

Open Virtual Mobility

O3-A1.5: Competency Directory

Final prototype and report

Outcome 3 Activity A1.5 Competency Directory	
Document submission and review information	
Declared due date of deliverable	01.06.2020
Reviewed due date of deliverable	04.09.2020
Actual submission date	08.09.2020
Organisation name of lead contractor	Beuth University of Applied Sciences, Berlin
Revision	1.2
Author and reviewer information	
Name of the author	Johannes Konert
Affiliation of the author	Beuth University of Applied Sciences, Berlin
Name of the reviewer	Diana And one, Vlad Mihaescu
Affiliation of the reviewer	UNIVERSITATEA POLITEHNICA TIMISOARA

Copyright licence: This work is licensed under a Free Culture Licence [Creative Commons Attribution-ShareAlike 4.0 International License](#).

*The creation of these resources has been (partially) funded by the ERASMUS+ grant program of the European Union under grant no. **2017-1-DE01-KA203-003494**. Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.*

Imprint

Imprint: This publication is O3-A1.5 of the Open Virtual Mobility Erasmus+ strategic partnership founded by the European Commission 2017 - 2020 under **2017-1-DE01-KA203-003494**, URL: <https://www.openvirtualmobility.eu/>

This paper is a final summary of the publications about the creation of the competency directory and the decisions made within the Open Virtual Mobility partnership. This includes a short and updated version of the installation manual and user manual.

PDF download

A full-text PDF of this report is available as a free download from:
<https://www.openvirtualmobility.eu/>

Social media

Find us on Twitter: @openVM_erasmus

Give us your feedback on social media with the following hashtag: #openvirtualmobility

Suggested citation

Konert, J. (2020) O3-A1.5: Competency Directory Final Prototype and Report. Open Virtual Mobility Erasmus+ (2017-2020). Berlin, Germany. Retrieved from <https://www.openvirtualmobility.eu/topics/outputs>

Corresponding author

Johannes Konert
Beuth University of Applied Sciences
Luxemburger Straße 10, 13353 Berlin, Germany
johannes.konert[at]beuth-hochschule[dot]de

Table of Contents

Executive summary	4
What are the objectives of this paper?	4
Who is this paper for?	4
What topics are addressed in this paper?	4
Contributors	4
Acknowledgements	5
1. Aims and Scope	6
2. Requirement analysis methodology and results	6
2.1 Requirement analysis	6
Survey results	6
Discussion and prioritization	8
C3: levels (as advancements in ~same~ competency)	8
C11: search by framework	8
C12: synchronization/copy of the competency framework into Moodle	8
C14: mobile interface support (small displays)	8
C16: criteria definitions for a competency	9
2.2 Desired functionality of the prototypical implementation	9
3. Guideline on setup of own competency directory	10
3.1 System setup requirements	10
3.2. Fetch the source code	10
3.3 Build a fresh version of the frontend UI	11
3.4 Build the API-backend server	11
3.5 Edit configuration	11
3.6 Edit docker-compose configuration	11
3.7 Run the server	13
4. Manual of using the existing competency directory	13
Comment on editor functionality for competency directory	15
5. Using URLs of competencies with Open Badges	16
Using competency mapping in learning design	16
6. Conclusion and outlook	17
7. References	18
Attachments	18

Executive summary

The competency directory is an Open Source development published, maintained and operated by the Open Virtual Mobility project. One instance is available at <http://cd.openvirtualmobility.eu>, which provides a web interface to the 33 competency and skill definitions of the Open Virtual Mobility skills framework. This document includes the methodology description and results of requirement analysis and the desired functionality. Additionally, it describes the setup of own instances using the Open Source code, explains the functionality and usage of the web frontend including the benefits of the unique deep links to entries which can be used as referencing URLs to either see the website or fetch machine readable data in JSON-LD format about the competencies. Overall, this document provides several guidelines to set up and use the competency directory, especially in use with Open Badges.

What are the objectives of this paper?

- Report on the requirement analysis and decisions made
- Guidelines on setup of competency directory
- Manual of competency directory

Who is this paper for?

- Technicians interested in a.) using semantic competency definitions or b.) setting up an own competency directory.
- Pedagogues and didacticians interested in the technical infrastructure behind semantic competency definitions.

What topics are addressed in this paper?

Requirements, Functionality, Setup, REST API, Frontend, Languages, References, IRIs, Manual, Open Badges

Contributors

- Johannes Konert, Beuth University of Applied Sciences, Germany
- Vlad Mihaescu, Centru Multimedia, Universitatea Politehnica Timisoara, Romania

Acknowledgements

Essential parts of this document are supported by scientific work, e.g. (Konert, Buchem, Stoye, 2019). We thank the corresponding authors for the scientific contribution and open source code releases.

