

# Helping Non-Experts Build Speech-Enabled Online CALL Courses

Manny Rayner

(joint work with Claudia Baur, Pierrette Bouillon,  
Nikos Tsourakis)



# Overview

- CALL-SLT → Open CALL-SLT
- Constructing CALL-SLT courses
- Deploying CALL-SLT courses
- And next...

# Overview

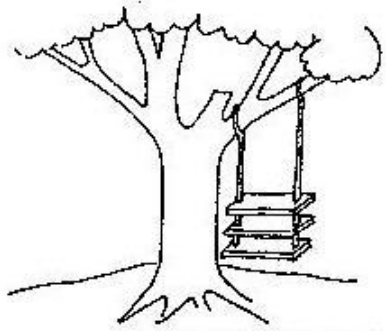
- CALL-SLT → Open CALL-SLT
- Constructing CALL-SLT courses
- Deploying CALL-SLT courses
- And next...



# A brief history of CALL-SLT

- “Spoken translation game” project
  - Funded by Swiss NSF, started mid-2009
- Second phase started 2014

# Original idea

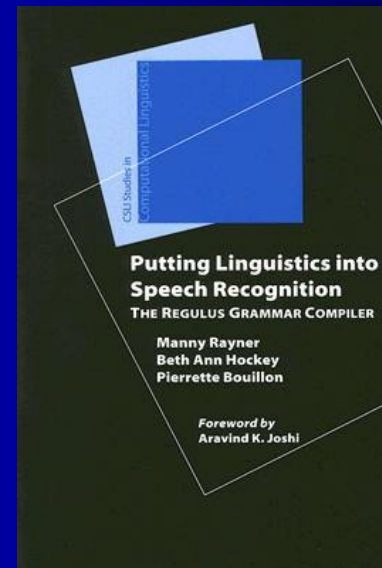


- We'd built this cool speech translation app
- People who played with it seemed to pick up spoken language skills
- Can we turn it into a spoken CALL app?



# Project proposal

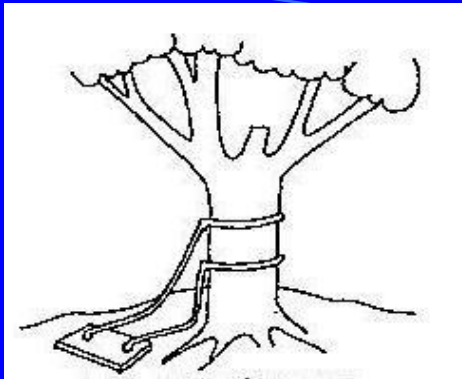
- Hm... MIT already did something similar
- We need to be different (research, right?)
- Linguistically motivated, grammar-based recognition!
- Regulus platform!



A black and white line drawing of a tree with a building or structure integrated into its trunk and branches.

# Project proposal

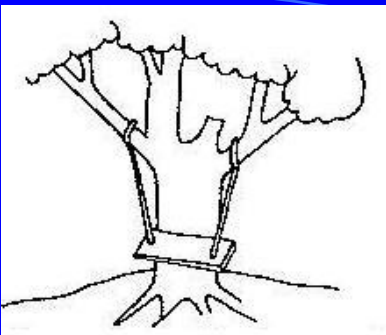
- Translate from L2 to human-readable interlingua based on L1
  - Prompts in interlingua
- Speech translation from L2 to L1
- Match result against prompt



# First implemented version

- Deployed on desktop machine
  - Typical prompt:  
« DEMANDER DE\_MANIERE\_POLIE  
TABLE 2 PERSONNE(S) »
  - Typical spoken responses:
    - “I would like a table for two”
    - “A table for two people please”

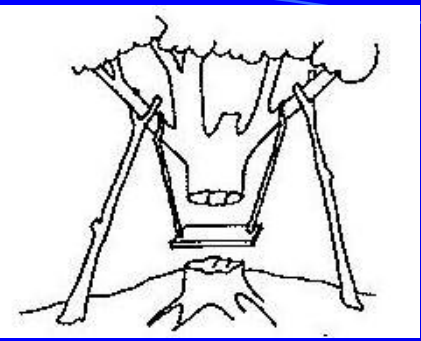




# Second implemented version



- Hm... no one can install system at home
  - Redeploy on internet, so actually usable
- Redesign interlingua
  - Make prompts more natural
- Content still un motivating
  - Students not interested

