

**Title of project (maximum two lines)**

Title: Integration of **Auralog Tell me more (TMM) Student Account System** with SITS (parallel to LSS VLE)

**Member of academic or academic-related staff responsible for leading the project.**

Name: Dr Thomas Plagwitz.....

**Any other staff who will be supported.**

**How will the project help to achieve the objectives of the FLD Centre?**

❖ Status

- TMM is a sophisticated learning environment which includes a much needed infrastructure for
  - Assigning learning tasks to groups of students and individuals;
  - Automatic evaluation and grading of student language output (including advanced skills writing and speaking)
  - Collecting student result data (including exportation for data-mining);
  - Providing computer-mediated instructor feedback (messaging).
- TMM seems to have been designed not for a secondary or post-secondary education environment, but for a business environment where a largely independent learner is only **loosely coupled with a tutor**, while LSS want to use it as a component in a highly-integrated service to enhance the value of our educational offering which revolves around **instructors and the modules they teach**. However, TMM is **agnostic of classes**. In particular, in TMM a student can be paired only with one “tutor” per target language at a time. However, for e.g. 2006 TP1 SITS data, more than 59% of student enrolment in LSS was in a 2<sup>nd</sup>, 3<sup>rd</sup> or higher language module.
- TMM’s **batch account creation** facility is severely **lacking** in functionality.
  - Student accounts can only be batch created once. Attempts to create duplicates will not only fail for the individual account, but let the entire batch creation fail. This affects
    - both new students if they are enrolled in more than module per language in a given term,
    - as well as all those students for which accounts have been created in previous terms. All updates to module enrolment can not be done in a batch, but only through a cumbersome GUI operation for each individual student.

❖ Objective

- In particular, we need to adapt TMM to the fact that our students enrol in module instantiations (= classes = student-instructor-pairings) to interact with instructors.
- In general, for second language acquisition (SLA), where “the medium is the message”, “contact hours” between student and instructor (using the second language) may be even more crucial than in other subjects. To convince instructors that such “contact hours” can be extended through an AI instead of instructor time, therefore, requires possibly more than in other fields a firm framework of interaction between instructor and student to fall back on.

- The project will **match instructors with the students in their class**, similar to what instructors and students have grown accustomed to from the VLE (Blackboard for LSS).
- This matching – that students and instructors can effectively communicate with each other – is **our business model**, as educational service for students by instructors that increasingly integrates technology.
- Without communication between instructor and student, and with communication just between student and technology:
  - The creation of customized learning paths (see **separate FLDC proposals by LSS involving TMM**) on a class level and even more so on an individual level, will be severely limited in its usefulness since
    - Exercises cannot be effectively (automatically or based on syllabus) assigned, and the manual navigation to exercises is notoriously difficult in TMM.
    - Student exercise results cannot be easily assessed since they cannot be aggregated like in the Blackboard grade book, and customization of learning provision will stop. From my past experience, resulting take-up will be minimal (“If it is not assessed, it is not assigned” attitude).
  - Instead of providing a complex integrated service, we would be reduced to reselling (tuition → software license, plus small installation service) of software which, moreover, our students can obtain online for free (even though not legally).

**Which precise groups of students, programmes and modules (with anticipated numbers) will directly benefit from the project?**

- ❖ Programme / modules
  - All language (English, French, German Spanish) learning (and reviewing) modules that will use TMM will benefit from this integration.
  - Specifically, the project leader has been asked to create accounts for all students enrolled in modules whose code matches the following regular expression pattern: “L[A-EFGIS][1-9]”,
    - Equals 339 modules overall, of which taught **163 modules** for 2006 P1 and P2 (SITS data from 19 Oct 2006 and 21 Dec 2006),
- ❖ Which, for e.g. Fall 2006, was
  - **Students** (= distinct **enrolments**): **1692** for P1 as of 19 Oct 2006, 1400 for P2 as of 21 Dec 2006, 2067 for 2007/2008 P1 [sans preessionals].
  - **Instructors** = Department code LES = **96** instructors (FALL 2006) [needs to be cross-referenced to “L[A-EFGIS][1-9]”]
- ❖ External benefit: Possible sale of the script as an extension to TMM.
  - TMM in has a large installed base which can multiply the benefit from this project’s outcome.<sup>1</sup> The Project leader has experienced Auralog’s marketing efforts at his two most recent institutions and expects to see many more.<sup>2</sup>

