Some of the main themes in negotiating a learner model, as identified in the approaches described above, include: identical negotiation moves (in fully negotiated learner models); evidence for the model data; objects or artifacts of discussion / for consultation; ability to challenge the model; learning through negotiation; control over the resulting negotiated model representations. To some extent, these will depend on the philosophies of the particular educational setting, associated tools and teacher choices. By definition, negotiation requires the availability of identical negotiation, argumentation or dialogue moves and rights, otherwise it is not a true negotiation. However, as seen above, there are also reasons to use partially negotiated learner models. We therefore consider all cases as relevant here.

For negotiation to have meaning, there must be evidence to support arguments. This may take various forms, from the work or artifact produced by a learner, to system (or NEXT-TELL OLM) explanation of its inferences and their sources (currently as shown in Figure 3). Evidence could be simple, such as a student claiming to have completed an activity. This may not be acceptable to teachers in a formal assessment situation, though it might be beneficial as a focus for teacher-student discussion (Zapata-Rivera et al., 2007). Simple cases such as a learner claiming to have forgotten, might be more immediately usable. In contrast, evidence could in principle be based on more complex reasoning such as approaches using Toulmin's argument structure, that use (some of) data, claims, warrants, backings, rebuttal, and rebuttal data (Van Labeke et al., 2007; Zapata-Rivera et al., 2007), but this is beyond the scope of the simpler approach designed for easy use by stakeholders in NEXT-TELL.

Much evidence will likely point towards objects or artifacts. Such artifacts could be an essay (where some form of simple statistical information can be drawn quite easily), a teacher appraisal, outcome of a quiz on a specific topic, a spreadsheet calculation demonstrating a skill, logs of avatar activity in a virtual world (perhaps also with video), wiki, blog or discussion entries (perhaps with associated peer appraisals - judgements of helpfulness, etc.), and so on. Again, as stated above, this will depend on the specific activities chosen by the teacher.

Challenge, as already seen from the above, is a crucial component of being able to negotiate a learner model, regardless of the extent to which the discussion is classed as a full negotiation (i.e. with equal rights and moves available to all negotiating parties). If a student cannot challenge the learner model content, there can be no negotiation of it. Challenge is commonly also intended to promote learner reflection and encourage metacognitive activities (Bull & Kay, 2010). This leads to the possibility that a student might learn while negotiating their model. The model will need to update accordingly. This raises the question of how such updates should be represented. If negotiation around data from a specific application results in learning related
to that application, a narrowly focused competency representation could be entered to the learner model. If such discussion could also apply in other contexts, it becomes a question of how broadly to apply the newly demonstrated learning. In NEXT-TELL this is in the hands of the teacher.

It is also necessary to consider which party has control over the outcome of negotiation. The examples above have shown that this can be the student (Van Labeke et al., 2007); system (Thomson & Mitrovic, 2010); separate and equally valid representations for each party (Bull & Pain, 1995); and teacher (in the case of student-teacher negotiation) (Zapata-Rivera et al., 2007).

Negotiating the learner model may be even more important in contexts such as NEXT-TELL, where there are multiple data sources contributing to the learner model, as students can easily lose track of which activities have contributed to their learner model; may not appreciate the relative importance of activities in terms of weighting in the model (e.g. due to recency or type of data available to the model); or may not recognise that so many data sources are taken into account. Offering the student the opportunity to negotiate the model content through discussion with a teacher aims to help them recognise this. This may result in them subsequently agreeing with the representations or, if not, providing them with the information they need to form a useful argument. OLM visualisations and related activity data and evidence (e.g. e-portfolio contents) form the objects or artifacts under discussion. Figure 12 illustrates the structure of the learner model components.

As seen from the examples in Section 1.3, various activities may contribute data to the model. These activities will often also produce the artifacts mentioned above. As in Figure 12, this information can be added manually to the learner model, and ongoing work is allowing automated updates from a range of software/activities (spreadsheet use, quiz, virtual world, etc.). In negotiation, the teacher will need to consider issues such as the assessment criteria, evidence, materials, extent of student influence over the model, how to take account of student learning during negotiation, and the ground rules for negotiation (as identified by Brna et al., 1999). Either directly with the student, or through the communication and negotiation tool, the teacher can then enter the model negotiation process with the student. Figure 13 presents the CoNeTo interface with the OLM data from the Group Meeting Activity (Example 1 in Section 3.3.1).
We describe below the update to the design and development of CoNeTo, the communication and negotiation layer tool for NEXT-TELL. The re-design takes into account results from a participatory design workshop, heuristic usability evaluation and reviewers’ comments to the first version presented in D4.2 annex.