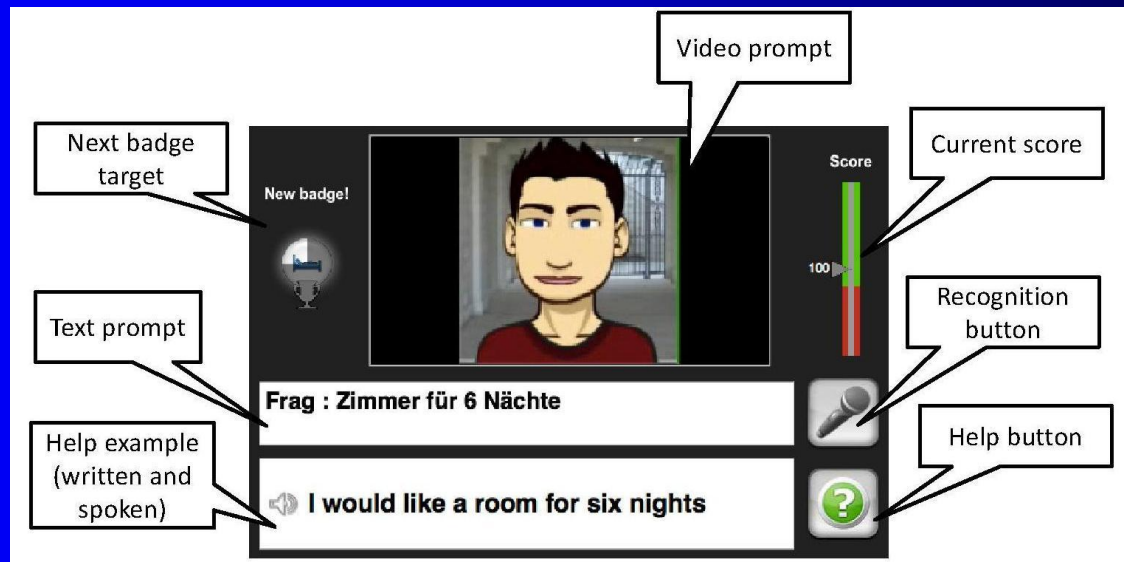


# Evaluation version

- Add scripts to allow interactive dialogues
- Ad hoc rewriting rules to make prompts more natural
- Original elegance of design gone
- Only experts can implement content
- Only experts can deploy the system

# Evaluation version

- ... but students do now want to use it
- Eight interactive English lessons for German 12-year-olds based on textbook

A screenshot of the CALL-SLT interface with several callout boxes pointing to specific elements. The interface includes a video prompt of a man's face, a score bar, a recognition button, a help button, and a text prompt area. A "Next badge target" is also visible on the left side.

Next badge target

Text prompt

Help example (written and spoken)

Video prompt

Current score

Recognition button

Help button

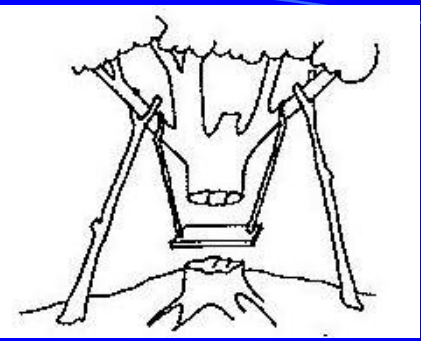
New badge!

