Curriculum Planner Service
Specification and Implementation

For a learner it is difficult to find a path of learner activities yielding a certain learning goal. The desired path depends on various variable, be it the amount of time available, the knowledge the learner already has, the difficulty of the learning objects and so on. The TENCompetence Curriculum Planner Service provide means to compute a sequence of learning objects tailored to the learners needs. Its reasoning bases on a set of learning activities metadata specifying the pre and post conditions of each learning activity, the number of credit points a learning activity has, and the learner profile, i.e., the knowledge of the learner and the courses she already attended. This document describes the functionality of the Curriculum Planner Service and the way it has to be called.
# Table of Contents

1 Introduction..............................................................................................................................1  
2 Input data structures................................................................................................................1  
  2.1 Datasets of learning activities to plan on........................................................................1  
  2.2 RDF representation of the learner’s profile.................................................................2  
  2.3 RDF representation of the actual request..........................................................................3  
3 Result format...............................................................................................................................4  
4 API Specification ......................................................................................................................6  
  4.1 Method specification..........................................................................................................6  
    4.1.1 Perform a request for curricula................................................................................6
### About this Document

<table>
<thead>
<tr>
<th>Title:</th>
<th>Curriculum Planner Service - Specification and Implementation</th>
</tr>
</thead>
</table>
| Authors:            | Philipp Kärger (kaerger@L3S.de)  
                      | Daniel Olmedilla (olmedilla@L3S.de)  
                      | Ingo Brunkhorst (brunkhorst@L3S.de) |
| Version (Date):     | 1.0 (2007-17-10)                                              |
| Document Location:  |                                                               |
1 Introduction

The Curriculum Planner Service described in this document is a Personalization Service of the TENCompetence System. It is exploiting the requirement information of a learning activity to assemble a sequence of courses (i.e., a curriculum) which guides the learner from her current state of knowledge towards her learning goal. In order to allow the TENCompetence System to plan arbitrary curricula, the Curriculum Planner Service has been made available as a Web Service at http://in.l3s.uni-hannover.de:3151/CPS. The Web Service's task is to provide a set of sequences of learning activities given a learner’s profile, a dataset of available learning activities, and the learner's learning goal.

The sources and all needed libraries are checked in in the TENCompetence CVS Server on Sourceforge under the Module wp7/CurriculumPlannerService. In addition to the Web Service implementation, a java package is provided supporting developers to implement a client application for the Curriculum Planner Service.

2 Basic Requirements

2.1 Requirements for a Client

In order to call the Curriculum Planner Service any SOAP Web Service Client application can be used. In the TenCompetence CVS module an example client application using the Apache CXF Web Service Framework is located. This example client called examples.ExampleClient poses a query to the service and prints out the results.

2.2 Requirements for a Server

Due to the fact that the Curriculum Planner Service bases on SWI-Prolog, running the server needs an environment providing SWI-Prolog with the packages semweb/rdf_db, semweb/rdfs, and http/http_open. The call to the Prolog system is to be configured in the config file config/plcall.config.

3 Input data structures

In this section we describe the structure of the input data used for the Curriculum Planner Service. As input, the service requires the following information:

1. number of curricula to be computed
2. limit of credit points to be acquired in each curriculum
3. a URL of an RDF data set describing a set of learning activities to plan on
4. the student's profile comprising the activities the learner already attended and the knowledge she already has
5. the learner's goal, i.e., the knowledge the learner want to learn
This input information will be described in detail in the following sections.

3.1 Datasets of learning activities to plan on

For using this service we assume that the metadata of the learning activities upon which the user is going to perform a planning request is available at a URL. One example for such a repository is the collection of lectures held at the Computer Science Department of the University of Hannover which we used as a test data set. This data set can be found at http://www.l3s.de/~kaerger/rdf/curriculumCourses.rdf. In Figure 1, an example dataset with two learning activities is depicted. For each activities there is a list of pre-conditions and a list of post-conditions:

- **Pre-condition** is a list of knowledge items which a learner has to know about in order to be able to follow the course.
- **Post-condition** is a list of knowledge items which the learner knows about after attending the learning activity.
Figure 1: An example RDF Dataset comprising two learning activities

3.2 RDF representation of the learner’s profile

The learner profile comprises information about the activities the learner has already attended and her current knowledge. Figure 2 depicts an example of a learner’s profile. The current knowledge of the learner is represented as a list of knowledge items. The activities already attended by the learner are represented as a list of URIs pointing to the dataset as it is described in the previous section.
3.3 RDF representation of the actual request

The information about a request is described in an RDF request document. This document comprises:

- **Number of Solutions.** This number represents how many solutions (i.e., curricula or sequences of learning activities) the planner is told to produce at maximum.

- **Limit of Credit points.** In some cases, a course should not exceed a certain amount of credit points. For this purpose a request contains a limit of credit points the curriculum will contain. Given this number is 25, the planner will not produce any curriculum containing more than 25 credit points.

- **Learning Goal.** The learning goal is a list of knowledge items the learner wants to learn about. The curricula the planner will produce for a request will always provide learning activities teaching this learning goal and all the other intermediate goals which are needed in order to attend the final activity.

Figure 3 shows an example for an RDF request document.
Figure 3 An RDF request asking for three curricula not exceeding 35 credit points and teaching the learning goals given in node A0.

4 Result format

The curricula created from the planning service are transformed into an RDF result document and sent back to the requester. For each curriculum planned, the result contains a sequence of learning activity identifiers pointing to the learning activities listed in the dataset (as described in Section 2.1). See Figure 4 for an example.
Figure 4 An RDF result showing two curricula.
5 API Specification

The location of the service is http://in.l3s.uni-hannover.de:3151/CPS. The service’s wsdl file is located at: http://in.l3s.uni-hannover.de:3151/CPS?wsdl.

5.1 Method specification

5.1.1 Perform a request for curricula

Used to pose a request. Returns an RDF description of the curricula with the conditions specified in the request.

<table>
<thead>
<tr>
<th>Method name</th>
<th>getCurricula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return type</td>
<td>String</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>profile</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>databaseURL</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>