Benedikt Reuter, Gottfried Zimmermann, Tobias Ableitner and Sebastian Koch

OpenAPETutorial – A Problem-Based Learning Unit for the Personalization of Smart Home Applications

Abstract

This paper describes the creation of a problem-based learning unit. The aim is to help students to learn how to make applications more accessible through aspects of personalization. In addition, a basic understanding of smarthome interfaces will be given. For this purpose, the personalization framework OpenAPE and the smarthome interface OpenHAB will be used. A java OpenAPEClient will be created for OpenAPE. The client will then be used in a tutorial application. This tutorial application is to be integrated as a kind of cloze text in the learning unit as a practical exercise. It implements a lamp control and a personalizable user interface. Final user tests showed that the learning unit is executable as planned. Some problems and inconsistencies were already identified and eliminated during the user tests.

Keywords

personalization, learning analytics, problem-based learning

1 Introduction

A problem-based learning unit design is described in this paper, which is supposed to introduce students to concepts of personalization and the area of Smarthome. For this purpose, a learning unit with a practical exercise is to be created. In the area of personalization, the personalization framework OpenAPE will be taught. This can be used to store personalizable settings in contexts in an account and synchronize them across multiple applications and devices. For the smarthome domain, OpenHAB is used. This interface is used in smarthomes and offers to connect, control and automate different areas in smarthomes.

In the research part of the work, three topics will be addressed. First, the field of personalization needs to be looked at in more detail. In the second topic, the concept of problem-based learning is to be further explored. The third topic is about learning analytics, which is a possible extension of the learning unit.

For OpenAPE a Java client is developed, which can be used in Java and Android applications. The client should reduce the complexity of these applications since the use of the client replaces direct communication with the rest interface of OpenAPE.

In the learning unit, a final exercise will be performed. For this purpose, the so-called OpenAPETutorial application has been developed. This application implements a simple lamp control via OpenHAB and basic personalization, which can be synchronized using OpenAPE. In the application, the functionalities concerning OpenAPE and OpenHAB are outsourced to services. The methods in these services are left empty for the task, the students should complete these functionalities.

1.1 Motivation

More and more people are dependent on barrier-free applications. 15% of people live with at least a moderate disability, a large number of whom require some level of accessibility. (World Health Organization, 2011) Accessibility can also play a major role in older ages, as eyesight or motor skills may decline. In addition, the population is getting older; from 2008 to 2018, the number of people over 60 increased by 3%. (Bevölkerungspyramide: Altersstruktur Deutschlands Von 1950 - 2060)

Politicians also see the need for accessibility. Therefore, guidelines and laws are written, which make accessibility mandatory for public authorities. (EUR-Lex - 32016L2102 - EN - EUR-Lex, 2016)

For these reasons, it is important that students learn early which methods can be used to increase accessibility in applications.

A lot is also happening in the area of the smarthome. More and more devices in the home can be networked and some even enforce this networking. 3.3 million people in Germany were already using connected household appliances at the beginning of 2021. The number has increased by 5% within the last 10 years. In the area of heating, lights and electricity, about 5 million use systems in the smart home. (Statistisches Bundesamt) A central management instance in your own smarthome, which brings together various devices or technologies, is therefore becoming increasingly relevant.

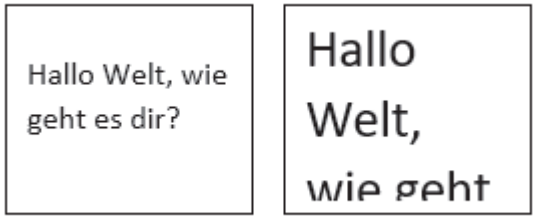

Figure 1 Example of two fields with different font-sizes

2 Personalization

is an aspect of accessibility in which the software can be adapted in such a personalized way that, in the case of various limitations, the software can still be used without any problems?

A distinction is often made between personalization and auto-personalization. Personalization is about possible changes to the interface according to the needs of the user. Auto-personalization allows this to be done automatically, i.e., without knowledge of exactly how this works in the system being used. (Vanderheiden & Jordan, 2019)

Many different ways can be used to increase accessibility in applications with personalization.