1. Aims and Scope

This document should serve as a summary and overview document, based on a retrospective in the conducted project and its results with a focus on the O3.A1 Competency Directory designed, created, running and integrated into the infrastructure of the Open Virtual Mobility project. The summarized requirement analysis sheds light on the decisions taken. The guideline should serve as a step-by-step manual for interested parties to use the competency directory on their own. Two scenarios are possible: First, download, installation and setup of the Open Source¹ release of the competency directory to provide own semantically cross-referenced competency definitions (of whatever competency framework). Second, use of the existing, maintained and running competency directory at <http://cd.openvirtualmobility.eu> with currently 33 entries of relevant competencies for virtual mobility as defined in the project Output 1.

In both scenarios the REST API can be used to directly reference the competency definitions from anywhere by unique URLs. As an example, the use with Open Badge Standard 2.0 (alignment field) is described.

For details about the semantic vocabulary used, the REST-API endpoints and the overall system structure refer to former documents, e.g. *O3-A1.3 Competency directory - functional prototype*.

2. Requirement analysis methodology and results

2.1 Requirement analysis

The collection of requirements to the competency directory were described in the prior document O3-A1.1 (see Appendix A1 *Survey items for requirements analysis*). The survey among all partner organizations was conducted as an online survey based on Google Sheets between 12.03.-12.04.2018. Six participants responded.

Survey results

The resulting answers are encoded as [-1 I disagree to have that in the project, 0 not needed, 1 optional but useful, 2 needed uncritical, 3 definitely needed] and were summed up per item². The result, ranked by sum, is shown in Table 1.

¹ <https://github.com/openVirtualMobility/competency-directory>

² The four Likert scale values have a number and a text value and are thus classified as ordinal values, but not strictly intervals. The calculation of sums is thus not a robust reliable measure to build upon. Therefore, interested readers find the raw answers in Appendix A2 Survey result table. The sum is used only as an indicator and basis for the sorting and discussion. We refrain from calculating standard deviations or distances.

Table 1: Ranked requirements from survey. Colored in green are the candidates which are kept as requirements, in yellow are optional non-critical requirements, in red are those dropped.

ID	Requirement(s)	SUM
C1 → R1	list of all competencies	18
C2 → R2	search by keyword	17
C3	levels (as advancements in ~same~ competency)	17
C4 → R3	unique URL to individual competency	17
C5 → R4	multi-language support (other interfaces than English)	16
C6 → R5	direct access by ID	15
C7 → R14	browsing by exploring search results ...	15
C8 → R6	description (for each competency)	15
C9 → R7	cross-referencing of competencies (similarity, includes, ..) within the VM competency framework	15
C10 → R15	competency management (adding, editing, ...)	14
C11	search by framework	14
C12	synchronization/copy of the competency framework into Moodle	14
C13 → R16	direct editing of competencies (add, edit, remove)	13
C14	mobile interface support (small displays)	13
C15 → R8	search by existing language translation	13
C16	criteria definitions for a competency	13
C17 → R9	low latency on response to requests (<200ms)	13
C18 → R10	encrypted communication via HTTPS	13
C19 → R11	REST-Level 2 conform HTTP-API	13
C20 → R12	JSON-LD format for input/output	13
C21 → R13	multi-language support for competency definitions (same competency in x languages)	12
C22 → R17	reference editing (similar/contains/prerequisite ..)	12
C23 → R18	translation support (web interface)	12
C24	cross-referencing of competencies with other frameworks	12
C25	evidence definitions for a competency ...	12
C26	identity management (login, credentials)	11
C27	read/write to an xAPI	11
C28	role management (for limited editing, reading of competencies)	10
C29	multiple competency framework support	10
C30	search by country of origin (of the framework)	10
C31	Service Level Agreement (SLA) as accessibility of a high level (95%)	10
C32	assertion management (sets of competencies per user)	9
C33	RDFa format for input/output	9
C34	import of complete other competency frameworks from different sources	8
C35	SOAP Webservice conform HTTP-API	8

Discussion and prioritization

Table 1 was amended by an ID for easier reference in the text. All possible candidates for requirements are named C1-C25. The candidates which are copied as a requirement for the competency directory are given additional IDs R1-R18. Must-Have requirements of higher priority are marked green (R1-R13) and the requirements which are not essential and nice-to-have (lower priority) are marked yellow (R14-18).