Score

100

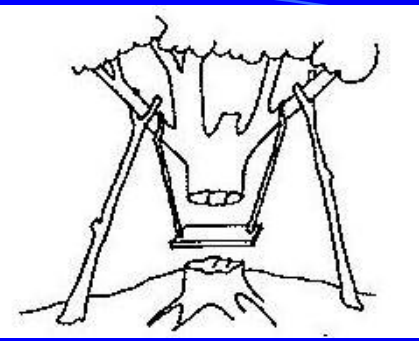
Frag : Zimmer für 6 Nächte

I would like a room for six nights



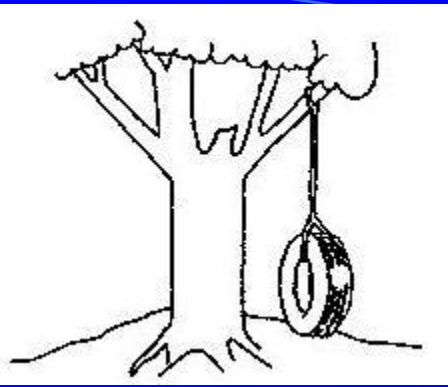
# Evaluation in Swiss schools

- Q4 2014, German-speaking Switzerland
  - 15 schools
  - ~225 students
  - ~25K logged utterances
- Teachers and students both liked it  
(Mostly work by Claudia Baur: several papers, 2013-2015)



## Bottom line so far

- We can build useful courses, but...
- ... the framework is ugly
- ... it takes too long
- ... only experts can do it



# Can we do better?

## BASIC DESIGN PRINCIPLES

- “Build One to Throw Away”
- “Keep It Simple, Stupid”
- “Web 2.0”

# Build one to throw away

- Do we need the sophisticated recognition architecture?

# Build one to throw away

- Do we need the sophisticated recognition architecture?
- No



# Recognition

- Recognition task is highly constrained
  - Prompt defines responses
  - Sophisticated recognition unnecessary
- Use enumerative approach
  - First cut: just list possible responses
  - Add simple grammar if necessary

# Keep It Simple, Stupid

- Minimal formalism
- Less is more
- Do not require specialist knowledge

# Web 2.0

- “Every downloader is a potential uploader”
- Be inclusive
- Paradigm: HTML

# Overview

- CALL-SLT → Open CALL-SLT
  - Constructing CALL-SLT courses
- Deploying CALL-SLT courses
- And next...

# Different kinds of designers

- Six different levels of complexity
- Lowest level: basic web-literate
  - Writing, saving and uploading files
- Highest level: non-trivial software skills
  - Ability to debug a simple program

# Six levels

1. Prompts and responses
2. Multimedia
3. Regular expressions and templates
4. Scripts
5. Gamification
6. Advanced scripts

# Level 1

## Prompts and responses

- Simplest kind of course
- List of prompt/response pairs
- Prompt is piece of text
- One or more responses

# Level 1

## Prompts and responses

Prompt

Lesson

pronunciation\_h

Group

4

Text/french

Dis que: tu déteste les légumes

Response

i hate vegetables

Response

\* i ate vegetables

EndPrompt



# Level 1

## Prompts and responses

The screenshot displays the CALL-SLT application interface. At the top, the title "CALL-SLT" is centered in white, with the subtitle "A PROTOTYPE CONVERSATION PARTNER" below it. The interface is divided into several sections:

- Left Navigation Panel:** Contains four vertical buttons with colored squares above them: "INTRODUCTION" (grey), "SELECT DOMAIN" (yellow), "SELECT LESSON" (purple), and "USE THE APPLICATION" (blue).
- Central Conversation Area:**
  - A white box contains the prompt: "dis\_que : tu détestes les légumes". To its left is a green left-pointing arrow, and to its right is a green right-pointing arrow.
  - Below this is a white input field containing the user's response: "I ate vegetables". This field is highlighted with a red border. To its right is a microphone icon.
  - Below the input field is another empty white box, and to its right is a green question mark icon.
- Right Navigation Panel:** Contains two vertical buttons with colored squares above them: "MY BADGES" (orange) and "COMPILE & REDEPLOY" (green).

# Level 2 Multimedia

- Same as Level 1
- ... except that Prompts can have multimedia

# Level 2 Multimedia

Prompt

Lesson

Multimedia

Text/english

Response

Response

Response

EndPrompt

who

pikachu.jpeg

Who is it?

pikachu

it is pikachu

it's pikachu

# Level 2 Multimedia



# Level 3

## RegExs, templates, grammar

- You often have
  - Similar responses in the same Prompt
  - Similar Prompt units
- Level 3 gives tools for handling this

## Level 3

# RegExs, templates, grammar

```
i (want | need) a single room ?please
```

instead of

```
i want a single room
```

```
i need a single room
```

```
i want a single room please
```

```
i need a single room please
```