Many available tools simply try to customize different parts of the user interface. On the web, for example, this can happen by customizing CSS files. These tools are mostly installed on the client side. On the web, these tools are often included as toolbars. Predefined toolbars can be used, but also user-defined toolbars with the desired functionalities can be created. (ATbar ; Wald et al., 2011 ; Web Developer) Another possibility is to intervene in the network traffic of the browser in order to apply the corresponding transformations there. This works via a proxy. (Iaccarino et al., 2006) Outside the web, operating system extensions can be used, which can directly access the operating system's functionalities. (Vanderheiden & Jordan, 2019)

If changes are made to the user interface where the application itself is not the initiator, some problems can arise. For example, increasing the font size could cause the font to become larger than the displaying area, making it unreadable. Figure 1 shows how this could look like.

In addition to client-side tools, there are also tools that are used on the server side. The adaptations of these tools are then mostly more deeply

anchored and equally available for every user of the website or application. (OpenAPE) But even here there are possibilities to focus only on surface changes. These are easier to implement, but may have the problem described in Figure 1 again. However, since it is integrated by the providers, it can be assumed that the functionalities have already been tested. (Cloud Based Accessibility Assistive Technology Toolbar | Products | Recite Me)

In operating systems there are many functionalities to personalize a device. These include colors, contrasts, font size or size of elements. But also, ScreenReader, voice control, zoom or the reduction of animations is mostly offered.

For developers, it seems relatively difficult to write a cross-operating system application that uses the corresponding personalization of each operating system to the full extent. This is due to the different interfaces of the operating systems. A possible solution would be a uniform interface, which addresses the appropriate functionalities depending on the operating system. (Gonzalez & Reid, 2005)

3 Problem-based Learning

In the design of the learning unit, aspects from the field of problem-based learning are used.

Problem-based learning originated in the medical context and was used to teach students content through real-world problems. After it had much success in the medical context, it was transferred to other areas as well. (Jonassen & Hung, 2012)

In problem-based learning, mostly small groups of learners should tackle the real-world problem together. In many cases, no further information about the problem is provided. But often information about the fundamentals is provided. (Ellis et al., 1998)

It usually starts with some kind of analysis of the problem. The aim is to find out exactly what the problem is and what the learners already know about it.

This is followed by an attempt to enhance the missing knowledge through research in order to get closer to the solution. The problem is then discussed again in the group and analyzed once more. This process can then be repeated until the solution is found. (Jonassen & Hung, 2012) This is only one possible process. Depending on the requirements and use case, the process can also be different.

(Ellis et al., 1998) differs into three different approaches within problem-based learning:

- problem-based approach
- guided problem-based learning
- full problem-based learning

While in the first approach the necessary information regarding the problem is already provided, in the second approach the information is only partially described. In the third approach the independence of the learners is highlighted. Here almost no information is given, this must be searched for in the process of finding a solution. Moreover, the way in which the problems are presented may vary. The problems to be addressed may be fully known and described by the teacher. So, the learners only have to look for a solution to the problem. On the other hand, the problems can also be completely unknown. In this case they have to be identified first. In between there is a mixture with partially described problems. (Ellis et al., 1998)

In the context of computer science and programming, problem-based approaches can be particularly effective. In this field, the competence to solve problems is needed every day and is therefore very important. This competence is also strengthened by problem-based learning. (Learning Programming: Enhancing Quality Through Problem-Based Learning, 2003)

Problem-based learning can increase the complexity of the learning process, as students need to have skills in problem solving and possibly also in communication and teamwork, in addition to technical skills. This can be a potential weakness. Another weakness is the teaching of fundamentals via the problem-based approach. (Learning Programming: Enhancing Quality Through Problem-Based Learning, 2003)

While problem-based learning engages learners deeply in the subject matter and thus develops a concrete understanding of that problem, it can create a gap in the learners' broader knowledge. The students are very focused on this one-use case, a transfer to another use case in the future could still bring problems despite a good understanding during the learning unit. To counteract this problem, learners must be offered sufficient information, which can then be accessed as needed. (Kolmos & de Graaff, 2003)

4 Learning Analytics

(Clow, 2012) describes learning analytics as a cycle consisting of four steps. The starting point in the cycle is the learners. During learning, the second