---

<sup>1</sup> Auralog® is deemed one of the leaders in the European foreign language software providers ([http://ec.europa.eu/education/programmes/elearning/doc/studies/market\\_annex1\\_en.pdf](http://ec.europa.eu/education/programmes/elearning/doc/studies/market_annex1_en.pdf), p. 14, 24). The company claims having deployments in “[m]ore than 300 corporate groups and organizations throughout the world, including: GlaxoSmithKline Biologicals, Corbis, Telefonica, Exxon Mobil, Toyota, FBI, The US Government, The US Army National Guard” and that “there are a large number of [HE] institutions in the United States equipped with TELL ME MORE Education, including: Rice University, Virginia Commonwealth University, Bryant University, Clemson University, North Carolina State University, The Citadel, Mercer University, St. Ignatius College Prep (Chicago), Miami-Dade County)” ([http://www.auralog.com/us/education\\_home\\_online.htm](http://www.auralog.com/us/education_home_online.htm)). Especially their speech recognition feature has received favorable commendation in reviews by foreign language

**On what timescale will the development take place and be implemented?** (please include a schedule of the major activities.)

- ❖ The programme leader has already
  - Negotiated an agreement with ISA to get from SITS a nightly updated CSV file of current student enrolment in courses “L[AEFGIS][1-9]” on a FTP server for download
  - Explored the TMM setup, including
    - Finding and testing the **StudentGroupType** (not, as one would expect, the studentgroup) workaround for module instantiations
    - Writing (and running at night time) scripts to create **all possibly materializing module instantiations**, including group subdivisions, from 2006 to 2016
      - scripts (addStudentGroupTypesAndNames) worked, TMM failed (controls stopped displaying all items, contrary to vendor assurance that there are no limits):
        - ◆ at slightly over 30000 studentgroup types in 1<sup>st</sup>,
        - ◆ at slightly over 60000 studentgroup types in 2<sup>nd</sup> attempt
    - Devising the workaround of dynamically reflecting **only actually materializing module instantiations** as studentgroup types

**List the outcomes and outputs from the project, and how they will be evaluated.**

- ❖ Deliverables: Script for creation and term-wise updating of TMM Accounts from CSV data of SITS for Students, comprising
  - SITS mirroring: automate reflection of SITS changes into TMM
    - And functions to manipulate STUDENTGROUP SETS ()
    - Outlined in Figure 1: SITS Integration
  - CSV – Function: intermediate Layer between SITS and TMM for data storage
    - Will load import the data from SITS
    - Will persist the data from TMM that can only be access through the GUI
    - " Outlined in Figure 3: Automating TMM
  - Subsidiary functions
    - Automation of STANDARD OPERATIONS IN TMM and helper functions for the same
      - A reusable library
      - Outlined in Figure 3: Automating TMM
    - BUSINESS RULES: mirror university regulations

---

technologists (<http://www.hull.ac.uk/cti/reviews/tellme.htm>,  
<http://www.comp.leeds.ac.uk/eric/ci2003/ObaAtwell.doc>,  
<http://www.alladin.ac.uk/SUPPORT/downloads/needsOppChall.doc>,  
<http://lt.msu.edu/vol8num3/review1/default.html>).

<sup>2</sup> Cf. Auralog Demo at VLE (Virtual Learning Environment) Languages User Group – Third annual meeting on 29/03/2007.

```

132 ⊕ //////////////// functions for SITs integration
134 ⊕ func CoursesSits2AuralogStudentGroups($astrCourses) ; todo: transfer the sits-module-insta
147 EndFunc
148 ⊕ func StudentCoursesSits2AuralogStudentGroups($ptrStudentModulesHash) ; todo: for all sits-
172 endFunc
173 ⊕ func StudentCoursesSits2AuralogLanguageGroups() ; a student's (module-instantiation enroll
191 endFunc
192 ⊕ //////////////// FUNC for STUDENTGROUP SETS ////////////////
197 ⊕ func getAuralogsPastStudentGroups(ByRef $astrCurrentStudentGroups) ; todo
201 EndFunc
202 ⊕ func persistStudentPastStudentGroups($ptrStudentStudentGroupsHash) ; todo:
205 EndFunc
206 ⊕ func getAllAuralogStudentGroups(ByRef $astrAllStudentGroups) ; return all (past and curren
210 EndFunc
211 ⊕ func removePastStudentGroups(ByRef $astrAllStudentGroups, ByRef $astrCurrentStudentGroups)
238 EndFunc