# Level 4 Scripts

- Link up prompts to create a dialogue
- Simple XML-based scripting language
- Script = set of <step> units

# Level 4 Scripts

```
<step>  
  <id>ask_for_number_nights</id>  
  <group>room_for_number_of_nights</group>  
  <limit>is_one_night_okay</limit>  
  <success>ask_type_of_room</success>  
</step>
```



# Level 5 Gamification

- Gamification may increase motivation
- Simple score/badge framework
- Designer can specify badges, thresholds, penalties, bonuses etc

# Level 5 Gamification



# Level 6

## Advanced scripts

- Allow multiple paths through script
  - “Conditional steps”
- Make one step depend on another
  - “Semantic tags”

# Level 6

## Advanced scripts

```
<step>
  <id>enjoy_your_stay</id>
  <group>thanks</group>
  <limit>exit</limit>
  <success cond="level >= silver">
    is_everything_okay
  </success>
  <success>exit</success>
```

# Overview

- CALL-SLT → Open CALL-SLT
- Constructing CALL-SLT courses
  - Deploying CALL-SLT courses
- And next...

# Uploading courses

- In principle, very simple: copy course to server and run it
- In practice, less straightforward
  - Many users sharing same server
  - Stop users interfering with each other
  - Hide complexity of upload process

# Uploading courses: file structure

- Each user owns a namespace directory
- Namespace directory contains one or more course directories
- Course directory has up to 4 subdirectories
  - course description
  - multimedia
  - scripts
  - doc

# Uploading courses: what the user sees

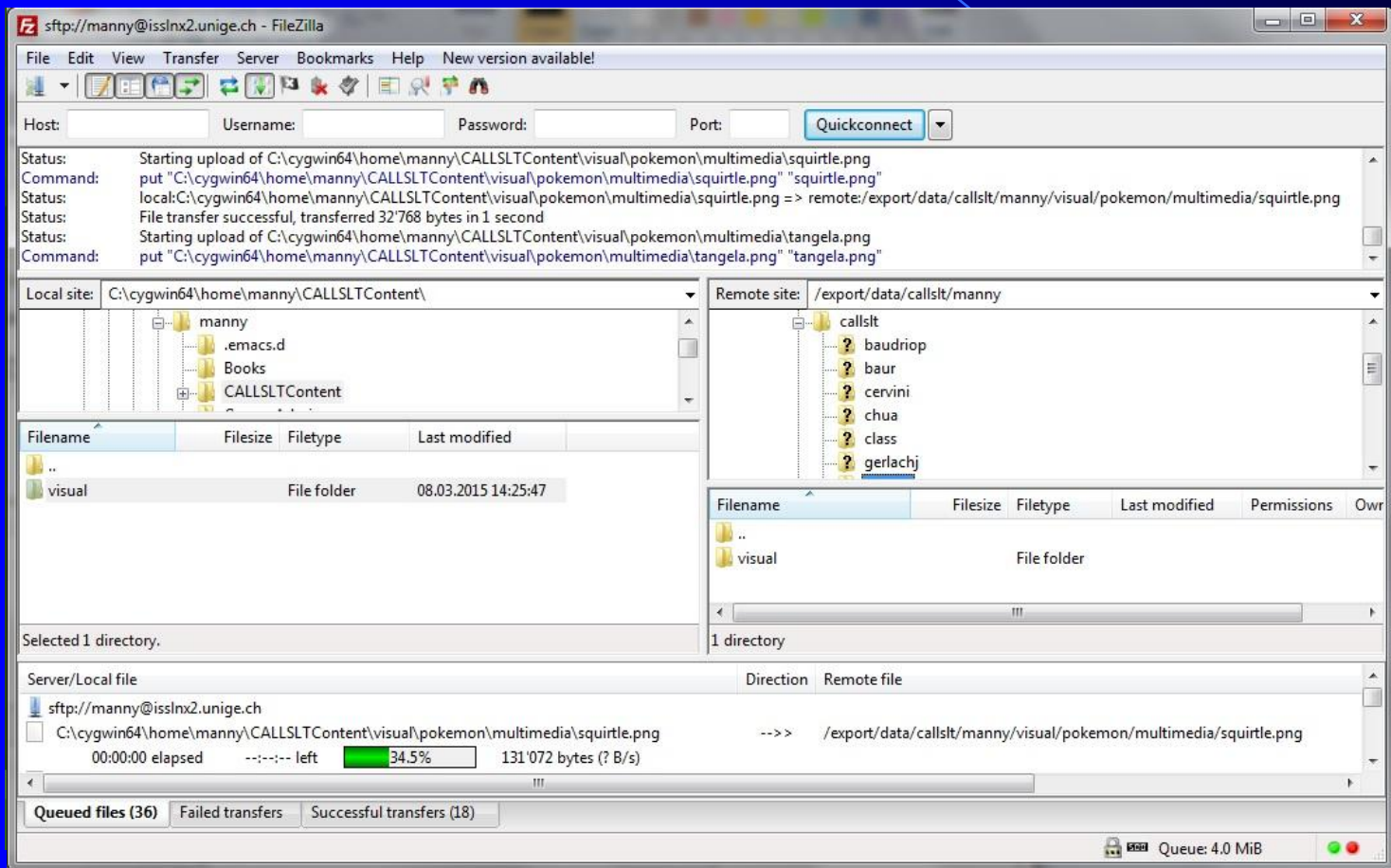
- Five step process
  - UPLOAD
  - SELECT
  - COMPILE
  - TEST
  - DEPLOY