Before the list of requirements R1-18 was finalized the raw survey results were presented in a transnational partner meeting in Timisoara (24.09.2018). For some of the items, the rated importance and sum values were quite surprising, for the author of this document as well as for the partner members present. Likewise, for some items, the range of ratings spread the full spectrum from -1 (I disagree to have that in the project) to 3 (definitely needed), as listed in earlier publication O3.A1.3--Appendix A2 Survey result table.

In the following all lower or non-prioritized candidates are briefly discussed, as they are later not taken into consideration any more (marked yellow and red in Table 1).

C3: levels (as advancements in ~same~ competency)

C3 was rejected as the competency framework created for open virtual mobility skills (Output O1) contains no levels. Additionally, the European Skills/Competences, Qualifications and Occupations Framework (ESCO) has no levels in its scheme³. During discussion, all partners agreed that instead of levels, the (existing) concept of references of type “is a prerequisite of” is used.

C11: search by framework

The competency directory will merely contain competency definitions of one framework – the open virtual mobility competency framework. It could technically support cross-referencing to external frameworks (e.g. ESCO)(C24), but all organizations’ members at the meeting agreed that this is not a needed requirement for any of the outputs (O1-O7).

C12: synchronization/copy of the competency framework into Moodle

As neither the eAssessment parts (O4) nor the OERs or MOOC sections (O6) will cross-reference directly to the competency areas or individual skill-sets of the open virtual mobility framework (from O1) there is no use for the project consortium of having a copy or synchronization of the competency definitions from the competency directory into Moodle competency lists.

C14: mobile interface support (small displays)

The competency directory will not be used by end-users, but primarily by project partners while defining open badges. During the badge design a screen with high resolution is used to fill all the references and text fields for a badge. In this scenario, from the found (C2) competency definitions (C8) the unique URLs are copied (C4/C6) into the AlignmentObject⁴ field(s) of the Open Badge standard. No mobile (smartphone) support of the user-interface is required (even though most web-design approaches implement a mobile-first approach anyway).

³ https://ec.europa.eu/esco/portal/escopedia/Skills_pillar , last accessed 05.03.2019 (currently with breaking changes in newer versions unfortunately)

⁴ <https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/index.html#Alignment>, last accessed 05.03.2019

C16: criteria definitions for a competency

Like C25 (*evidence definitions for a competency*) criteria are needed to decide whether or not an individual has shown the competency, and this can receive a certificate for it. The competency directory is meant to list competencies for open virtual mobility with as few context dependencies as possible. Listing of criteria to fulfill for a certain competency (or listing of evidence needed to be shown) is beyond that scope and should better be defined in the context (e.g. OER, MOOC or assessment) where the learner wants to proof his competency and needs to fulfill certain requirements (criteria to pass, evidence to show).

The following candidates are all related to editing competency definitions or extending the existing framework. All discussion partners agreed that this would be nice-to-have, but practically there is no need for editing functionality as the resulting competency framework from Output O1 has been created based on the group concept mapping with many experts and the partnership does not expect to change the definition in near future.

- C10 → R15 (competency management (adding, editing, ...))
- C13 → R16 (direct editing of competencies (add, edit, remove))
- C22 → R17 (reference editing (similar/contains/prerequisite))
- C23 → R18 (translation support (web))

All other candidates (C26-C35) received a score below 12 and were agreed during the transnational partner meeting to be not considered for implementation.

2.2 Desired functionality of the prototypical implementation

The derived list of requirements (without any priority indicated by the order) is:

- R1 list of all competencies
- R2 search by keyword
- R3 unique URL to individual competency
- R4 multi-language support (other interfaces than English)
- R5 direct access by ID
- R6 description (for each competency)
- R7 cross-referencing of competencies (similarity, includes, ..) within the VM competency framework
- R8 search by existing language translation
- R9 low latency on response to requests (<200ms)
- R10 encrypted communication via HTTPS
- R11 REST-Level 2 conform HTTP-API
- R12 JSON-LD format for input/output
- R13 multi-language support for competency definitions (same competency in x languages)

The list of optional, nice to have functionalities is:

- R14 browsing by exploring search results ...
- R15 competency management (adding, editing, ...)
- R16 direct editing of competencies (add, edit, remove)
- R17 reference editing (similar/contains/prerequisite ..)
- R18 translation support (web interface)

From these, in the final prototype, R1, R2, R3, R4, R5, R6, R7, R8, R9, R11, R12, R13, R14 have been implemented and are available in the running prototype.