As described in D4.2 Annex, the central idea of the Communication and Negotiation Tool (CoNeTo) is to provide computer supported collaborative argumentation of the linkages between student activities, knowledge inferences, and OLM visualizations of students’ activities and knowledge inferences. In other words, what is being communicated and made available for negotiation is the information about the student’s current OLM representation (knowledge/competency inference) and their activities (from activity capture and tracking). The negotiation tool embodies the “artefact-centered discussion” design principle (Suthers et al., 2008). The artefacts are the OLM visualizations of the student knowledge levels, activities, and their visualizations. The following three usage scenarios and use cases are provided as examples, with point 1 of “during instruction” (OLM visualisations) being the central example we have used in this deliverable.

Three usage scenarios were described in D4.2 Annex. Additional usage scenarios are discussed based on reviewers’ comments, the design critique sessions with Danish teachers participating in the NEXT-TELL project and participatory design workshop with university college teacher. The usage scenarios are organized into pre-instruction, during-instruction, and post-instruction phases of a course module.

Pre-Instruction

- CoNeTo is used by teachers to engage the students in a discussion of the relationship between planned teaching activities and intended development of student competencies. This use case builds on the modelling activities of teachers with the NEXT-TELL ECAAD planner and activity stepper and opens up teachers’ models for student discussions.

- CoNeTo is used by students to construct individual and/or group learning activities and knowledge artifacts. This use case seeks to foster reflective practices of students and allows the possibility of including open educational resources and learning objects from the informal learning ecosystem into the teaching mix.
During Instruction

- CoNeTo is used by students to negotiate the current status of learning activities and OLM representations of competencies.
- CoNeTo is used by students to discuss possibilities of alternate learning activities, assessment forms, and/or learning trajectories.
- CoNeTo is used by teachers to engage the classroom on the current status of the course module, activity progress, and competence development.

Post-Instruction

- CoNeTo is used by teachers to identify the teaching improvisations and knowledge discourses and reflect on them.
- CoNeTo is used by both teachers and students to identify more capable peers in the zone of proximal development.

CoNeTo is a web-based application that is built using HTML5, JavaScript and ASP.Net. JavaScript and HTML5 are responsible for the frontend i.e. drawing and interaction, whereas ASP.Net provides server side functionality such as converting XML from and to the OLM format, and saving and loading diagrams. Figure 14 presents the technical architecture of CoNeTo.

Figure 14: Technical Architecture of CoNeTo

The frontend is designed using HTML5 and JavaScript. The basic shape drawing and the interaction with the drawing i.e. clicking, dragging etc. is being done using a JavaScript library Kinetic.js. Each basic Kinetic shape (a
rectangle representing Activity and an oval representing Competency) is contained in a JavaScript object called Node.

Each Node stores various properties pertaining to that particular node, i.e. Description, Details, Position, URL and Chat. Each node displayed in the diagram displays its name, whereas all the other properties are displayed in the Properties Panel whenever the shape is selected, by clicking on it.

Different nodes can have any of the three possible relationships (‘For’, ‘Against’, or a ‘Unspecified’ between them and represented as a JavaScript object called Relationship. Each relationship is represented in the diagram by a straight line with a circle in the centre, displaying either a ‘+’ (For) ‘−’ (Against), or a ‘?’ (Unspecified).

A Diagram consists of a list of Nodes and Relationships. Besides, it also contains some other properties such as the student email, teacher email and the topic of the discussion that is being represented by the diagram in question.

A sample JSON object representing a Diagram object is represented below.

```json
{
    "index":3,
    "studentName":"demo",
    "teacherName":"",
    "course":"",
    "topic":"",
    "nodes":[
        {
            "nodeId":0,
            "nodeType":1,
            "name":"Activity 1",
            "details":"This is Activity 1",
            "url":"",
            "messages":[],
            "position":{
                "x":100,
                "y":50
            }
        },
        {
            "nodeId":1,
            "nodeType":2,
            "name":"Competency 1",
            "details":"This is Competency 1",
            "url":"",
            "messages":[],
            "position":{
                "x":328,
                "y":52
            }
        }
    ],
    "relationships":[
        {
            "relationshipType":2,
            "firstNodeId":1,
            "secondNodeId":0
        }
    ]
}
```

The backend is coded using ASP.Net. The backend provides basic authentication (via Nexttell) and archiving functionality i.e. saving and loading diagrams.

The data is transferred back and forth between the frontend and backend using JSON. When the diagram is to be saved, the data is sent to the server as a JSON string, which is then serialized in XML format. Then the application’s native XML format is converted to the OLM XML format and saved on the server. Similarly, when a diagram is to be loaded from the server, it is deserialized from the OLM XML format and then converted to application’s native XML format and finally into a JSON string and returned back to the frontend, where it is parsed and restored as the currently active diagram.