# Uploading courses: UPLOAD

- What the user sees
  - Connect to server
  - Drag and drop directory

# Uploading courses: UPLOAD



# Uploading courses: SELECT

- What the user sees
  - Select course from menu
  - Confirmation

# Uploading courses: SELECT

- What the user sees
  - Select course from menu
  - Confirmation
- What happens on server
  - Check FTP directory
  - Register new courses
  - Copy to compilation directory

# Uploading courses: SELECT

The screenshot displays the CALL-SLT web application interface. At the top, the title "CALL-SLT" is prominently displayed in white on a green background, with the subtitle "A PROTOTYPE CONVERSATION PARTNER" below it. A navigation sidebar on the left contains six items: "INTRODUCTION", "SELECT DOMAIN", "SELECT LESSON", "USE THE APPLICATION", "MY BADGES", and "COMPILE & REDEPLOY", each with a corresponding colored square. The main content area features a breadcrumb trail: "Tutorial: Writing CALL-SLT Lite courses [more](#)". Below this, a progress bar shows four steps: "Step 1: Select a course" (highlighted in purple), "Step 2: Compile", "Step 3: Test", and "Step 4: Release". The "Step 1" area contains a large empty rectangular box, a "Refresh" button, and three input fields labeled "Namespace:", "Course:", and "L1:" (with "L2:" next to it). The "L1:" and "L2:" fields are split into two separate input boxes.

# Uploading courses: SELECT

The screenshot displays the CALL-SLT web application interface. At the top, the title "CALL-SLT" is prominently displayed in white on a green background, with the subtitle "A PROTOTYPE CONVERSATION PARTNER" below it. A navigation sidebar on the left contains six vertical buttons: "INTRODUCTION" (grey), "SELECT DOMAIN" (yellow), "SELECT LESSON" (purple), "USE THE APPLICATION" (cyan), "MY BADGES" (orange), and "COMPILE & REDEPLOY" (green). The main content area features a breadcrumb trail: "Tutorial: Writing CALL-SLT Lite courses [more](#)". Below this, a progress bar shows four steps: "Step 1: Select a course" (highlighted in purple), "Step 2: Compile", "Step 3: Test", and "Step 4: Release". The primary content area is a white box containing a list of course options: "english\_course", "hello", "pronunciation", "toy", and "visual", each with a small plus icon to its left. A "Refresh" button is positioned to the right of the list. Below the list, there are input fields for "Namespace:", "Course:", "L1:", and "L2:", each with a corresponding blacked-out text box.