- Optional R10 (HTTPS based communication) is yet missing, but if needed, easy to add by free-to-use HTTPS-certificate solutions for the node.js server base⁵.
- Optional R15, R16, R17 (editing) themselves are implemented and functional, but as no proper user authentication could be finished, these were not yet integrated (but source code is published), see section 4 comment on editor functionality for details.
- Optional R18 (translation support) was not implemented due to project budget limitations.

3. Guideline on setup of own competency directory

Note: Readers interested in the discussion of related work, design process and decisions on the technologies chosen, are recommended to read earlier publication O3.A1.2 and O3.A1.3, available on the project website⁶

3.1 System setup requirements

The competency directory code can be run on any machine that supports Docker containers. Thus, install the Docker Engine⁷ and Docker Compose⁸ on your system. The source code itself needs NodeJS to serve the REST-API and to build the REACT-frontend. Install NodeJS v12⁹ or higher. Additionally, it is recommended to have GIT version control installed¹⁰ to fetch and update the source code directly from the repository. Alternatively, it can be downloaded manually.

3.2. Fetch the source code

The current repository of the competency directory can be found at GitHub¹. Either use the command

```
git clone https://github.com/openVirtualMobility/competency-directory.git
```

or manually download the ZIP¹¹ and unpack it. The benefit of using GIT is an easy update to the latest source code version by the command `git pull` which keeps your local changes, e.g. to configuration files or database. You should see at least the files and directories shown in Figure 1.

⁵ <https://letsencrypt.org/>

⁶ <https://openvirtualmobility.eu>

⁷ <https://docs.docker.com/install/>

⁸ <https://docs.docker.com/compose/install/>

⁹ <https://nodejs.org/en/download/>

¹⁰ <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

¹¹ <https://github.com/openVirtualMobility/competency-directory/archive/master.zip>

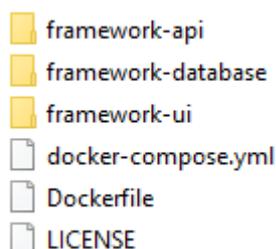


Figure 1: Source code of competency directory components (file/dir listing)

3.3 Build a fresh version of the frontend UI

Change into directory `\framework-ui\` and run `npm install`. It takes a while to fetch all needed modules. Afterwards run `npm run build`. This will create the folder `.\build` within the folder `\framework-ui\`. Copy the whole folder `build` into the folder `\framework-api\src\`.

Why? Originally the frontend was running as an independent component, but this caused issues with port forwarding and JavaScripts. Currently the best solution is to let the server, which provides the REST-API, deliver the frontend as well. This is automatically done when you copy your build to the `framework-api\src` folder (afterwards there is a folder `framework-api\src\build\` with all these files).

3.4 Build the API-backend server

Change into folder `\framework-api\` and run `npm install`. This will fetch all modules needed for the server backend.

3.5 Edit configuration

In the file `\framework-api\src\config.json` you need to change the currently set base URL to your server's URL. The format is as shown in Figure 2.

```

{
  "baseurl": "http://cd.openvirtualmobility.eu",
  "neo4j": "blabla"
}
  
```

Figure 2: config.json to set the base URL of the server

For security reasons change in the same file as well the password for your neo4j database-connection (from `"blabla"` to something secure). The same password needs to be set in your settings for docker-compose and the docker configuration for the database itself (see next section).

3.6 Edit docker-compose configuration

The file `\docker-compose.yml` contains the major information about which docker containers should be run and on which ports they will be available. As you see in Figure 3, you need to set the same password for the neo4j connection here (which is then used to create this neo4j authorization that is used by the server API from step 2.5 above).

```

version: "3"
services:
  server-app:
    build: ./framework-api
    image: framework-api
    ports:
      - "6060:6060"
    volumes:
      - ./framework-api/src:/app
    networks:
      - intranet
  db:
    build: ./framework-database
    image: framework-database
    environment:
      - NEO4J_AUTH=neo4j/blabla
    ports:
      - "7474:7474"
      - "7687:7687"
    networks:
      - intranet
networks:
  intranet:

```

Figure 3: docker-compose.yml file (here you need to change the password for neo4j database as well)

In the same manner, edit the `Dockerfile` for the docker container of the database itself in folder `\framework-database\` and change the password accordingly to the same one as above.

```

FROM neo4j:3.4.10
ENV NEO4J_AUTH=neo4j/blabla

```

Figure 4: Dockerfile of `\framework-database\` which is the third place to set the DB password

Be informed that the `Dockerfile` password is only read and used once on first startup (creation) of the docker container. In case you change it later, e.g. after testing the default settings, it won't be changed and connection to DB fails. Then, the fastest way is to simply delete the container named `framework-database` and let `docker-compose` recreate it (see next section about running the server with `docker-compose`).