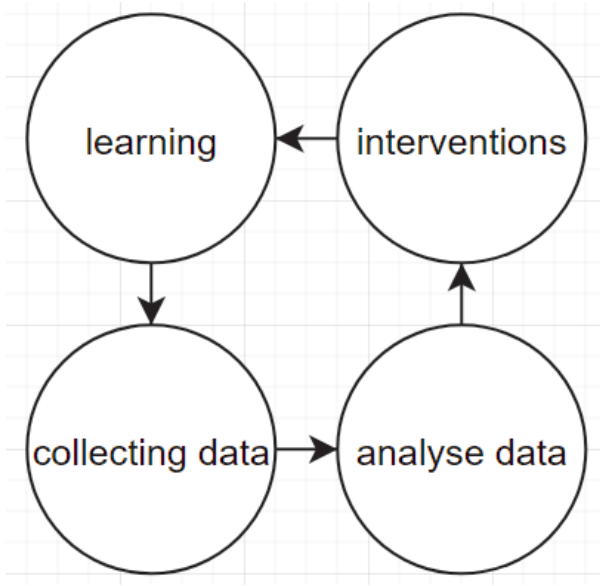

Figure 2 Simplified representation of the learning analytics cycle

step is to collect data. This can be either from learners or about learners. This data is then processed in the third step, for example through analysis. After the data has been processed, the results of this processing can be used to influence the learners in the final step. This step completes the cycle. The Figure 2 visualizes this approach.

In order to implement learning analytics, various methods and tools are required. First of all, it must be clarified which data needs to be collected. This can be done on different levels. For example, in programming exercises, the number of compilations or the exceptions thrown during an execution of the application can be collected. If this data is not sufficient, more detailed data can be collected. Here you might log every keystroke the learner makes. Additionally, data can be collected on a social level by using discussion forums for the tasks. There, the learners can exchange information about the problems in the tasks and the teachers gain information about which problems occur during the execution. (Ihantola et al., 2015)

Since the data often contains person-related data, ethical aspects and privacy should not be neglected. However, according to (Ihantola et al., 2015), this is not addressed enough in many studies.

To integrate learning analytics various ways can be chosen. For example, plugins in the development environment can help to collect data and provide a direct forum for communication inside of the development environment. (Olivares & Hundhausen, 2017)

Furthermore, interfaces can be used, which can receive data from different tools and applications. The data can then be analyzed on an appropriate platform. (Ihantola et al., 2015)

5 Introduction OpenAPE

OpenAPE is an open-source framework that offers a way for developers to implement personalization in their own application.

The biggest difference between OpenAPE and many other tools for personalization is that OpenAPE must be integrated into the architecture of the application. Here the integration of a plugin or a toolbar is not sufficient. The personalization functions must be implemented in the source code and must be retrievable from OpenAPE or stored in OpenAPE. However, once it is integrated into the architecture, properties can be synchronized across multiple devices and multiple applications that have implemented OpenAPE. This will then work without any further installations on the user side.

In OpenAPE everyone can create an account, in which he can store so-called contexts. These contexts can contain different data. In OpenAPE, there are the following four context types:

- User-Contexts: can be used to assign specific properties to a user.
- Task-Contexts: intended for various tasks, e.g. properties for surfing on the Internet or writing e-mails.
- Equipment-Contexts: represents the properties for a specific device, e.g. desktop PC or a laptop.
- Environment-Contexts: for specific environments, e.g. a daytime or nighttime use.

In this work only, the user contexts are relevant, therefore the other contexts are not described in more detail. If a context is mentioned, the user context is always meant here.

Each user can have several user contexts, where he can store different preferences. For example, font size, colors, language or other information about applications can be stored here. This stored information can then be accessed by any application after logging in via the Rest interface.

The Listing 1 shows such a context. Line 2 contains the unique name or ID of the context. There can be multiple of these objects behind line 2, they are identified by the ID. Line 3-6 shows the stored preferences of the context. In this case a font size of 60 was stored in the context with the name "default".

```
1 {
2   "default": {
3     "preferences":
4       {
5         "..fontSize": "60"
6       }
7   }
8 }
```

Listing 1 Simple OpenAPE user context

The entries in the preferences are simple key-value pairs, where the value can be a string, boolean or a number. To make the preferences consistent across multiple applications, the OpenAPE Registry Service should be used. (OpenAPE Registry Service - HdM) This is available at "https://terms.gpii.eu" and already contains several publicly available entries. Here an id, a description of the data type to be selected and other information is stored. Additionally, a json schema can be defined, which describes the format of the value. All developers can access this database, so that the data is saved in a uniform way. However, this is not controlled, so there is no control whether the data is really stored according to the given specification.