# Uploading courses: SELECT

The screenshot displays the CALL-SLT web application interface. At the top, the title 'CALL-SLT' is prominently displayed in white on a green background, with the subtitle 'A PROTOTYPE CONVERSATION PARTNER' below it. A navigation sidebar on the left contains six items: 'INTRODUCTION', 'SELECT DOMAIN', 'SELECT LESSON', 'USE THE APPLICATION', 'MY BADGES', and 'COMPILE & REDEPLOY', each with a corresponding colored square. The main content area is titled 'Tutorial: Writing CALL-SLT Lite courses' and features a progress bar with four steps: 'Step 1: Select a course' (highlighted in purple), 'Step 2: Compile', 'Step 3: Test', and 'Step 4: Release'. Below the progress bar, a tree view shows a folder structure: 'toy' (expanded), 'visual' (expanded), 'visual1\_course', 'visual2\_course', 'visual3\_course', and 'english' (selected and highlighted in blue). A 'Refresh' button is located to the right of the tree view. Below the tree view, there are three input fields: 'Namespace: visual', 'Course: visualcourse', and 'L1: english' (with 'L2: french' also visible).

CALL-SLT  
A PROTOTYPE CONVERSATION PARTNER

Tutorial: Writing CALL-SLT Lite courses [more](#)

Step 1: Select a course Step 2: Compile Step 3: Test Step 4: Release

- toy
  - visual
    - visual1\_course
    - visual2\_course
    - visual3\_course
    - english

Refresh

Namespace: visual  
Course: visualcourse  
L1: english L2: french

# Uploading courses: COMPILE

- What the user sees
  - Press Compile button
  - Receive feedback



# Uploading courses: COMPILE

- What the user sees
  - Press Compile button
  - Receive feedback
- What happens on server
  - Compiles course description
    - Creates Nuance grammar, tables etc
  - Compiles Nuance grammar
  - If successful, adds touchfile

# Uploading courses: COMPILE


The screenshot displays the CALL-SLT web interface. At the top, the title "CALL-SLT" is prominently displayed in white on a green background, with the subtitle "A PROTOTYPE CONVERSATION PARTNER" below it. A navigation sidebar on the left contains six items: "INTRODUCTION", "SELECT DOMAIN", "SELECT LESSON", "USE THE APPLICATION", "MY BADGES", and "COMPILE & REDEPLOY". The "COMPILE & REDEPLOY" item is highlighted in green. The main content area features a progress bar with four steps: "Step 1: Select a course", "Step 2: Compile" (highlighted in purple), "Step 3: Test", and "Step 4: Release". Below the progress bar, a large white box contains a vertical scrollbar, indicating a scrollable area. At the bottom of this box, the text "Compile:" is followed by a green circular button.

CALL-SLT  
A PROTOTYPE CONVERSATION PARTNER

Tutorial: Writing CALL-SLT Lite courses [more](#)

Step 1: Select a course Step 2: Compile Step 3: Test Step 4: Release

INTRODUCTION  
SELECT DOMAIN  
SELECT LESSON  
USE THE APPLICATION  
MY BADGES  
COMPILE & REDEPLOY

Compile: 

# Uploading courses: COMPILE

The screenshot displays the CALL-SLT web interface. At the top, the title 'CALL-SLT' is centered in white on a dark green background, with the subtitle 'A PROTOTYPE CONVERSATION PARTNER' below it. A navigation sidebar on the left contains six items: 'INTRODUCTION' (grey), 'SELECT DOMAIN' (yellow), 'SELECT LESSON' (purple), 'USE THE APPLICATION' (cyan), 'MY BADGES' (orange), and 'COMPILE & REDEPLOY' (green, which is highlighted). The main content area shows a progress bar with four steps: 'Step 1: Select a course', 'Step 2: Compile' (highlighted in purple), 'Step 3: Test', and 'Step 4: Release'. Below the progress bar, a text box with a green border contains the following status report:

- 2 templates internalised,
- 4 template applications expanded,
- Removed 38 incorrect responses duplicating correct ones,
- Found 45 units and 227 responses for namespace "visual",
- Stored 0 scripts,
- Loaded 0 multimedia file definitions,
- Loaded 0 phrase scores,
- All groups in scripts are defined,
- All multimedia references in prompts are defined,
- Not yet checking multimedia files,
- No Lite compile errors in namespaces "visual".]

Below the text box, the label 'Compile:' is followed by a green circular progress indicator.