3.7 Run the server

In your main installation folder (same level as the docker-compose.yml file) run the command `docker-compose up -d`

Your server is now available on port 3000. You can use a REST-API client like Postman or a web browser like Chrome to check if the server is running.

If you want to stop the containers, run

`docker-compose stop`

Note: For security reasons it is recommendable to use a webserver proxy as a HTTP service on a public domain server which accepts the HTTP requests (usually on port 80), forwards them internally to your local machine (that runs the competency directory on port 3000) and sends back the response to the client system. We recommend using Apache2 Reverse Proxy¹².

4. Manual of using the existing competency directory

When you access the web frontend the first time via a web browser you see the default search interface in English language (see Figure 5).

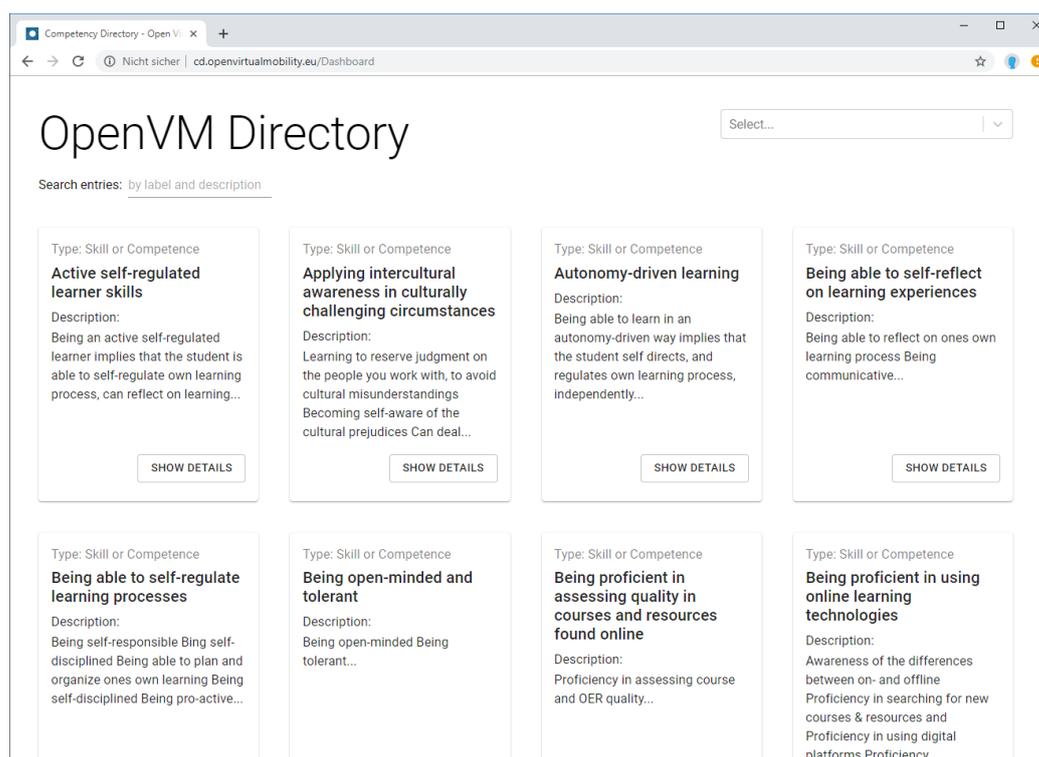


Figure 5: Main search interface of competency directory at <http://cd.openvirtualmobility.eu/>

By typing phrases into the search field on top, the list narrows directly to all entries that match these word-parts either in title or in description (see Figure 6, left). The order of the entries is decided by relevance. This is calculated by major weights for title matches and secondary by description matches.