5.1 OpenAPE Focus Group

In a focus group with students from the Computer Science and Media bachelor's degree course at the Hochschule der Medien, the goal was to gather information about what the students think of the idea and implementation of OpenAPE. In addition, the aim was to find out how such a framework could be integrated into a learning unit.

As part of a lecture in which students are required to program an application with OpenAPE connectivity, 9 students participated in the focus group. The focus group took place online, which may have led to reduced activity in the discussions.

Only one student had already used the interface, all others had only the introduction to the topic. Nevertheless, some points could be discussed.

Especially important for the students was a type safety for the preferences. The types of the entries can be described by a second service, but a control mechanism, which then also checks whether the correct values were entered, does not exist. Another point was lists. In the current implementation it is not possible to store lists in OpenAPE.

Furthermore, it was mentioned by the students that it would be helpful to have direct examples of the use of OpenAPE in order to learn OpenAPE. For example, a "Hello World!" application that implements the benefits of OpenAPE could help to understand how exactly OpenAPE should be used.

6 Introduction OpenHAB

The open-source project OpenHAB stands for open Home Automation Bus. It is used in Smarthomes, where it can do different jobs. Many devices, systems or technologies can be connected to the OpenHAB system and OpenHAB can become the centre of a smart home. Moreover, with OpenHAB automation can be achieved in the Smarthome area. A cloud connection is not necessary with OpenHAB, but if desired, it can be connected to common cloud-based systems. You can host OpenHAB on your own local system or via a cloud connector directly in the cloud. (OpenHAB Documentation) Several keywords are relevant for OpenHAB: Bindings, Things, Items, Rules.

To connect OpenHAB with other devices, you can install Bindings in your OpenHAB system. These Bindings create a connection to the devices and then create so-called Things. These Things are the corresponding representations of the devices. Things can then be used to generate items. Items represent functionalities of the things. They have a status which can be changed if necessary. For example, there is the item type "Switch", which can represent an on/off switch of a lamp. (OpenHAB Documentation)

For automation, rules can be created that can execute different commands in different situations. To be able to use OpenHAB from everywhere, a rest interface is offered. (OpenHAB Documentation)

7 Java OpenAPEClient

In addition to the learning unit, a Java client for OpenAPE was also implemented. This client will also be discussed in the learning unit and used in the tutorial application of the learning unit. The aim of this client is that

```
1 OpenAPEClient openAPEClient = new OpenAPEClient();
2 openAPEClient.login("username", "password");
3 UserContextMapDTO contextMap = new UserContextMapDTO();
4 UserContextDTO context = contextMap.getOrCreateContext("default");
5 context.addOrUpdatePreference("http://terms.gpii.eu/common_fontSize", "60");
6 context.addOrUpdatePreference("http://terms.gpii.eu/common_language", "de");
7 String id = openAPEClient.createUserContext(contextMap);
```

Listing 2 OpenAPE Client - codesample create context

it can be used in any Java and Android application in order to reduce the complexity of these applications.

In the client, an http client is implemented, which makes the appropriate calls to the Rest interface. The client then offers the corresponding calls as methods.

So far, only the methods for user contexts are available in the client. However, it is built in such a way that the other context types can be added without a major modification of its structure.

The OpenAPEClient is close to the Rest interface. Listing 2 shows how a context containing two preferences can be created using the client. Lines 1 and 2 initialize the client and log in the user. Lines 3-6 show how to create the structure of the context and insert the appropriate data. In line 7 the context is then created. What happens with the returned ID must be decided depending on the use case.

8 Learning Unit

The goal of the learning unit "Universal Design & Personalization for Smarthomes - Implementation" is to get a technical understanding of how to implement accessibility through personalization already during the development of applications. The personalization framework OpenAPE will be used for this purpose.