© NEXT-TELL consortium: all rights reserved
The native XML model of CoNeTo is different than the OLM XML model, therefore an explicit conversion takes place when loading an OLM model in order to render it in the Coneto application. A code snippet that is used in order to achieve this is displayed below. Code documentation is provided as inline comments.

```csharp
public Diagram OLM2Diagram(OLM olm) {
    if (olm != null) {
        var diagram = new Diagram { course = olm.scope_class, topic = olm.scope_subject, teacherEmail = olm.scope_teacher };
        int index = 0;
        var nodes = new List<Node>();
        var relationships = new List<Relationship>();

        // Add activity
        nodes.Add(new Node
            { name = olm.activity.activityname, // olm.scope_activity,
              details = HtmlUtility.HtmlDecode(olm.activity.narrative),
              nodeId = Convert.ToInt32(olm.activity.activityid),
              nodeId = 1,
              position = new Position {x = 50, y = 50} });

        // Add competencies
        foreach (var competency in olm.activity.competencies) {
            nodes.Add(new Node
                { name = competency.competencyname,
                  details = HtmlUtility.HtmlDecode(competency.narrative),
                  nodeId = Convert.ToInt32(competency.competencyid),
                  nodeType = 2,
                  position = new Position {x = 300, y = 50 + 75*index++},
                  url = competency.visualisation });
            relationships.Add(new Relationship
                { firstNodeId = Convert.ToInt32(olm.activity.activityid),
                  knowledgelevel = Convert.ToDouble(competency.knowledgelevel, new CultureInfo("en-us")),
                  relationshipType = Convert.ToDouble(competency.knowledgelevel, new CultureInfo("en-us")) > 0.6 ? 1 : 2,
                  secondNodeId = Convert.ToInt32(competency.competencyid) });
        }

        diagram.nodes = nodes.ToArray();
        diagram.relationships = relationships.ToArray();
        diagram.index = diagram.nodes.Max(n => n.nodeId);
        return diagram;
    }
```
return null;
}

The above code converts the OLM XML to CoNeTo XML. For a valid provided OLM XML, a corresponding Diagram object is created. The main properties of the OLM artefacts are then copied into the corresponding properties represented in the Diagram object. An arbitrary graph canvas position is assigned to the nodes, as position information is not available in the OLM XML. Finally, for each of the competencies related to the activity in question, a corresponding Competency node and a relationship node is created. The type of the relationship is determined by the knowledge-level defined within each competency. In the Group Meeting example presented in Figure 13, if the probability value is greater than 0.6, a relationship represented by a ‘+’ (For) sign is created between the nodes. However, for a value less than or equal to 0.6, a relationship is represented by a ‘-’(Against) sign. All these nodes and relationships are then appended to the Diagram object and displayed. In the future, these values can be determined automatically by the OLM system and also manually by the teachers when necessary.

Several suggestions were made by the university college teachers during the usability assessment exercise of CoNeTo during the participatory design workshop (for example, increase in canvas size, re-organization of the panels, re-design of the buttons, and highlighting of nodes upon selection). The final release of the version 2 of CoNeTo incorporated the usability suggestions from the participatory design workshop and a heuristic usability evaluation. The change request of changing the node title from “Knowledge/Competency” to “Goal” requires more consideration as it requires a change in the underlying ontology. Future work will involve further integration with the Open Learner Model (OLM) using the OLM API, language localization, and testing the usage scenarios listed in in classroom settings.

5.3.2 Using CoNeTo

When the user first opens the website, the user is redirected to authenticate him/herself using CAS Authentication. On successful authentication, user is redirected to the application’s main page.

Once the appropriate Teacher Name, Course, and Topic are selected from the Diagram panel, the student can click on Load OLM button to start the negotiation process of their current OLM state. The student can also load an OLM XML file using the file browser.

The student can add a new Activity by clicking on the button Activity which leads to a rectangular shape appearing on the canvas representing the newly added activity with the default name Activity; and to add new Competency, clicking on the button Competency leads to an oval shape appearing on the canvas representing the newly added competency with default name Competency (Figure 15).
In order to add a relationship between two nodes, selecting/clicking one of the nodes leads to the selected node becoming highlighted. The type of relationship is identified by clicking on the appropriate button (green, red, and cyan coloured lines in Figure 15).

On selection of the second node, the relationship will be displayed between the two nodes as a straight line with a circle in the center displaying the corresponding symbol and background color depending on the selected relationship (Figure 16).

In order to edit a node, it is selected by clicking on it. The properties panel displayed on the right side of the canvas are populated by the current properties of the selected node (Figure 15). Each editable property can be edited.
To add a chat message related to a particular node, the intended node must be selected and highlighted. Using the text box for chat writes in a message, and clicking on Send records the message for this node (Figure 16) in the chat box, prefixed by the username of the currently logged in user.