¹² https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html

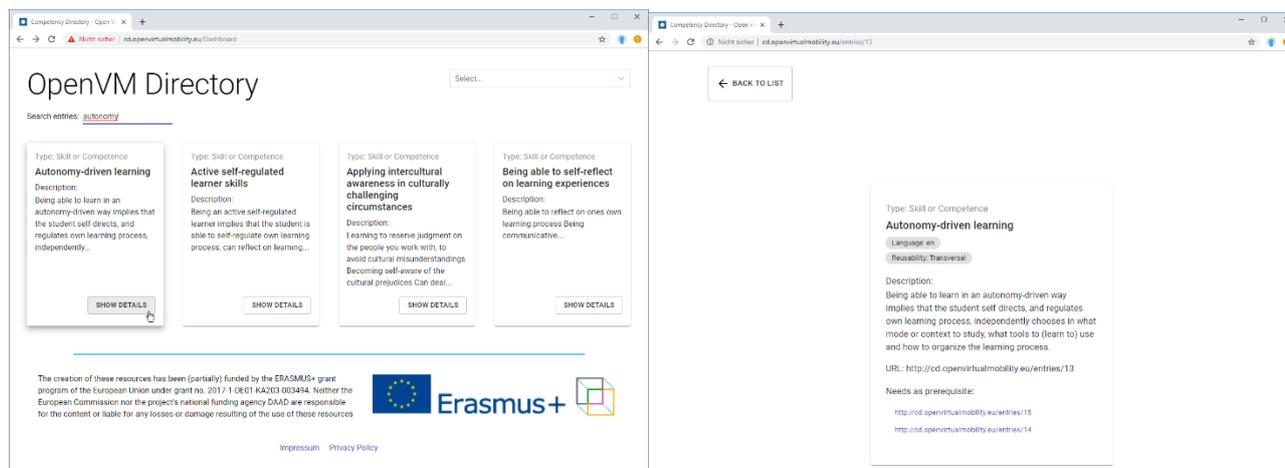


Figure 6: Search interface narrows down the list of matching entries while typing

When you click on one of the SHOW DETAILS buttons, the corresponding entry is opened solely, and the URL that can be used to deep-link to this competency is visible. In Figure 5 (left side) you see Entry ID 13 Autonomy-driven learning with URL <http://cd.openvirtualmobility.eu/entries/13>. This URL can be used in documents to cross-reference the proper competency descriptions. When a user opens this URL via a web browser the web frontend is shown as in Figure 5 (right side). It is also possible to use these URLs with a client-system that expects machine-readable semantic data in JSON-LD format. If the accept header of the HTTP-request is set to application/json the competency entries are delivered in JSON-LD instead (Example for the same entry with ID 13 is shown in Figure 7).

```
{
  "@context": "http://cd.openvirtualmobility.eu/context/",
  "id": "http://cd.openvirtualmobility.eu/entries/13",
  "language": "en",
  "isEssentialPartOf": [],
  "isOptionalPartOf": [],
  "isSameAs": [],
  "isSimilarTo": [],
  "needsAsPrerequisite": [
    "http://cd.openvirtualmobility.eu/entries/15",
    "http://cd.openvirtualmobility.eu/entries/14"
  ],
  "skillReuseLevel": "1 Transversal",
  "skillType": "Skill or Competence",
  "description": {
    "language": "en",
    "value": "Being able to learn in an autonomy-driven way implies that the student self directs, and regulates own learning process, independently chooses in what mode or context to study, what tools to (learn to) use and how to organize the learning process."
  },
  "altLabel": [],
  "prefLabel": {
    "language": "en",
    "value": "Autonomy-driven learning"
  }
}
```

Figure 7: HTTP response for <http://cd.openvirtualmobility.eu/entries/13> with accept-header set to application/json

Finally, the competency directory frontend supports all partner organizations' languages besides English (Dutch, German, Italian, French, Romanian, Spanish). The interface language can be chosen by the dropdown on the upper right corner (see Figure 8). The database of competency entries

supports multi-language entries as well, but currently the competency definitions are available in English only. If the database would contain translated entries, the search results will reflect the language selected from the interface, too. If no translation is available, the English entries are shown. Once the language has been selected, the browser is set to remember it on the next visits to the web frontend.

Note on missing translated entries: During the last 6 months of the project period the translations of the competencies were created as a joint effort of all partner organizations and are available as reported by Output O1. Unfortunately, it was not integrated into the database any more as the provided format needed some manual adjustment which was prioritized lower than other duties. Still, with reasonable effort it is possible to add the multi-language entries, as the database structure and frontend is prepared for this (e.g. as seen in Figure 8).