In addition, basic knowledge in the area of Smarthomes will be taught; this will be implemented with the help of the Smarthome framework OpenHAB. After the learning unit has been completed, the students should be able to implement and use both interfaces in applications. Especially the personalization with OpenAPE should be kept in mind as an option to improve accessibility in applications.

The learning unit is aimed at students who already have experience in the field of programming. In addition, the students must have already com-

pleted another course unit before this learning unit, which is the learning unit "Universal Design & Personalization for Smarthomes - Concepts".

The course is intended to be carried out by each student alone, so no group work is foreseen. The learning unit is also well suited as an online course since the students should work alone either way. Nevertheless, a lecturer should be available at any time for questions. However, this could also be solved via a forum in which the students can exchange information with each other.

In the subdivision of (Ellis et al., 1998), the learning unit falls between the first and second approach. Students get most of the information they need through the learning unit. However, with the help of various documentations about interfaces and tools, the students should read more in depth and thus gain more information. As the problem of the learning unit, it is described to the students how people with a visual impairment operate applications and what problems can arise. More concretely, the programmed application should be operated by different people, therefore it should be made usable in a personalize way with OpenAPE.

Through the information about the interfaces in the learning unit, it attempts to counteract the problem that problem-based learning is less suited to fundamentals. By providing additional links to documentation and further information, students will also have the opportunity to know where to find this information at any time in the future. (Learning Programming: Enhancing Quality Through Problem-Based Learning, 2003)

The verification of the tasks can be designed dynamically. If no verification is provided, the students can verify themselves by checking the functionalities they implemented. Otherwise, the tasks could be checked by handing in the programmed source code.

8.1 Content

The content of the learning unit can be divided into two parts.

Part one covers the theory of the unit. The learning unit is introduced by trying to give the learners an understanding of why personalization can be useful in an application. For this purpose, four images of a seat selection in a movie theater are shown. Reserved seats are marked in red and free seats in green. On three of the images a filter is applied, where the seat selection is shown with different visual impairments. The learner sees the selection as a person with a visual impairment would see it. On two of the images no differences in color can be seen through these filters, on another one the colors are no longer the original colors. The fourth image shows the

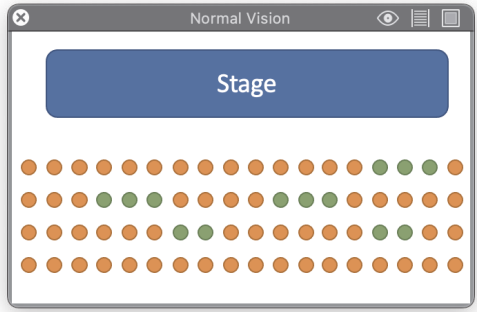

Figure 3 Seat selection without any filter

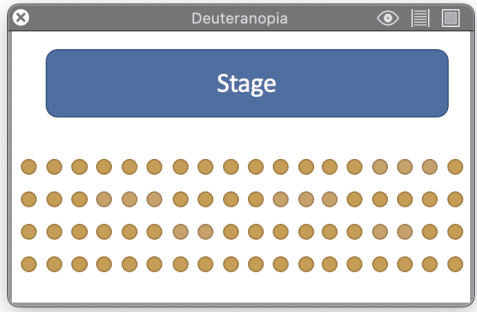

Figure 4 Seat selection with red-green filter

selection without the filter. The Figure 3 and Figure 4 show examples of the used use case.

Chapter two describes OpenAPE. The basic information about it is already explained in the preceding learning unit. The possibilities of the rest interface are described, and the individual calls are explained in more detail. Afterwards the usage of the programmed Java OpenAPEClient is explained. Here the available methods and Exceptions are presented. In addition, an illustrative use is demonstrated in a sequence diagram and parts of the data structure used within the client are shown in a UML diagram. In the third chapter OpenHAB is introduced. In addition to background information, the required calls of the rest interface are described in more

detail. In the fourth and fifth chapter, the OpenAPETutorial application and the assignment for this application are presented. Necessary information about the implementation of this application is also given. More about the application and the task will be described in the next chapter, chapter 8.2. The second part of the learning unit is the practical exercise. The exercise is a fully programmed application with parts of the source code removed. The removed content relates to the implementation of the OpenHAB and OpenAPE interfaces. So, it becomes a kind of cloze. As already mentioned, the application is described in the following chapter 8.2.