Clicking on the Save button (Figure 15) leads to the diagram being saved in both the native CoNeTo format as well as OLM XML format. On successful save operation, a confirmation box is shown. A sample of the XML document representing the saved file on the server is represented below.

```xml
<?xml version="1.0" encoding="utf-8"?>
  <index>3</index>
  <studentEmail>demo</studentEmail>
  <teacherEmail />
  <topic />
  <nodes>
    <Node>
      <nodeId>0</nodeId>
      <nodeType>1</nodeType>
      <name>Activity 1</name>
      <details>This is Activity 1</details>
      <url />
      <position>
        <x>100</x>
        <y>50</y>
      </position>
      <messages />
    </Node>
    <Node>
      <nodeId>1</nodeId>
      <nodeType>2</nodeType>
      <name>Competency 1</name>
      <details>This is Competency 1</details>
      <url />
      <position>
        <x>328</x>
        <y>52</y>
      </position>
      <messages />
    </Node>
  </nodes>
  <relationships>
    <Relationship>
      <relationshipType>2</relationshipType>
      <firstNodeId>1</firstNodeId>
      <secondNodeId>0</secondNodeId>
    </Relationship>
  </relationships>
</Diagram>
```
In line with existing negotiated learner models, a student can challenge the model data. This may occur, for instance, if they believe certain activities they completed, were not taken into account in the model. This may take place using CoNeTo, the outcome of which is sent to the teacher; or it could occur face to face (with or without the support of CoNeTo), with the teacher inputting the result of the negotiation or discussion, if any changes to the model are needed (as in Figures 2 and 17).

If a teacher receives an argument from a student in the OLM, they can connect to a URL given as evidence. In Figure 17, this evidence is in the form of a spreadsheet, with a student claiming their ability to use mathematical information in communication (in English as a second language) through their construction of a spreadsheet for calculating expenditure for a holiday. Figure 17 also shows how the teacher can add new data (clicking on stars relating to competencies) and provide additional feedback to the student, in fields relating to strengths and suggestions for how to proceed (discussed in Section 1.2). Our example here is from the Norwegian national competence goals and curriculum plan for English (see Bull et al. (2012) and D4.3 for further discussion), using the “communication” competence of “use of technical and mathematical information in communication”. After receiving a student’s challenge from the negotiation tool, a teacher may choose to acknowledge the student’s evidence (spreadsheet) in the feedback fields, but also, for example: explain that the weighting of the evidence is low compared to that of other, more extensive activities (e.g. a teacher-marked essay, interaction in a virtual world); explain that this data has since been superseded; or explain that, when aggregated with other data sources for this competence, this entry has relatively little influence - if, for example, a student was challenging a skill at a broader level such as communication rather than use of technical and mathematical information in communication. Alternatively, discussion may be face-to-face. The negotiation tool may be used as a way for the teacher to help a student understand the argument/evidence relationships (they may explain nodes, change nodes or add new evidence nodes), or negotiation may occur around the learner model visualisations (Figures 8, 9 and 10) and evidence (Figures 3 and 17), without requiring use of the negotiation tool. Discussion would be similar to that described above, but will allow the teacher to more easily obtain further details from the student, about their perception of their competences. However discussion occurs, some degree of learning might take place during this process (Brna et al., 1999). This will also
need to be reflected in the learner model. The current NEXT-TELL solution is for the teacher to further update the model (again, as in Figure 2), but automatic updates may, in the future, also be applicable as an alternative update mechanism in NEXT-TELL.

In line with concerns that teachers may be reticent to accept student claims without evidence (Zapata-Rivera et al., 2007), control of the model data does not lie with the student. The student can enter their own self-assessments, which will be retained, but these will not override existing data unless this is agreed by the teacher (e.g. through one of the negotiation processes described above). This teacher control is similar to the power of the system in persuadable learner models (e.g. Mabbott & Bull, 2006; Thomson & Mitrovic, 2010), but the teacher may also initiate negotiation if they consider this to be beneficial for a student (e.g. to encourage reflection). Thus, there is also some similarity to the symmetrical approach of fully negotiated learner models (e.g. Bull & Pain, 1995; Dimitrova, 2003; Kerly & Bull, 2008).
6 Repertory Grids for Formative Assessment

This section describes the update to the design and development of RGFA, Repertory Grids for Formative Assessment designed and developed by CBS as a general-purpose spatial diagnostic tool for students’ concepts and their discriminations. The second release of RGFA takes into account results from two participatory design workshops with teachers and classroom usage by participating teachers (reported in D2.8). In addition to the originally envisioned uses, we are exploring possibilities for creating OLM representations of RGFA information (for example, the difference between the repertory grid of an “expert” vs. the students).

