Supporting Developers in Creating Web Apps for Education via an App Development Framework

Juan Carlos Farah[1], Sandy Ingram[2], Denis Gillet[1]
[1]School of Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland,
[2]School of Engineering and Architecture of Fribourg, University of Applied Sciences (HES-SO), Switzerland.

Abstract

As more educational activities are conducted online, the need for interactive applications (apps) that can effectively support educators in their practice is increasing. These apps are often created by web developers or by researchers, educators, and even students with programming experience. While a large body of work has focused on incorporating these apps into educational contexts, fewer studies have focused on their development. In this paper, we first present the design and implementation of an app development framework aimed at supporting developers in creating apps for education. We then report the results of a study comprising interviews with 12 developers who used the framework. Our findings highlight that while the creation of web apps for education can be facilitated by a purely software-based app development framework, effectively exploiting such a framework requires domain knowledge that could be acquired through in-depth documentation, tutorials, and collaboration between developers and educators.

Keywords: *app development framework; apps for education; online learning; developer support; interactive apps; digital education.*

1. Introduction

As more educational activities move to online learning scenarios, the need for adequate tools to support these scenarios is becoming more pertinent. However, developing these tools requires programming skills, which can be a limitation for teachers and researchers in education looking to incorporate custom tools into their practice. In this paper, we present the design of an open-source development framework aimed at supporting the creation of web applications (apps) for education. This framework was used by 20 developers to create over 35 apps that were deployed in an online learning environment. To evaluate our framework, we conducted a qualitative study comprising semi-structured interviews with 12 developers who used the framework. Results from these interviews were analyzed using thematic analysis, shedding light on the (i) technical, (ii) educational, and (iii) open-source aspects of our framework, as well as on how to make app development for education more accessible. Our results point toward the need for collaboration between developers, researchers in education, and educators, which could be facilitated by our framework.

2. Background and Related Work

Educational design approaches, frameworks, and software proposed in the literature often focus on the production of learning content (Puggioni et al., 2020) or the design and development of tools supporting learning activities (Hohenwarter, 2002). Furthermore, frameworks supporting software development for education generally target specific platforms or subject matter. On the one hand, some learning management systems (e.g., Moodle) support the development and integration of custom plugins (Moore & Churchward, 2010). These plugins, however, are tightly coupled with the content hosted on the respective platform and cannot be deployed externally without custom software, such as an LTI adapter (IMS Global Learning Consortium, 2019). On the other hand, some online simulation providers, such as the PhET Interactive Simulations project (Perkins et al., 2006), provide access to their development frameworks but are specifically tailored for certain subjects.

Our goal is to provide an open-source, platform-agnostic, software development framework that can be harnessed to create apps aimed at supporting online learning across academic disciplines. This approach is best aligned with the H5P project (h5p.org), which supports the creation of reusable interactive HTML content. Nevertheless, while H5P is widely used in education (Amali et al., 2019), it targets publishing platforms in general and is designed primarily for content creators, not developers. Our framework exclusively targets educational contexts and is designed for developers working on creating web apps for such contexts. Furthermore—to the best of our knowledge—no study has focused on gathering feedback from developers on their experience creating apps for education. In this paper, we address these gaps by presenting the design and evaluation of our app development framework.

3. Design

Our app development framework is designed following a headless and decoupled architecture similar to that of the popular Gatsby framework (Gatsby, 2022). Gatsby is a JavaScript framework for creating and deploying websites. It allows developers to use a command-line interface (CLI) to create websites based on templates, some of which are featured in a public repository. Developers can then build on those templates and deploy their websites to their own infrastructure or to one provided by Gatsby. Our design adopts and adapts this strategy to online education platforms. In this section, we present the five core components of our framework: (i) context, (ii) application programming interface (API), (iii) templates, (iv) CLI, and (v) repository.

3.1. Context

Our app development framework is primarily designed for apps that are incorporated into an online learning context, which we refer to as running in *contextual mode*. Specifically, apps are loaded by calling the URL where the app is hosted, either in an iframe (if embedded directly in a learning activity) or as a standalone web page (if opened as a separate link). To inform an app of its context, the URL includes a query string that contains its unique ID within the learning platform, as well as the IDs of the activity that it is a part of (i.e., its context) and of the user that is interacting with it. The query string also informs the app whether it is running within the platform's teacher or student interface and provides the endpoint that the app needs to call to access the platform's API. Apps can then render role-specific views or components, such as a settings panel that is only available to teachers. If this context is missing, apps can fall back to running in *standalone mode* (i.e., without support from an online learning platform). Standalone mode is particularly useful for apps that do not require user-generated data (e.g., simulations) or to perform demos.

