

Lea's Box Learning Landscape Documentation

1. Introduction

For Lea's Box, we developed a landscape view to visualise in a sophisticated way the learning progress of students, as presented in Figure 1. The visualisation presents both, already visited learning objects necessary to gain new skills of competences on a specific topic (e.g. algebra in maths) and corresponding positive completed test items.

The visualisation is complemented with an interactive tag cloud for an intuitive navigation through the landscapes.

The Lea's Box Landscape consists of the following two tools:

- LandscapeGeneratorTool
- LandscapesVisTool

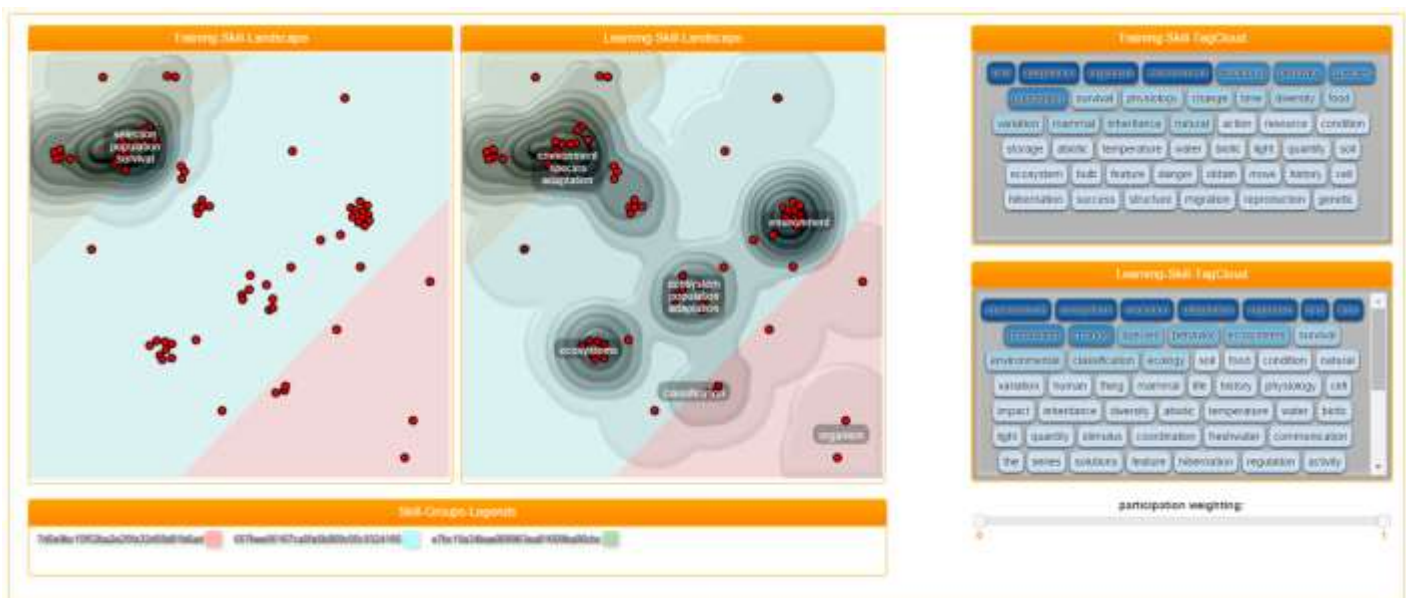


Figure 1: Lea's Box Landscape View

2. LandscapeGeneratorTool

The LandscapeGeneratorTool uses as input a JSON array representing skill-data. It generates two JSON files, which serve as input for the LandScapeVisTool.

This tool is packaged in the form of an executable JAR file (LandscapePipelineServer.jar). It requires a JAVA SDK or JAVA Runtime Engine version 1.7 or higher. The executable class (SkillLandscapeGenerator.java) takes two arguments. The first argument is the path to an input file in json format. The second argument is the path to a folder which will hold the two generated files.