6.1 RGFA Technical Description

RGFA 2 is divided into three different layers: the Graphical User Interface (GUI), the Data Access Layer, and the Database. The GUI is an ASP.Net Web application. In the Data Access Layer we use ADO.NET and the database is a Microsoft SQL Server 2008 R2. We use Team Foundation Server (TFS) for source control. Below is a simplified diagram that shows an overview of RGFA’s implementation (Figure 18):

![Figure 18: Overview of RGFA implementation](image)

There are two different projects in the RGFA solution, RepertoryGridWeb and RepertoryGridModel. The RepertoryGridModel is imported in the RepertoryGridWeb project as an assembly. The actual communication with the database is done using ADO.NET Dataset objects. Figure 19 shows which dataset objects are used in RGFA.
One of the Dataset objects that is used by RGFA is the GRIDDEFELEMENT Dataset object. This object makes it possible to store the images, videos and texts from the triads to the database. The GRIDDEFELEMENT table has the following columns: GRIDDEFELEMENTID, GRIDID, ELEMENTENTITYNAME, SORTORDER, TYPE AND VIDEOURL.

The following code sample (with inline documentation comments) that shows how data is read from the database:

```csharp
/* This function populates the object representing one element of the exercise from the database. Every element in the database is saved using unique reference called GRIDDEFELEMENTID */
public void ReadFromDB() {
    //Initialise ADO.NET table adaptor
    _GRIDDEFELEMENT_TA = new GRIDDEFELEMENTTableAdapter();
    if(!GridDefElementID.HasValue)
        throw new Exception("id not found");
    //Fill the Datatable container fr GridDefElement within dataset
    _GRIDDEFELEMENT_TA.FillByGRIDDEFELEMENTID(_data.GRIDDEFELEMENT, GridDefElementID.Value);
    //Strong typed casting
    DataSetRGFA.GRIDDEFELEMENTRow row = (_data.GRIDDEFELEMENT.Rows[0]);
    GridID = row.GRIDID;
    ElementEntityName = row.ELEMENTENTITYNAME;
    SortOrder = row.SORTORDER;
    Type = row.Type;
    URL = row.VideoURL;
}
```
6.2 RGFA New Features

RGFA 2 supports triads with text, image, and video elements. The image and video elements are implemented as separate web controls that are added to the ctrlTriadAnswer Control. The image element is implemented as an ASP.NET Image control and the video element is implemented as an iframe. Images can be added, as shown in Figures 20 and 21; and video elements, as in Figure 22.

![Figure 20: Add Image Element Feature in RGFA 2: Number of Elements Specification](image1)

![Figure 21: Adding Image Element to RGFA 2: File Selection](image2)
When the triad is created all the data entered is saved to the database. The code sample (with inline documentation comments) shows how data is stored in the database:

```csharp
//This method writes the different types of grid elements to the database.
//We have three different types of elements, they are Image, Text and Video.
//Each element is stored in the GridDefElement Table in the RGFA database.

public void WriteNewToDB() {
    //Create a new instance of the GRIDDEFELELEM Table adapter
    _GRIDDEFELEMENT_TA = new GRIDDEFELEMENTTableAdapter();
    if (GridDefElementID.HasValue)
        throw new Exception("Already written");
    //A new instance of the GRIDDEFELEMENTRow is created
    DataSetRGFA.GRIDDEFELEMENTRow row =
        _data.GRIDDEFELEMENT.NewGRIDDEFELEMENTRow();
    Utility utility = new Utility();
    row.GRIDDEFELEMENTID = utility.GetNextValue("GRIDDEFELEMENT");
    if (Type == "IMAGE")
        //Check if Element type is Image
        row.Type = Type; //Set the row type to Image.
        row.ELEMENTENTITYNAME = ElementPath; //Set the Name of the Image
        row.VideoURL = URL;
    else if (Type == "VIDEO")
        //Check if Element type is Video
        row.Type = Type; //Set the row type to Video.
        row.VideoURL = URL; //Set the URL of the Video
        row.ELEMENTENTITYNAME = ElementEntityName; //Set the Name of the Video
    else //Else type is Text
    {
        row.Type = Type; //Set the row type to Text.
        row.ELEMENTENTITYNAME = ElementEntityName; //Set the Name of the Text
        row.VideoURL = URL;
    }
    //Set the GRIDID property
    row.GRIDID = GridID;
    //Set Sort Order
    row.SORTORDER = SortOrder;
    //Add the Row to the GRIDDEFELEMENT Dataset object
    _data.GRIDDEFELEMENT.Rows.Add(row);
    //Updated the GRIDDEFELEMENT Dataset
    _GRIDDEFELEMENT_TA.Update(_data.GRIDDEFELEMENT);
    //Set the GridDefElementID property
    GridDefElementID = row.GRIDDEFELEMENTID;
}
```

RGFA 2 has functionality requested by teachers to measure the time for answering triads and the time taken for the ratings phase. Each time segment is measured in seconds and is then stored in the database. All the
RGFA 2 incorporates visual reports for time consumed while answering a grid. This is made available under the "My Analytics" tab of the RGFA interface.