The description of the functionalities of both interfaces are all technical. For example, the rest calls describe all the technically required data. This leads to the fact that it might be difficult to imagine the execution. Therefore, there is a short video in the learning unit for each rest and method call, which has a length of about 30-60 seconds. In these videos the corresponding call is executed using an example and in addition it is explained. These videos have the additional advantage that the students can watch individual videos about the problematic functionalities at later stages if they have problems with these certain functionalities. In addition to the videos, a file is provided that can be imported into the rest interface tool insomnia. It can be used to perform the rest calls described in the videos themselves. The focus group from chapter 5.1 showed that students find practical examples useful for learning the interface. Since a complete example application would simplify the tasks in the OpenAPETutorial application too much, only the videos and the exported file are given here as examples.

8.2 OpenAPETutorial Application

In the learning unit, basic knowledge about the implementation of the OpenHAB and OpenAPE interface will be acquired. This basic knowledge will be tested and deepened in the OpenAPETutorial application.

For this purpose, a simple Smarthome application was programmed, which shows a simple lamp control on the first page. The items retrieved via OpenHAB are displayed. In addition to the status of the item, there is also the option to change the status. When switching to the settings page, possible settings for personalizing this application are offered. These are the following settings:

- language
- font size
- font color
- background color

- font color of controls
- background color of controls

These settings are integrated in the application and can be changed, depending on the needs of the user. Through this page, the user can get to a login page for OpenAPE. After a login, the settings stored in a user's context are retrieved and applied in the application. If no settings have been saved there yet, nothing will happen. In addition, after the successful login, all changes to the settings in the application are also saved in the context in OpenAPE. If necessary, a context is created for this purpose.

In order to integrate the application into a learning unit, the functionalities of OpenHAB and OpenAPE were outsourced to services, which are called from the rest of the source code. For the assignment, the contents of the methods implemented in these services are removed. What remains are empty methods, which the students then must fill.

The following empty methods remain for the OpenAPEService:

- constructor
- login()
- isLoggedIn()
- saveSettingsInOpenAPE()
- loadSettingsFromOpenAPE()

In addition, a method remains which returns an object of the global settings. This object must be used if settings are to be read from the application or applied.

The following empty methods remain in the OpenHABService:

- constructor
- getItems()
- sendCommand()

For OpenHAB one more class remains: OpenHABRest. Here an interface is implemented, which is needed by retrofit. It contains wrapper methods with information regarding the rest calls to be executed. These must also be filled in by the students. The methods that are included there are the same.

The possibility to personalize the application has already been implemented and is not relevant for the learning unit. If the students have to implement these aspects additionally when doing the learning unit, another big topic would need to be opened up. By keeping this content out, no knowledge about the framework in which the application is written needs to be taught.

9 Usertests

In order to check whether the designed learning unit can be carried out as planned, final qualitative user tests are executed.

9.1 Preperation

For the user tests, students from the target group, i.e., with experience in programming, are recruited. The problem here is that the students who are recruited have not done learning unit 1, so they do not have the prior knowledge. In addition, the user tests should not last longer than one hour.

For these reasons, the learning unit is only carried out in sections. Parts of the learning unit will only be discussed orally. For some content, assistance will be provided so that the content can be processed more quickly. In addition, not everything will be completely programmed in the programming exercise; comments or pseudocode are usually sufficient. This procedure is intended to validate the learning unit in terms of content and still not exceed the time frame of the user tests.

In the individual user tests, different priorities are set in each case, so that each area is carried out as planned at least once. The focus is set depending on previous user tests, depending on what worked well or poorly there.

During the user tests, the users were asked to think aloud and try to explain how they understood the subject at hand. This leads to the identification of whether the described content conveys what it is supposed to.

9.2 Evaluation

A total of 5 user tests were carried out. Many weaknesses that were noticed by the users could be fixed directly, in the following some points for such problems are described.

In general, many small logical or syntactical errors have been fixed. But also the structure of how the information is provided was adjusted according to the findings of the users.

Some subjects were unclear about the data structure of the OpenAPE client. As a result, corresponding UML diagrams were added for the most important classes in the data structure.