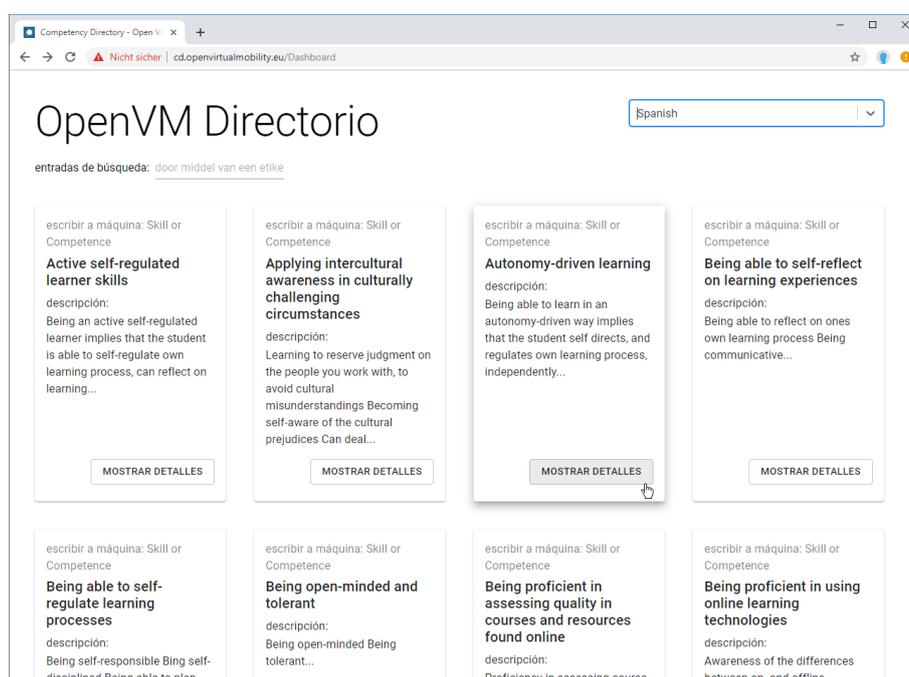


Figure 8: Interface language selection

Comment on editor functionality for competency directory

As the title of the milestone O3.A1.4 expresses, editor functionality (to update competencies via web frontend) was originally planned, too. Due to the vanishing demand from the project consortium due to shifting project priorities, this part was delayed and lower prioritized. Significantly more time investment was needed for the joint establishing and deriving of the Open Virtual Mobility competency framework (see Output O1 reports) and the envisioned ESCO mapping of the newly created competency framework.

Still, the editor functionality is finished in an prototypical state but no user authentication (login, profile page) has been securely finished, yet. The editor functionality itself is ready to use (to edit, add or delete competencies and their relations within the database) and can be inspected in the version control system GIT, branch 'editor'¹³.

Likewise, the ESCO mapping has been finished during the last months of the project period. For details please refer to Output O1 reports. Due to project time and budget limitations these

¹³ <https://github.com/openVirtualMobility/competency-directory/tree/editor>

mappings have not been added to the database of the running instance at <http://cd.openvirtualmobility.eu> until the end of the project, but are planned to be added soon afterwards.

5. Using URLs of competencies with Open Badges

One additional application scenario to use the direct links to competency entries, is the interlinking of Open Badges with the corresponding competency definitions. If an Open Badge is designed as a digital certificate for a certain competency or set of competencies, the Open Badge Standard allows (since Version 2.0) to reference the competency definitions as so called `AlignmentObjects`. These are a set of well-defined fields in the metadata of an Open Badge as listed in Table 1.

Table 1: Metadata fields for `AlignmentObjects` of Open Badges (source: IMS Open Badge Spec 2.0¹⁴). Fields marked * are mandatory.

Property	Expected Type	Description
<code>targetName</code> *	Text	Name of the alignment.
<code>targetUrl</code> *	URL	URL linking to the official description of the alignment target, for example an individual standard within an educational framework.
<code>targetDescription</code>	Text	Short description of the alignment target.
<code>targetFramework</code>	Text	Name of the framework of the alignment target.
<code>targetCode</code>	Text	If applicable, a locally unique string identifier that identifies the alignment target within its framework and/or <code>targetUrl</code> .

As `targetName` we recommend to use the title of the competency (the name, e.g. “Autonomy-driven learning”) and as `targetUrl` the direct URL to the competency entry from the competency directory (e.g. <http://cd.openvirtualmobility.eu/entries/13>). Optionally, the `targetDescription` can be copied, the `targetFramework` is the name of the competency framework (e.g. Open Virtual Mobility skills) and the `targetCode` is the ID of the entry (e.g. 13).

Using competency mapping in learning design

There are manifold benefits of using the `AlignmentObjects` to reference competencies from Open Badges. First, if later on, the text of the competency is updated or new cross-references (dependencies, successors etc.) are defined, these are automatically visible to the Open Badge user because data is no longer copied but referenced. Second, if several Open Badges reference the same competencies, these machine-readable information allow detection of similar/identical Open Badges, coverage of competency/skill sets and much more. Third, applications that visualize Open Badges can use this metadata to fetch the competency information (as JSON-LD data), display it,

¹⁴ <https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/index.html#Alignment>

aggregate the information and navigate the graph of cross-referenced competency definitions. Fourth (and finally for now), users can follow these hyperlinks to competency definitions and browse the definition (in several languages), find out what prerequisites are defined, what other competencies are similar to this one and which other entries the competency framework offers that provides such an interactive competency directory frontend (as described in section 3 of this document).