During the process of responding a grid there are two kinds of time on tasks: the first is the time used to answer a triad and second is the time taken while rating other elements in relation to a given triad. In RGFA 2, we are currently focusing on the time used to answer a triad. And on the landing page after successful login (MYGrid.aspx) a collective graph is shown for chosen Repertory Grid exercise. The collective graph depicts the average time used per triad for a given exercise. Average time for a given triad is calculated using total response available for given triad of the given exercise.

RGFA also provides the option of choosing whether the visual representation should be rendered as column, line or pie chart (Figures 23-29). Landing page after successful login loads the very first exercise in the list and also the visual graph is calculated and rendered. To view visual results for other exercises simply the exercise has to be selected from drop down list and graph will be updated accordingly.

Google charts have been used to render the graph on the web page. As mentioned before, RGFA is primarily developed using Microsoft technologies ASP.NET and C#. Structure, definitions and as well results of the
responded exercises are saved in the Microsoft SQL Server database. Google chart API requires data to be fed in a particular format known as JSON. To accomplish this, C# objects are transformed at runtime into data tables that are then further converted to Google chart API acceptable format (JSON string). JQuery is used to actually trigger the needed functions at the correct time that from the client machine sends request to Google Charts API server and eventually draws the resulting graph embedded on the web page.

Word cloud visualization has also been implemented in RGFA. Word clouds are also available for individual exercise as well as collective for all responses from given grid. The words used in the cloud are the words that users provide while defining similar and opposite constructs. A Google hosted open source project, named Visapi has been used for generating word clouds. Visapi word cloud uses Google visualization and so the input data format for the word cloud generation engine and method for generation is same as Google chart solution explained above.

Examples of teaching analytics screenshots are given below (Figures 23-29). As with the OLM (this deliverable), and the activity visualisations (D3.4), these screens can be helpful for teachers’ planning or for immediate use in the classroom. Furthermore, based on this information, teachers can add data to the learner model for consideration alongside that from activities, should they wish (as in Figure 2). Informed by the laboratory study on multiple representations of repertory grids data reported in D4.2, we designed three different representations for the time taken on the triads (elicitation of personal constructs + rating of elements) for the whole classroom (Figures 23-25) and for individual students (Figures 26-27). Word cloud visualizations are also shown at the collective classroom level (Figure 28) and the individual student level (Figure 29).

![Figure 23: RGFA 2 Teaching Analytics: Bar Chart of Triad Response Time](image)

![Figure 24: RGFA 2 Teaching Analytics: Line Chart of Triad Response Time](image)
Figure 25: RGFA 2 Teaching Analytics: Pie Chart of Triad Response Time

Figure 26: RGFA 2 Teaching Analytics: Pie Chart of Individual Student Response Time

Figure 27: RGFA 2 Teaching Analytics: Line Chart of Individual Student Response Time
Other new features in the release of RGFA version 2 are:

- Single login implementation
- When teacher/student login they can both created and answered repertory grids
- Visual analytics for the individual student
- Backward compatibility with all previous exercises

Current technical work with RGFA is implementing the feedback from teachers to provide teaching analytics support for personal constructs and the element ratings. We have also evaluating two standalone research prototypes that were developed based on the feedback from the Danish teachers. These two research prototypes will be integrated into RGFA for the next release. Design based research studies in the classrooms during Fall 2012 will inform the future research and technical developments of the use of repertory grids for formative assessment and RGFA respectively.
7 Shared Infrastructure for OLM and Activity Capture

All data that is modelled and collated about learners’ understanding and actions requires a level of context to be recorded, to make the information useful and for automated data manipulation to occur. Particulars such as the following are associated with each item of data:

For both Learner Modelling and Activity Visualisation

ID int – id of the data element
TIME timestamp – the time the data event occurred
STUDID int – the id of the student the information relates to
FORENAME varchar(30) – the forename of the student the information relates to
SURNAME varchar(30) – the surname of the student the information relates to
EVIDENCESOURCEID int – the id of the place the information originated from
EVIDENCESOURCENAME varchar(100) - the name of the place the information originated from
CONTRIBUTORID int – the user id of the person who contributed the information
CONTRIBUTORFORENAME varchar(30) – the forename of the person who contributed the information
CONTRIBUTORSURNAME varchar(30) – the surname of the person who contributed the information
CONTRIBUTORTYPE varchar(30) – the stakeholder type of the contributor {student, teacher, parent, peer}
TEACHERID int – the id of the teacher who is associated with this information
TEACHERFORENAME varchar(30) – the forename of the teacher associated with the information
TEACHERSURNAME varchar(30) – the surname of the teacher associated with the information
CLASSGROUPID int – the id of the group the student is associated with for the collection of this data element
CLASSGROUPNAME varchar(100) – the name of the group the student is associated with for this data element
ACTIVITYID int – the id of the activity used to collect this item of data
ACTIVITYNAME varchar(100) – the name of the activity used to collect this item of data
SUBJECTID int – the id of the subject the data element is associated with
SUBJECTNAME varchar(100) – the name of the subject the data element is associated with
UNITID int – the id of the unit of work with which the data element is associated
UNITNAME varchar(100) – the name of the unit of work with which the data element is associated
ARTEFACT varchar(3000) – the URL of the learning based artefact associated with this data item (optional)
TAGS varchar(1000) – any tags that accompany this data item (optional)