3.2. Application Programming Interface (API)

Apps running in contextual mode are aimed at communicating with online learning platforms and therefore need access to data hosted by these platforms, such as learning resources and activity traces. Access to this data is often made possible via APIs. While each platform has its own API, our goal was to identify the core data required to support educational apps and design our API specifications accordingly. To do this, we created a public API that exposed endpoints for the app to (i) get information about the user that is currently logged in, (ii) get a list of learners that have interacted with the app or with the lesson in which the app is embedded, (iii) read and update its configuration, which is visible to all learners (e.g., an app providing an online programming environment could be configured for Python or JavaScript), (iv) create, read, update, and delete its own resources, which include user-generated learning outputs, and (v) create, read, and delete activity traces generated by users while interacting with the app. These endpoints are loosely based on the—now deprecated—OpenSocial standard, which defined a social API specification that was adopted by some open education platforms (Gillet et al., 2013). Once our API was set in place, apps could exploit it to get access to the content they required to run.

3.3. Templates

While it is possible to develop an app for our framework from scratch, templates allow developers to maximize code reusability and focus on app- and use-case-specific code. Templates are code repositories that serve as a starting point for the development of an app. While we provide default templates, developers can create custom templates, either from scratch or by building on our default templates. This reduces the amount of boilerplate that developers need to write in order to create apps. That is, templates allow for the basic structure for a particular set of apps to be only written once. The structures provided by templates can facilitate many development tasks, including enforcing code style, setting up development and testing frameworks, providing design components, and abstracting the API through ready-made functions that pre-establish the link between the app and the platform(s) it will be hosted on. Furthermore, templates can be generated in various frontend development languages (e.g., JavaScript, TypeScript, Elm) and frameworks (e.g., React, Angular, Vue) to attract a wider community of developers. Our default templates are written in JavaScript and cover both the React and Angular frameworks. These templates encapsulate the core structure of an app targeted at our online learning platform, including—on top of the aforementioned development tasks—privacy and role-aware data access and visualization modes, frontend state management, an open-source license, and compatibility with an offline desktop environment.

3.4. Command Line Interface (CLI)

To make it straightforward to get started with a template, we created a CLI that guides developers through the process of bootstrapping an app. The CLI's `new` command prompts developers for information concerning the template that they want to use, the name for their app, and optional IDs required for access to our API and deployment to our infrastructure. The CLI can also be used to test an app locally, package it as a compressed file, deploy it to our infrastructure, and publish a new version. These commands harness lower-level utilities (e.g., `git`, `npm`, `aws`) and custom shell scripts, abstracting this logic from the developer.

3.5. Open Educational App Repository

A final component of our design is the app repository. While the code for the apps could be hosted in any source code repository (e.g., GitHub, GitLab), these repositories are not specifically aimed at education. To allow educators to more easily discover and use apps built with our framework, we created a repository of open educational resources that could host these apps and provide information about them in a way that is accessible to non-technical users. Our design followed the approach used by other app repositories such as Go-Lab (de Jong et al., 2014), whereby educators can quickly search for, filter, and test apps that are relevant to their practice.

4. Applications

Over 35 apps have been created and published in our open repository. These apps range from virtual labs targeting physics education, to learning analytics visualizations for both teachers and students, to interactive chatbot interfaces. Figure 1 illustrates four examples of apps that were created using our app development framework. *Code* (top left) allows students to write and execute Python code in the browser (Farah et al., 2020). *Sticky Notes* (top right) enables collaborative design thinking and brainstorming. *Global Model* (bottom right) helps students visualize the effects of global warming. Finally, *Light Pollution Simulator* (bottom left) aims at raising awareness of the effects of light pollution on the night sky (Gomes et al., 2019).