```

Figure 1: SITs Integration

```

971 ⊕ func downloadCSV()
974 EndFunc
975 ⊕ func ftpCSV() ; todo: run ms ftp with connection data
988 EndFunc
989 ⊕ func renameCSV() ; may be superfluous, ftpcsv can rename
995 EndFunc
996 ⊕ func findPreviousCSV() ; search for the most recent existing older sits csv to compare with
1010 EndFunc
1011 ⊕ func compareOldNewSPRCodeNumber(ByRef $astrOnlyInOld, ByRef $astrOnlyInNew)
1067 EndFunc
1068 ⊕ func CSVfile2Array($CSVFilePath) ; read (and returns) AN (old or new) sits csv file into an
1083 EndFunc
1084 ⊕ func getCSVColumnHeaders($lRecords) ; return columnheaders-array for a a standard (sits-pro
1093 EndFunc
1094 ⊕ func identifyCSVColumns($astrCSVColumnHeaders) ; determine (in global variables), where (=
1127 endfunc
1128 ⊕ func CoursesCSV2Array(); returns array of (module-codes, not courses) from the (current) s
1134 EndFunc
1135 ⊕ func CoursesArray2PersistCSV($astrCoursesArrayNew, $blnMerge = 1, $base = 1) ; persist todo:
1165 endFunc
1166 ⊕ func AuralogStudentGroups() ; todo: for starters, get once, into a global variable, the
1179 EndFunc
1180 ⊕ func StudentCoursesCSV2Hash($ptrStudentCourses, $strCSVFile, $astrSprCodeFilter = 0) ; conv
1210 EndFunc

```

Figure 2: CSV Layer

lotmail.com

```

340      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; FUNCTIONS for STANDARD OPERATIONS IN AURALOG;;;
341  ⊕ func studentsRemoveGroups() ; for all auralog students, remove all studentgroups
376  EndFunc
377  ⊕ func studentRemoveReturnStudentGroups() ; remove current student from all student groups, but a
382  EndFunc
383  ⊕ func studentAddStudentGroups($astrStudentStudentGroups) ; todo: add current student to (paramet
397  EndFunc
398  ⊕ Func addStudentGroupTypesAndNames(); this needs to be run only once; later additions to modules
641  EndFunc
642  ⊕ ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; HELPER FUNCTIONS for STANDARD OPERATIONS IN AUF
644  ⊕ func SetupProgramPart($windowtitle,$strPart)
691  EndFunc
692  ⊕ Func SetupProgram($windowtitle)
723  EndFunc
724  ⊕ func copyEditControl($windowTitle, $editname)
728  EndFunc
729  ⊕ func clearEditControl($windowTitle, $editname); for auralog edit controls
733  EndFunc
734  ⊕ func openWindowModify($intindex = "")
767  EndFunc
768  ⊕ func closeOKWindowModifyStudentStudent()
786  EndFunc
787  ⊕ Func openwindowStudentGroupTypes()
805  EndFunc
806  ⊕ Func openwindowStudentGroupTypesAdd()
823  EndFunc
824  ⊕ Func openwindowStudentGroupAdd()
841  EndFunc
842  ⊕ Func sortSysListViewByName($windowTitle, $controlname); thru click on listview column header, a
896  EndFunc
897  ⊕ Func SearchStudent($strAppname, $strStudentname); todo: search = put into listview, the (paramet
907  EndFunc
908  ⊕ func _listviewAllItemsRemoveReturn($windowTitle, $listviewClassNN, $removebuttonclassNN); from
949  EndFunc
950  ⊕ func _listviewFirstItemRemoveReturn($windowTitle, $listviewClassNN, $removebuttonclassNN)
969  EndFunc

```

Figure 3: Automating TMM

N@hotmail.com