JSON-Structure

The input-file in JSON format has the following structure:

```
{
  "totalStudents": 105,
  "skillGroups": [
    {
      "Country": "Texas ",
      "Subject": " Science ",
      "Topic": "Adaptations",
      "Concept": " 7th Grade Science ",
      "SkillGroup_ID": "e7bc19a34bae068963ea81609ba06cbc",
      "SkillGroup": "Biological Adaptations: Bird Beaks",
      "Skills": [
        {
          "Skill_ID": "0F0176C0-2B33-11E0-AF83-54F7BCAA0334",
          "Skill": "Change over Time",
          "LO_ID": ["8a675c", "7001d2e5c8658b"],
          "LOweight": [ "99","100"],
          "TI_ID": [ "question0","question1"],
          "TIweight": [ "91", "88" ]
        },
        {
          "Skill_ID": "0F03ECB6-2B33-11E0-AF83-54F7BCAA0334",
          "Skill": "Natural Selection in Action",
          "LO_ID": [ "8a675c4b397417bf96bb7fab780d192b" ],
          "LOweight": [ "99"],
          "TI_ID": ["question0","question1"],
          "TIweight": ["98", "88" ]
        }
      ]
    }
  ]
}
```

Example: Javacall for executing the LandscapeGeneratorTool on a linux environment:

```
java -cp LandscapePipelineServer.jar
"landscape/SkillLandscapeGenerator/input.json" "./"
```

3. LandscapesVisTool

The LandScapeVisTool visualises two interactive landscapes, one for Learning Objects and one for Test Items as well as the corresponding tag clouds for each landscape. The LandScapeVisTool needs to be executed on a web server.

Setting the InputFiles for the Visualisation

The LandScapeVisTool needs to be executed on a web server.

For setting the paths to the necessary two input files, enter the main.js file and go to the method "renderLandscapes". With the two variables "trainingLandscapePath" and "learningLandscapePath" the path can be set:

```
var renderLandscapes = function () {  
    var trainingLandscapePath = "datasets/trainingLandscape.json"  
    var learningLandscapePath = "datasets/learningLandscape.json"  
    ...  
}
```

Landscape User Interface

The Landscape User Interface can be categorized into four components: landscape view, tag-clouds, legends for the skill groups and skill-weighting filter. All these components will be described below in more detail.

Landscapes

The landscape (see Figure 2) is built by using scalable Vector Graphics (SVG). SVG-Filters and Shaddow have been used to present the landscape in a 3D-effect. The skill-groups are rendered as voronoi-areas using different colours to make them distinctive at one glance. The individual skills are always positioned within the corresponding voronoi-area. The height of the hills depends on the weighting of each individual skills. Similar skills, computed with a similarity analysis are presented closely together in the respective area.

The following features for interaction are implemented:

- skills-selection: select an area of skills by using brush
- skill-details: by hovering the mouse pointer over a skill the details (id+description) will be shown in a caption
- label-repositioning: by dragging using mouse
- zooming : use mouse wheel to zoom in or zoom out
- panning : strg + dragging using the mouse