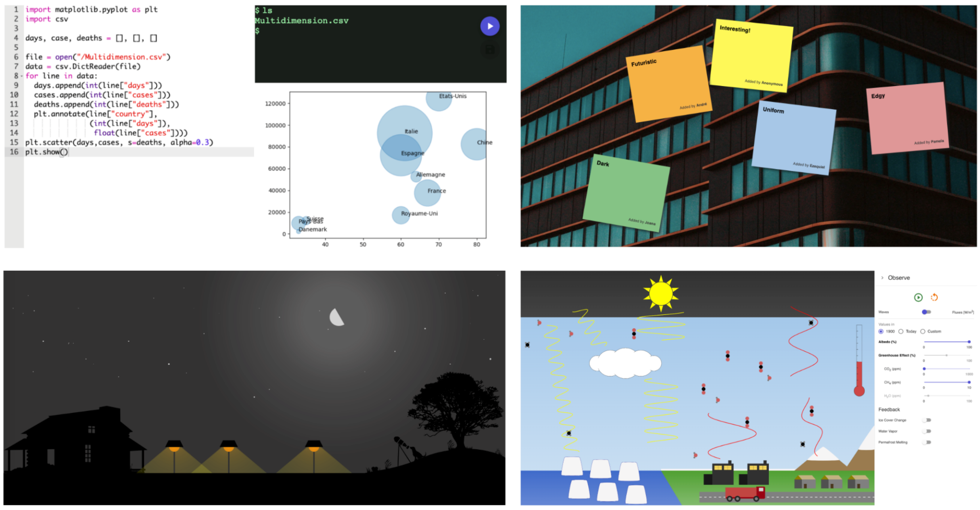

Figure 1. Four example apps (clockwise from top left): Code *for Python programming,* Sticky Notes *for design thinking,* Global Model *for climate change awareness, and* Light Pollution Simulator *to visualize light pollution.*

5. Evaluation

Our evaluation addressed the following research question: *How did our app development framework facilitate the creation of apps for education?* Specifically, we focus on three aspects: (i) *technical*, (ii) *educational*, and (iii) *open-source*.

5.1. Methodology

Our qualitative analysis consisted of semi-structured interviews, given their applicability to small-scale research (Drever, 1995). Interviews lasted approximately 30 minutes, took place using an online video conferencing tool, and were loosely guided by 15 questions covering four topics: (i) developer background, (ii) experience with the app development framework, (iii) educational context, and (iv) importance of open-source code. Following the total population purposive sampling method, we contacted all developers who were identified as having worked with our app development framework. A total of 13 developers agreed to participate and were interviewed in January and February 2022. One interview was excluded from our analysis due to technical issues with the transcript. We analyzed our qualitative data using thematic analysis, performing line-by-line coding on the interview transcripts to identify emergent themes with respect to the three aspects of our study.

5.2. Results

Concerning the technical aspect, all 12 developers were overall positive about the support received by the framework, although five developers expressed that they found it challenging to get started due to their lack of experience either with JavaScript (two developers) or React (three developers). Five developers noted that the boilerplate provided by the default template was helpful, while nine developers alluded to the consistent and adequate structure of the code. Eight developers highlighted the need for more technical support, with five developers suggesting the creation of video tutorials. These tutorials were suggested as a way to explain what was included in the template (two developers) and to help developers create a sample app (two developers). Four developers also noted the need for more templates to support other technology stacks (one developer), keep the code up-to-date with current standards (three developers), and make parts of the code optional (one developer).

Concerning the educational aspect, all developers were overall positive about the framework's impact on helping create apps for education, with six developers noting that the framework's support for learning analytics was an important feature. Nevertheless, five developers reported that understanding the educational context and nomenclature of the framework was challenging. To address this challenge, four developers highlighted the importance of collaboration between developers and educators. Two developers specifically suggested incorporating requirements elicitation and validation processes into the framework. The importance of collaboration is best illustrated by a quote from one of the developers, who mentioned that "*one of the challenges is that very often it's not really about the code and how skilled the coders are… it's more [about] the teachers and the teachers who know the [subject matter] and who know how the students will receive [the app] and what they will find challenging and what they will find interesting… I'm not a teacher, so I wouldn't have that knowledge. I don't know the [subject matter]… maybe I could learn it, but*

I'm not going to learn it in a way that someone who's been teaching it for 5-10 years knows it, and also [knows the] students".

Finally, concerning the open-source aspect, all 12 developers were positive about the fact that the framework encouraged open-source. Reasons put forth for the importance of open-source included (i) not reinventing the wheel (two developers), (ii) making knowledge and examples accessible (seven developers), and (iii) building a community (three developers).

6. Discussion

The results of our evaluation show that on the one hand, developers found our framework useful in terms of the technical advantages it provided through convenient boilerplate, the ease of use of the CLI, and the consistent structure of the code. On the other hand, the framework failed to fully support developers in understanding the educational aspects involved in the development of apps aimed at learning contexts. The need to better support and encourage collaboration between the different stakeholders in education is aligned with work by Tavares et al. (2020), who proposed a participatory design process to bring together researchers in education and end-users of educational mobile apps. Indeed, our findings suggest that collaboration should also be established between software developers, researchers in education, and educators. This collaboration could be facilitated by the development framework, which could include educational aspects in its documentation and tutorials, incorporate an automated validation process, provide a library of reusable components that are strongly linked to pedagogical interfaces, and follow nomenclature defined through a participatory design process involving both developers and educators.