Open Badges, which are issued by the OpenVM project, are hosted by project partner Cineca's Bestr platform and all are based on Open Badge Standard 2.0, all contain the proper metadata `AlignmentObject` information for corresponding competency definitions. For a list of existing Open Badges in Open Virtual Mobility see <https://bestr.it/project/show/107#!#badge>.

6. Conclusion and outlook

This document provided a summarizing overview over the requirements derived from the projects' aims and the requirement analysis conducted during the first project year. Based on the three-tier architecture design with REST-API-based backend, database and single page frontend a good basis has been established and been brought into practice to integrate and align learning activities, learning resources and digital credentials to semantic competency definitions. A step-by-step guideline has been provided about setting up the competency directory to offer one's own competency framework with a REST-API and web interface. Additionally, the use of the frontend, search interface, multi-language interface and deep links was explained for interested end users. As an example, the up and running instance at <http://cd.openvirtualmobility.eu> was shown in the screenshots. This instance currently hosts all 33 competency and skill definitions of the Open Virtual Mobility skills framework. It is integrated into the Open Virtual Mobility infrastructure, e.g. the unique URLs are used by the Open Badges, which learners can achieve while taking online courses in the Open Virtual Mobility Learning Hub¹⁵.

A more technical insight was given into the use of the unique URLs for entries to fetch the data in JSON-LD semantic and machine-readable format. Therefore, the unique URLs to competency entries can serve as well as identifiers and can be used as references in the `AlignmentObject` of Open Badges to point to the corresponding competency definitions. This has been explained in the last section of the document.

The published source code¹⁶ of the competency directory is actively maintained until the end of the project and beyond as long as the Open Source community supports these efforts.

¹⁵ <http://hub.openvirtualmobility.eu>

¹⁶ <https://github.com/openVirtualMobility/competency-directory>

7. References

References to tools and software are as footnotes directly in the text.

- Buchem Ilona & Konert Johannes (2020). *Semantic Competency Directory for Constructive Alignment in Digital Learning Designs and Systems*. In: Popescu E., Hao T., Hsu TC., Xie H., Temperini M., Chen W. (eds) *Emerging Technologies for Education*. SETE 2019. Lecture Notes in Computer Science, vol 11984. Springer, Cham. Retrieved from: https://link.springer.com/chapter/10.1007/978-3-030-38778-5_11
- Buchem, I., Konert, J. (2020). *Competency Mapping 3.0: Modelling, Cross-Referencing and Credentialing of Competencies with Semantic Technologies. Instruments and Implementation in the Open Virtual Mobility Project*. In: Schumacher, C. (ed.) *Proceedings of the DELFI Workshops 2020*. p. (accepted for publication). Gesellschaft für Informatik e.V, Heidelberg, Germany (2020).
- Konert, J., Buchem, I, Stoye, J. (2019). *Digital competence directory with semantic web technology linking digital proofs of competence with machine-readable competence definitions: A practical solution used in the Open Virtual Mobility project*. In Sandra Schulz (Eds.): *Proceedings of DELFI Workshops 2019 Berlin, Germany, September 16, 2019*. Retrieved from: <https://dl.gi.de/bitstream/handle/20.500.12116/27971/DELFI-Workshopband.pdf>
- Konert, J., Buchem, I., & Stoye, J. (2019). *Digitales Kompetenzverzeichnis mit Technologien des Semantic Web*. In S. Schulz (Ed.), *Proceedings of DELFI Workshops 2019* (pp. 61–70). Gesellschaft für Informatik e.V.z. <https://doi.org/10.18420/delfi2019-ws-108>
- Konert, J., Buchem, I., Lewis, L., Hamilton, G., & Riches, T. (2017). *Competency Alignment of Open Badges*. In C. M. Stracke, M. Shanks, & O. Tveiten (Eds.), *Official Proceedings of the International WLS and LINQ Conference 2017* (pp. 29–36). Kristiansand, Norway: Logos Verlag GmbH Berlin.
- Raschke, T., Konert, J. (2020). *Automatic Extraction of Semantic Competence Descriptions Using German Module Descriptions from Higher Education as an Example*. In: Schumacher, C. (ed.) *Proceedings of the DELFI Workshops 2020*. p. (accepted for publication). Gesellschaft für Informatik e.V, Heidelberg, Germany (2020).

Attachments

none