In the chapter about OpenAPE, a sequence diagram was shown, which is intended to reflect a possible flow of the OpenAPEClient in a real application. Here the steps of the initialization of the OpenAPEClient were skipped

and the return value void was not described. This led to students missing the initialization step in the programming exercise and the void return value causing confusion.

In one user test, the test person described that he first had to orient himself in the task and especially in the OpenAPETutorial application. He found this relatively difficult. To help students orient themselves better, the tasks could be broken down and described in more detail. In addition, code comments could be added, which describe again what must be done at the respective locations in the source code. This is probably a problem in time-limited user tests. In real-life situations, however, students should deal with the topics more intensively and not be given exact instructions for the tasks. This should lead to students learning the knowledge better as it is not a simple fill in the blanks. This issue should nevertheless still be noted in possible pilot tests.

The videos were rated as particularly positive. The demonstration of the functionalities with real data made it possible to understand the contents very well. Three test participants had little to no experience with the use of rest interfaces, but after a brief introduction to the basics and watching the videos, they were able to complete the learning unit without any problems. The videos were rated particularly well by these test participants, as they were able to help very well in understanding the calls to the rest interface.

In the tasks regarding OpenHAB, retrofit (Retrofit) is used as http client. For this topic, little information is described in the learning unit. The learners must read the documentation of the tool to understand the functionalities. After a user had problems with it, the section of the explanation about retrofit was filled with additional information. In a subsequent user test, there were still unclear points. These were not corrected in the following, because here again the assumption is that in a real situation there is enough time to read the documentation. This assumption is strengthened by the fact that the users could understand the functionalities well afterwards.

10 Conclusion and future work

The OpenAPEClient implements only basic functionalities so far. The further development of this client can go into many areas. Probably the most important further development would be the support of all context types, the client is designed to implement this extension. Moreover, type safety or lists in the preferences could be implemented via the client. These are the functionalities that stood out through the focus group. The functionalities

should first be implemented in the OpenAPE interface itself, so that everyone can access them via the rest interface.

The execution of the learning unit could be validated by the user tests. The tests showed that the learning unit is understandable and the tasks can be performed with the knowledge taught. In addition, some weaknesses could be identified, most of them could also be fixed directly. Some issues were intentionally not fixed, as they may have occurred only under the circumstances of the user test.

Through the focus group, it became clear that students liked having hands-on examples like a Hello World application to learn from. Although a Hello World application was not provided, the explanatory videos of the individual calls to the interfaces provided practical examples with sample data, which were very popular during the user tests.

Despite the successful user tests, a pilot test under real conditions should be done in the future. In the conducted user tests different aspects of the learning unit were reviewed, but due to the way it was conducted, e.g. the time needed for the theory part and the tasks could not be checked. In addition, this pilot phase could further monitor the vulnerabilities encountered in the user tests and identify additional issues. Especially the problems that were not fixed due to the circumstances of the test should be monitored further.

Regarding learning analytics for the application, it must be decided which data can be relevant for a possible analysis. From this, it can then be decided at which level the data must be collected. In order to then embed this in the learning unit, it may need to be integrated into a development environment that has been designed for this purpose. Another possibility would be an interface which collects the data.

References

ATbar. *ATbar*. https://www.atbar.org/ (last accessed: 23.02.2022).

Bevölkerungspyramide: Altersstruktur Deutschlands von 1950 - 2060. https://service.destatis.de/bevoelkerungspyramide/ (last accessed: 23.02.2022).

Cloud based Accessibility Assistive Technology Toolbar | Products | Recite Me. https://reciteme.com/product/assistive-toolbar (last accessed: 23.02.2022).

Clow, D. (2012). The learning analytics cycle. In S. Dawson & C. Haythornthwaite (Eds.), *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK '12* (p. 134). ACM Press. https://doi.org/10.1145/2330601.2330636

Ellis, A., Carswell, L., Bernat, A., Deveaux, D., Frison, P., Meisalo, V., Meyer, J., Nulden, U., Rugelj, J., & Tarhio, J. (1998). Resources, tools, and techniques for problem based learning in computing. *ACM SIGCUE Outlook*, *26*(4), 41–56. https://doi.org/10.1145/309808.309825

EUR-Lex - 32016L2102 - EN - EUR-Lex. (December 26, 2016). https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32016L2102 (last accessed: 23.02.2022).