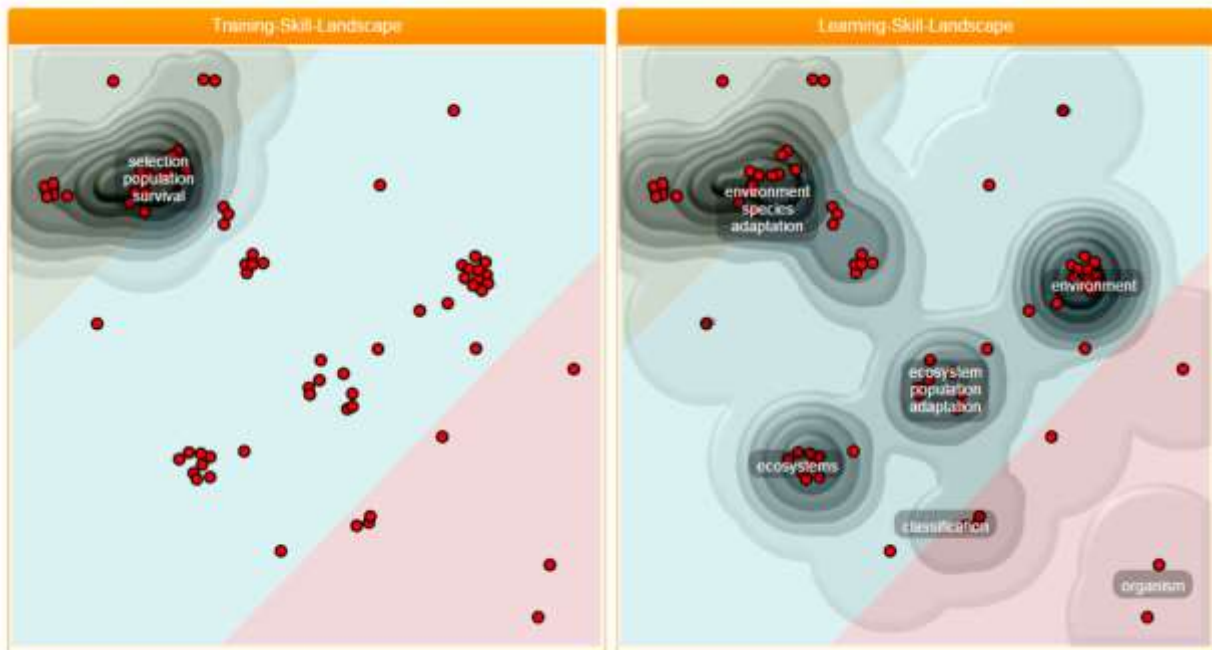


Figure 2: Landscape View

Tag-Clouds

TF-IDF method and weighting of the skills is used to calculate the relevance of keywords. Figure 3 presents the tag cloud of the training skills. The darker the colour the more relevant is the keyword. High relevant keywords are positioned on top of the tag cloud. By hovering the mouse pointer over a keyword, all skills containing the keyword in their description are highlighted in the corresponding landscape.



Figure 3: Tag Cloud

The following interactions lead to an update of the tag-clouds

- selecting skills in the landscape using brush
- de/selecting legends (skill-groups)
- after changing the weighting and lifting up the left mouse-button

Legends (skill-groups)

Each legend entry represents one of the available skill groups (see Figure 4).

The description of a skill-group is used to describe a legend, by moving over the description, the Skill Group ID is presented. The colours of the square beside the skill group description shows its association with the corresponding voronoi-area in the both landscapes.

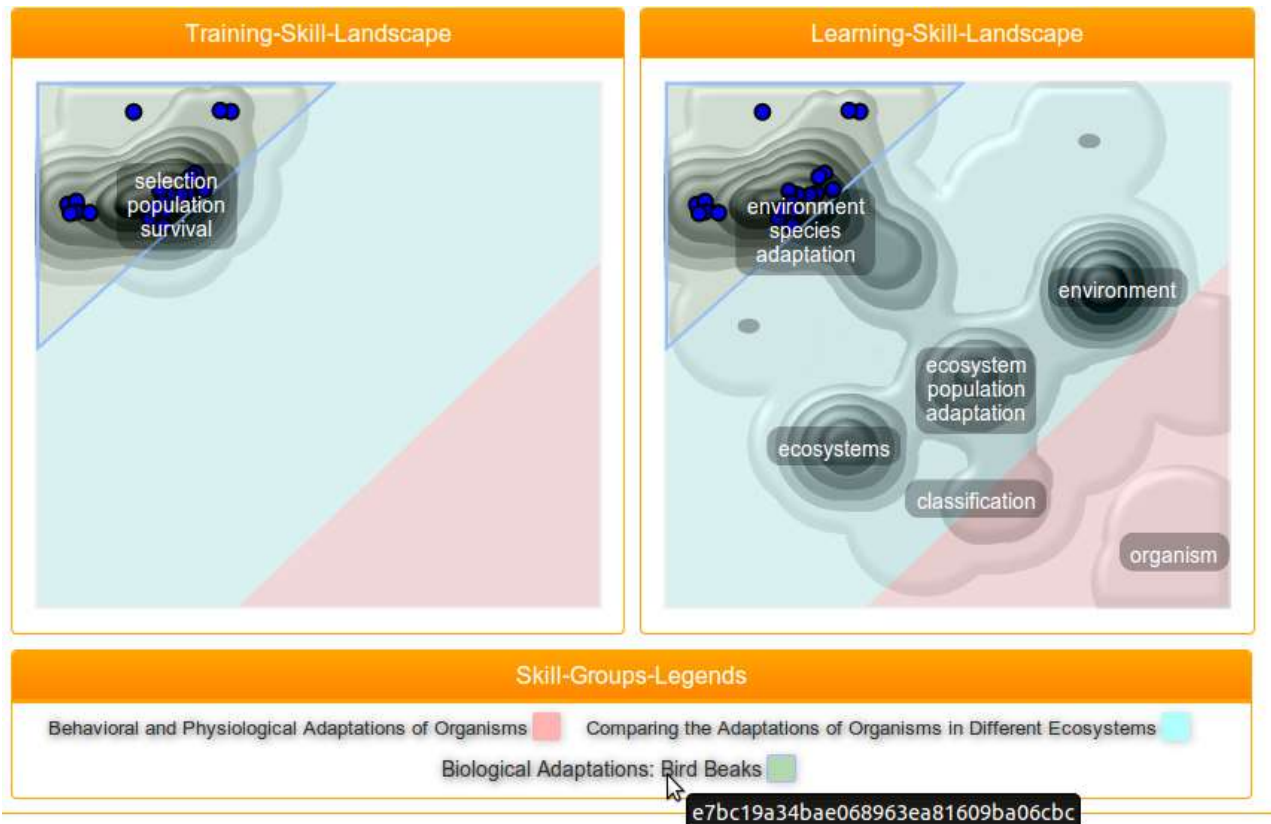


Figure 4: Legends and updating

Following interactions can be performed:

- Mouse-Over
 - the id of the skill-group is shown in a caption
 - highlighting corresponding voronoi-areas in both landscape
 - highlighting corresponding skills within voronoi-area as well
- Legend Selection (Mouse-Click)
 - highlighting corresponding voronoi-areas along with other selected legends in both landscape
 - highlighting corresponding skills within selected voronoi-areas as well
 - updating both tag-clouds
 - hiding all skills of skill-groups which are not selected
- Legend DeSelection
 - updating both tag-clouds
 - hiding corresponding skills

Weighting (Participation/Skill)

For each single skill two weights are calculated, based on the learning objects visited and the test items correctly answered. The weights are calculated as follows:

- **Weight for learning objects :**
visitingStudents = total number of students visited learning objects
learningObjects = number of learning objects
weight = $\text{visitingStudents} / (\text{learningObjects} * \text{total number of students})$
- **Weight for test items :**
answeringStudents = total number of students answered test items correctly
testItems = number of test items
weight = $\text{answeringStudents} / (\text{testItems} * \text{total number of students})$

The weighting slider can be used for filtering the skills in specific range of weighting (see Figure 5). Once the user has selected the range and lifted up the left-mouse-button the both tag-clouds are updated.

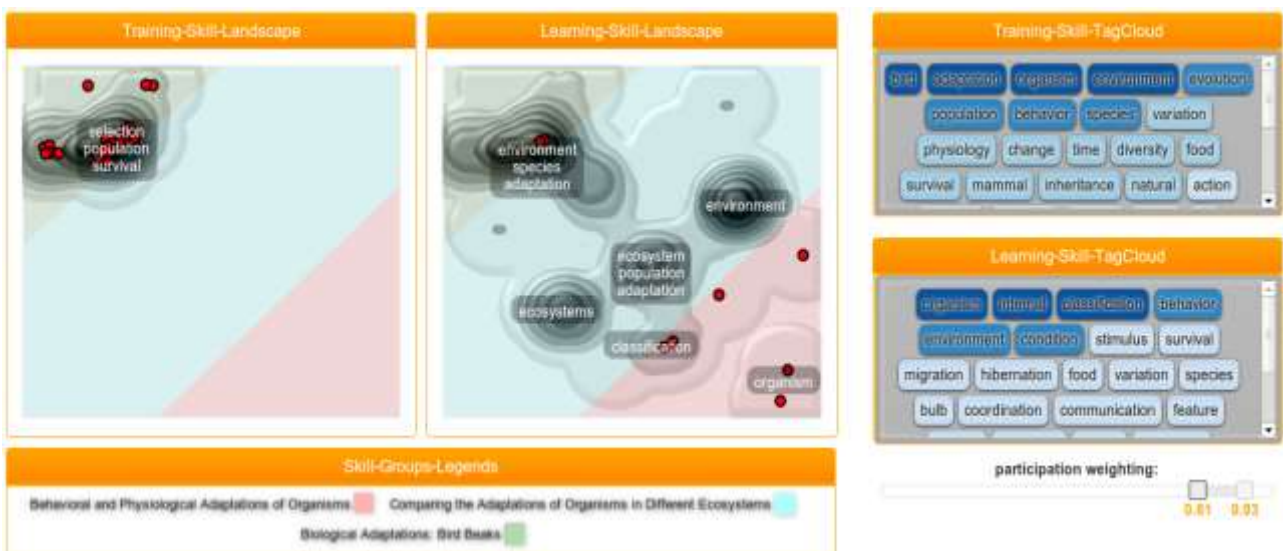


Figure 5: Weighting slider

Time Lines

The timeline visualizations show the temporal dynamics of the 'training-skill landscape' as well as the 'learning-skill landscape'. Each line in the two charts reflects the course of a particular object (either learning or assessment object). By default all objects are selected. All filters and selections applied to the main charts affect the time line charts as well.