For Learner Modelling Only

COMPETENCYID int – the id of the competency the data element is associated with
COMPETENCYPATH varchar(1000) – the full name of the competency (describing the competency’s ‘parents’)
COMPETENCYNAME varchar(100) – the name of the competency
UNITINFLUENCE float(52) – the level of influence the unit has when combined with other units
ACTIVITYINFLUENCE float(52) – the level of influence the activity has when combined with other activities
COMPETENCYINFLUENCE float(52) – the level of influence the competency has when combined with others
DEPRECIATION float(52) – the influence of this information when combined with data points of different ages

The OLM and aspects of the learning process visualisations (detailed in D3.4) share this common technology infrastructure, with regard to context. This is done so that the tools may be used with or without other NEXT-TELL components, such as the Activity Stepper or ECAAD planner (see D2.4), whilst retaining the
information it needs to make its output useful. Therefore a set of tree-structure based configuration facilities are included to maintain 3 principal aspects of the underlying database (GUIs are shown in Figure 30):

- competency configuration (what we want to model)
- student and groups (who we want to model, and how they are related)
- curriculum content (how we want to collect the information, in terms of subjects, units of work, and data sources)

Creating an ‘activity’ (data collection exercise) in the curriculum content database draws together elements of each of these three database strands to define what we want to model, who we expect data from and what we expect the data to update (Figure 31). These are context setting factors. All changes are automatically saved.

Further to the development work of the first version of this platform (see D4.2), the competency database has been established (see Figure 6), the graphical interface reengineered, the curriculum content database established, the platform has been integrated with the CAS login system to synchronise users between tools, and an XML based API may be used to allow the platform to be updated and managed remotely from other pieces of NEXT-TELL software. For each set of configurations Create, Read, Update, and Delete facilities are
provided. Furthermore, the information database has been denormalized to improve the duration of search times.

Components of the OLM and activity visualisation are made available to the CoNeTo tool. Items are retrieved from the common infrastructure using the HTTP protocol, authenticated by the CAS authentication system. These components are then used by the CoNeTo tool to instantiate a negotiation episode.
8 Work in Progress

8.1 OLM

Development work on the shared infrastructure for the OLM and activity visualisation (Section 7) is ongoing. The following advancements are partially complete:

1) **Interaction logging protocol**, to capture human interaction with the software for research and evaluation purposes. The proposed fields of information captured are shown below.

| Information      | Data Content                                      | Rationale                                                      |
|------------------|---------------------------------------------------|                                                               |
| Event ID         | Unique number detailing the identity of the interaction event | Key field                                                    |
| Time/Date        | When the event occurred                           | Stage at which the event occurred                             |
| Instigator ID    | The ID of the user who triggered the event        | Who instigated the event                                       |
| User type        | The stakeholder type {student, peer, parent, teacher} | Role in the learning process the instigator holds             |
| Event Mode       | CONF configuration event NAVI inter-screen navigation LMOD Learner model event ADD evidence added READ evidence/model read UPDATE evidence/model updated | In general terms which type of event occurred and where it originated from. Information in this field may be used to filter and categorise origin quickly. |
| Event Identifier | Precise description of the event that took place   | Event identity (including event ID)                          |
| Event Data       | Data specific to the event                        | Supplementary data defining the instance of the event (e.g. outcome of modelling process, updated value) |

2) **Continued API development**, for configuration of the infrastructure, for supplementing evidence, and for remote reading of the model and database contents. This will allow for further integration between other project tools when complete. The range of information input and output functions will be increased.

3) **Embedded data entry facilities**, using Google spreadsheets. This will allow easy batch updates and configuration of the model using Google tools. For example in Figure 32 input values for the model, feedback and feedforward are entered into a student – competency matrix, which is to be automatically read during the upload process.
Teachers in particular will be able to collaboratively share configurations, designs, and entered student data (in accordance with Google’s privacy configurations) supporting the enquiry process. Templates will also be provided to support data collection and initial analysis to generate inferences, e.g. Figure 33, where student responses to quiz questions collated in a Google spreadsheet are used to determine a normalised score for each student and each competency, in preparation for input into the model. In this example each question can contribute to understanding of multiple competencies. Data collection may be automated, e.g. by using Google Forms.