Gonzalez, A., & Reid, L. G. (2005). Platform-independent accessibility API. In S. Harper, Y. Yesilada, & C. Goble (Eds.), *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A) - W4A '05* (p. 63). ACM Press. https://doi.org/10.1145/1061811.1061824

Iaccarino, G., Malandrino, D., & Scarano, V. (2006). Personalizable edge services for web accessibility. In S. Harper, Y. Yesilada, & C. Goble (Eds.), *Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A) Building the mobile web: rediscovering accessibility? - W4A* (p. 23). ACM Press. https://doi.org/10.1145/1133219.1133224

Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M. Á., Sheard, J., Skupas, B., Spacco, J., Szabo, C., & Toll, D. (2015). Educational Data Mining and Learning Analytics in Programming. In N. Ragonis & P. Kinnunen (Eds.), *Proceedings of the 2015 ITiCSE on Working Group Reports* (pp. 41–63). ACM. https://doi.org/10.1145/2858796.2858798

Jonassen, D. H., & Hung, W. (2012). Problem-Based Learning. In N. M. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 2687–2690). Springer US. https://doi.org/10.1007/978-1-4419-1428-6_210

Kolmos, A., & de Graaff, E. (2003). Characteristics of Problem-Based Learning. *International Journal of Mechanical Engineering Education*, *19*(5), 657–662.

Learning programming: Enhancing quality through problem-based learning. (2003). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.516.8969&rep=rep1&type=pdf

Olivares, D. M., & Hundhausen, C. D. (2017). Supporting learning analytics in computing education. In A. Wise, P. H. Winne, G. Lynch, X. Ochoa, I. Molenaar, S. Dawson, & M. Hatala (Eds.), *Proceedings of the Seventh International Learning Analytics & Knowledge Conference* (pp. 584–585). ACM. https://doi.org/10.1145/3027385.3029472

OpenAPE. https://openape.gpii.eu/ (last accessed: 05.09.2021).

OpenAPE Registry Service - HdM. https://terms.gpii.eu/web/login/ (last accessed: 24.10.2021).

OpenHAB Documentation. https://www.openhab.org/docs/ (last accessed: 23.02.2022).

Retrofit. https://square.github.io/retrofit/ (last accessed: 23.02.2022).

Statistisches Bundesamt. *3,3 Millionen Menschen nutzten 2020 smarte Haushaltsgeräte*. https://www.destatis.de/DE/Presse/Pressemitteilungen/Zahl-der-Woche/2021/PD21_27_p002.html (last accessed: 23.02.2022).

Vanderheiden, G., & Jordan, J. B. (2019). Personalization and Layering to Simplify Computer Accessibility. In J. P. Bigham, S. Azenkot, & S. K. Kane (Eds.), *The 21st International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 685–687). ACM. https://doi.org/10.1145/3308561.3354601

Wald, M., Draffan, E. A., Skuse, S., Newman, R., & Phethean, C. (2011). Southampton accessibility tools. In L. Ferres, M. Vigo, & J. Abascal (Eds.), *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility - W4A '11* (p. 1). ACM Press. https://doi.org/10.1145/1969289.1969319

Web Developer. https://chrispederick.com/work/web-developer/ (last accessed: 23.02.2022).

World Health Organization. (2011). *World report on disability*. WHO. https://www.who.int/disabilities/world_report/2011/report.pdf

Authors

Benedikt Reuter
Hochschule der Medien
Nobelstraße 10
70469 Stuttgart
https://www.hdm-stuttgart.de/remex
reuter@hdm-stuttgart.de
benedikt.reuter@posteo.de

Prof. Dr. Gottfried Zimmermann
Hochschule der Medien
Nobelstraße 10, 70469 Stuttgart
https://www.hdm-stuttgart.de/remex
zimmermanng@hdm-stuttgart.de

Tobias Ableitner
Hochschule der Medien
Nobelstraße 10
70469 Stuttgart
https://www.hdm-stuttgart.de/remex
ableitner@hdm-stuttgart.de

Sebastian Koch
Hochschule der Medien
Nobelstraße 10
70469 Stuttgart
https://www.hdm-stuttgart.de/remex
kochs@hdm-stuttgart.de