# Uploading courses: TEST

- What the user sees
  - Press Test button
  - Receive feedback

# Uploading courses: TEST

- What the user sees
  - Press Test button
  - Receive feedback
- What happens on server
  - Copies course dir to staging server
  - Recompile all course descriptions
  - Redeploys system with new Nuance grammar

# Uploading courses: TEST

The screenshot displays the CALL-SLT web interface. At the top, the title "CALL-SLT" is centered in white on a green background, with the subtitle "A PROTOTYPE CONVERSATION PARTNER" below it. A navigation sidebar on the left contains six items: "INTRODUCTION" (grey), "SELECT DOMAIN" (yellow), "SELECT LESSON" (purple), "USE THE APPLICATION" (cyan), "MY BADGES" (red), and "COMPILE & REDEPLOY" (green). The main content area features a progress bar with four steps: "Step 1: Select a course", "Step 2: Compile", "Step 3: Test" (highlighted in purple), and "Step 4: Release". Below the progress bar is a large white rectangular area, currently empty. At the bottom left of this area are four black squares, and at the bottom right is a "Test:" label next to a green circular button. A link "Tutorial: Writing CALL-SLT Lite courses [more](#)" is visible above the progress bar.

# Uploading courses: TEST

**CALL-SLT**  
A PROTOTYPE CONVERSATION PARTNER

Tutorial: [Writing CALL-SLT Lite courses](#) [more](#)

Step 1: Select a course   Step 2: Compile   **Step 3: Test**   Step 4: Release

```
---[ok, *** Warning: directory
z:/timissco/litecontentstaging/bologna/se_presenter not found for
namespace bologna and course se_presenter, *** Warning: directory
z:/timissco/litecontentstaging/bologna/rendezvous not found for namespace
bologna and course rendezvous, *** Warning: file
z:/timissco/litecontentstaging/bologna/se_presenter/grammars/se_presenter.
txt not found,
--- No phrases found,
--- Checked phrases for cycles, none found,
--- No templates found,
--- No template applications found, *** Warning: file
```

Test:

The interface features a vertical sidebar on the left with the following menu items: INTRODUCTION, SELECT DOMAIN, SELECT LESSON, USE THE APPLICATION, MY BADGES, and COMPILE & REDEPLOY. The main content area displays a progress bar with four steps, where 'Step 3: Test' is currently active. Below the progress bar is a terminal window showing the output of a test command, which includes several warnings about missing directories and files, and status reports for phrases, cycles, templates, and applications. A green progress indicator is visible below the terminal output.