Figure 32: Model input using Google Spreadsheets

Figure 33: Calculating inferences from multiple choice quiz questions
4) Integration with other OLM tools such as OLMlets (Bull et al., 2010a). OLMlets is an OLM tool that uses learners’ responses to multiple choice questions to create and model inferences about their understanding of different areas of study. OLMlets is domain independent and is a formative assessment tool that can be configured by the teacher. An instance of OLMlets has been configured for the project’s use (Figure 34). Current work is integrating this tool as a data source into the main OLM.

![Figure 34: Skill meter view from OLMlets](image)

5) Students able to share their learner model with their peers. A mechanism for students to optionally release their models to their peers (if permitted to do so by the teacher), is being developed. This is to support aspects of collaboration, such as useful in the meeting scenario; or to facilitate spontaneous collaboration and discussion as previously demonstrated for sharing OLMs at university level (Bull & Britland, 2007).

8.2 RGFA

In addition to OLM related developments, future work for RGFA includes:

- Continue design based research with school and university college teachers
- Continued OLM Integration
- Language Localization for German and Norwegian Schools
- Scripting and Scaffolding Support
- Integrate the Research Prototypes from VisTool into RGFA Web Solution
- Develop Principle Component Analysis and Multi-Dimensional Scaling Support
9 Summary

This report accompanies the D4.4 tools. The updates for this second version of the tools have been described, taking into account the requirements arising from the first annual review, the extensions and revisions suggested in D4.3, and recent research developments in the field. The deliverable has also introduced a new competency set (student meetings) for use by all project partners, as an illustration of the kind of 21st Century skills that NEXT-TELL can support.
10 References


11 Glossary

Terms used within the NEXT-TELL project, sorted alphabetically.

**BSCW** The document store used in NEXT-TELL used for storing internal documents

**Document store** see BSCW

**EuresTools** The reporting tool used in NEXT-TELL

**PM** Person month

**T** Task

**WP** Work package

**Partner Acronyms**

**JRS** JOANNEM RESEARCH Forschungsgesellschaft mbH, AT

**UniRes** UNI RESEARCH AS, NO

**KMRC** Medien in der Bildung Stiftung, DE

**TUG** Technische Universität Graz, AT

**CBS** Copenhagen Business School, DM

**BHAM** University of Birmingham, UK

**IOE** Institute of Education, University of London, UK

**EXACT** eXact Learning Solutions SPA, IT

**TALK** Verein offenes Lernen, AT

**BOC-AT** BOC Asset Management GmbH, AT

**BOC-PL** BOC Information Technologies Consulting SP.Z.O.O., PL

**MTO** MTO Psychologische Forschung und Beratung GmbH, DE

**Abbreviations**

**BS** Baseline Study

**CbKST** Competence-based Knowledge Space Theory Training Course

**CBT** Computer Based Training

**DBR** Design-Based Research

**ECAAD** Evidence Centered Activity and Appraisal Design (builds on the ECD)

**ECD** Evidence Centered assessment Design (e.g. PADI project)

**EFL** ‘English as a Foreign Language’; EFL refers to learning English in a non-English-speaking region, such as studying English in an Asian or Latin American nation. Typically, EFL is learned as part of a student’s school curriculum or for career purposes if working for an international corporation.

**ENA** Epistemic Network Analysis

**ESL** English as a Second Language; refers to learning English in the target language environment

**HCl** Human Computer Interaction

**ICT** Information and Communication Technology

**IT** Information Technology

**LEPP** Longitudinal Evaluation of Performance in Psychology (2nd generation e-portfolio)

**NEXT-TELL** Next Generation Teaching, Education and Learning for Life

**OLM** Open Learner Model
The PADI project aims to provide a practical, theory-based approach to developing quality assessments of science inquiry by combining developments in cognitive psychology and research on science inquiry with advances in measurement theory and technology.

Requirement Analysis

Researcher-led Design Study

Stanford Research Institute

The Science, Technology, Engineering, and Mathematics (STEM) fields are collectively considered core technological underpinnings of an advanced society, according to both the National Research Council and the National Science Foundation.

Teacher-led Design Study

Technology Enhanced Learning

Teaching English as Second Language

Teachers Inquiry into Students Learning

NEXT-TELL partners responsible for generating tools and methods

- BOC-AT: ECAAD
- BOC-PL: SPICE
- EXACT: Moodle
- JRS/EXACT: Google Docs and Google Spreadsheet
- TALK: OpenSim
- CBS: Rep5
- JRS: EVE
- EXACT: Mahara ePortfolio
- BHAM: OLM

Acknowledgement: The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 258114.