7. Conclusion

In this paper, we presented the design and evaluation of an app development framework aimed at supporting developers in creating apps for educational contexts. Our results suggest that developers would benefit from support from experts in education, which could be facilitated by the app development framework. Nevertheless, our evaluation has some limitations worth noting. First, while the sample size is appropriate for a qualitative study using semi-structured interviews, it would be useful to complement these findings with a larger sample size, along with a quantitative analysis of the actual usage of the framework. Second, our app development framework allows apps to be deployed in standalone or contextual mode. However, contextual mode currently only supports the Graasp online learning platform (Gillet et al., 2022). To increase portability, contextual mode should be compatible with multiple platforms. In future work, we aim to address these limitations and incorporate the feedback received into the next iteration of our framework.

References

Amali, L. N., Kadir, N. T., & Latief, M. (2019). Development of E-learning Content with H5P and iSpring Features. *Journal of Physics: Conference Series*, 1387(1), 012019. doi: 10.1088/1742-6596/1387/1/012019

de Jong, T., Sotiriou, S., & Gillet, D. (2014). Innovations in STEM Education: The Go-Lab Federation of Online Labs. *Smart Learning Environments*, 1(1), 3. doi: 10.1186/s40561-014-0003-6

Drever, E. (1995). *Using Semi-Structured Interviews in Small-Scale Research: A Teacher's Guide*. Edinburgh: The Scottish Council for Research in Education.

Farah, J. C., Moro, A., Bergram, K., Purohit, A. K., Gillet, D., & Holzer, A. (2020). Bringing Computational Thinking to non-STEM Undergraduates through an Integrated Notebook Application. *Proceedings of the Impact Papers at the 15th European Conference on Technology-Enhanced Learning (EC-TEL 2020)*, 14.

Gatsby (4.7). (2022). [Computer Software]. Gatsby. gatsbyjs.com

Gillet, D., de Jong, T., Sotirou, S., & Salzmann, C. (2013). Personalised Learning Spaces and Federated Online Labs for STEM Education at School. *2013 IEEE Global Engineering Education Conference (EDUCON)*, 769–773. doi: 10.1109/EduCon.2013.6530194

Gillet, D., Vonèche-Cardia, I., Farah, J. C., Phan Hoang, K. L., & Rodríguez-Triana, M. J. (2022). Integrated Model for Comprehensive Digital Education Platforms. *2022 IEEE Global Engineering Education Conference (EDUCON)*, 1586–1592.

Gomes, N. R. C., Farah, J. C., Doran, R., & Gillet, D. (2019), A Light Pollution Simulator. *European Planetary Science Congress - Division for Planetary Science Joint Meeting (EPSC-DPS 2019)*, 1838.

Hohenwarter, M. (2002). *GeoGebra - ein Softwaresystem für dynamische Geometrie und Algebra der Ebene*. Universität Salzburg.

Learning Tools Interoperability Core Specification (1.3). (2019). IMS Global Learning Consortium, Inc. imsglobal.org/spec/lti/v1p3

Moore, J. & Churchward, M. (2010). *Moodle 1.9 Multimedia Extension Development: Customize and Extend Moodle by Using its Robust Plugin Systems*. Birmingham: Packt Publishing Ltd.

Perkins, K., Adams, W., Dubson, M., Finkelstein, N., Reid, S., Wieman, C., & LeMaster, R. (2006). PhET: Interactive Simulations for Teaching and Learning Physics. *The Physics Teacher*, 44(1), 18–23. doi: 10.1119/1.2150754

Puggioni, M. P., Frontoni, E., Paolanti, M., Pierdicca, R., Malinverni, E. S., & Sasso, M. (2020). A Content Creation Tool for AR/VR Applications in Education: The ScoolAR Framework. In L. T. De Paolis & P. Bourdot (Eds.), *Augmented Reality, Virtual Reality, and Computer Graphics*, 12243, 205–219. doi: 10.1007/978-3-030-58468-9_16

Tavares, R., Vieira, R. M., & Pedro, L. (2020). A Participatory Framework Proposal for Guiding Researchers through an Educational Mobile App Development. *Research in Learning Technology*, 28. doi: 10.25304/rlt.v28.2370