# Uploading courses: RELEASE

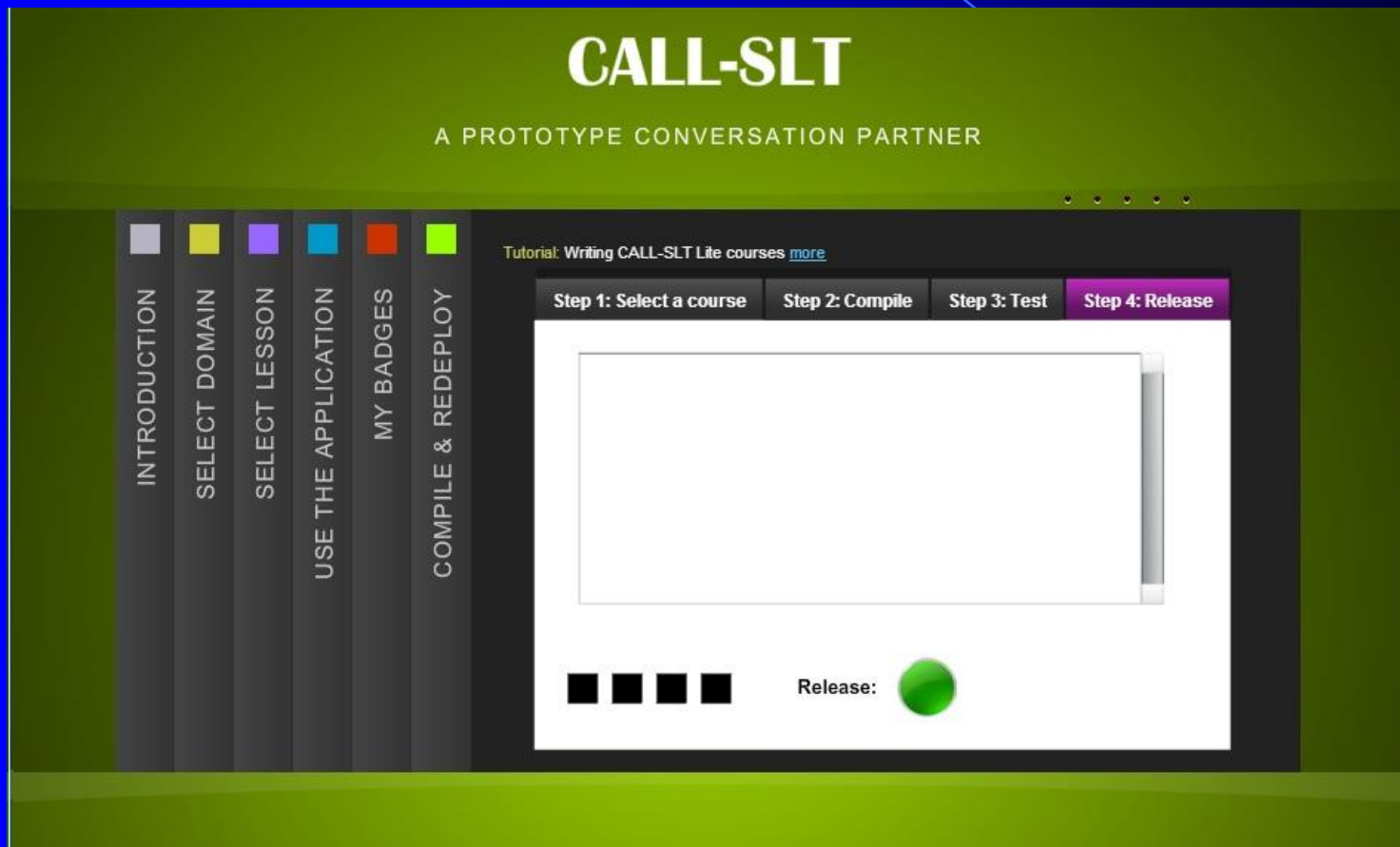
- What the user sees
  - Press Release button
  - Receive feedback



# Uploading courses: RELEASE

- What the user sees
  - Press Release button
  - Receive feedback
- What happens on server
  - Copies course dir to production server
  - Recompile all course descriptions
  - Redeploys system with new Nuance grammar

# Uploading courses: RELEASE



# Uploading courses: RELEASE

**CALL-SLT**  
A PROTOTYPE CONVERSATION PARTNER

Tutorial: [Writing CALL-SLT Lite courses](#) [more](#)

**Step 1: Select a course** | **Step 2: Compile** | **Step 3: Test** | **Step 4: Release**

INTRODUCTION | SELECT DOMAIN | SELECT LESSON | USE THE APPLICATION | MY BADGES | **COMPILE & REDEPLOY**

```
---[ok,  
--- Read Lite file (3035 lines, encoding = ISO-8859-1)  
c:/speechtranslation/callslt-code/trunk/call-  
slt/litecontent/english_course/english_course/grammars/english.txt,  
--- No phrases found,  
--- Checked phrases for cycles, none found,  
--- 55 templates internalised,  
--- 494 template applications expanded, *** Warning: line 2690 ("  
...") in file is longer than 50000 chars, truncating to 200,  
--- Read Lite file (3434 lines, encoding = UTF-16LE)  
c:/speechtranslation/callslt-code/trunk/call-
```

Release:

# Overview

- CALL-SLT → Open CALL-SLT
- Constructing CALL-SLT courses
- Deploying CALL-SLT courses
- And next...

# So what now?

- Just about to begin alpha testing
- External users starting next week
- First courses
  - Practice academic terminology
  - Shopping in Geneva
  - Telephone conversation

# So what now?

- What courses can users build?
- Can they be deployed on an interesting scale?
- We should know more soon

To learn more...

<http://callslt.unige.ch/demos-and-resources/>

Thank you!