Building an IDE for an embedded system using web technologies

Liam Donovan, S. M. Astrid Bin, Jack Armitage, Andrew P. McPherson
Centre for Digital Music
Queen Mary University of London
Mile End Road
London, UK, E1 4NS
l.b.donovan, a.bin, j.d.k.armitage, a.mcpherson @qmul.ac.uk

ABSTRACT

Implementing an understandable, accessible and effective user interface is a major challenge for many products in the microcontroller and embedded computing community. Bela, an embedded system for ultra-low latency audio and sensor processing, features a browser-based integrated development environment (IDE) using web technologies (Node.js, HTML5 and CSS). This methodology has allowed us to create an IDE that is simplified and intuitive for beginners while still being useful to those more advanced, thus supporting users as they evolve in expertise.

1. INTRODUCTION

Embedded systems have increased in popularity as well as proliferation over the last decade. The popularity of platforms such as Arduino[1] can be tied to their simplified interface and focus on usability.

Based on research indicating the importance of perceptually low latency in the context of musical interaction [1], Bela was developed to provide ultra-low latency sensor and audio processing to musicians, audio developers, instrument designers, artists and makers. (Descriptions of Bela's history, technology and applications in research and design can be found in the citations listed in [1].) However, this is not a platform targeted only towards beginners, and the original toolchain and workflow were complex. We were challenged to develop an Integrated Development Environment (IDE) that powerful enough for experienced users, but also supported beginners and those just starting out.

The solution was a browser-based IDE, developed using web technologies. Sitting between the user and the system, the IDE features tools for programming, project management, technical referencing and device settings, and an oscilloscope for direct graphical interaction with system behaviour. This paper presents the design and architecture of Bela's browser-based IDE, how web technologies enable the platform to serve users with a wide range and depth of experience.

[1]https://arduino.cc

Web Audio Conference WAC-2017, August 21–23, 2017, London, UK.

2. ARCHITECTURE AND DESIGN

2.1 Technical description and interface

Bela is a Linux computer about the size of a credit card. Its primary application is for real-time audio and sensor processing for building embedded interactive audio applications (such as instruments, effects boxes, and installations).

The IDE runs on Bela as a Node.js[2] application (Figure 2). The interface contains a text editor and tab-organised tools above, and a horizontal toolbar and console below. The text editor, based on Ace.js[3], features syntax highlighting and auto-completion (Figure 1). The tab-organised section features a project and file manager, system settings, an interactive pin diagram, and an API reference (the C++ API is designed to be familiar to users of environments like Arduino and Processing). In addition, there are currently 12 chapters of examples available demonstrating techniques such as sensor use and audio effects, as well as fully functional digital musical instruments (DMIs).

The horizontal toolbar features buttons for running/stopping the current project, downloading files, clearing the console, opening the scope, as well as displaying current IDE and system CPU usage. The console displays output from the text editor, output from running programs, and system warnings.

The oscilloscope, launched in a separate tab, features time and frequency domain visualisation of multiple channels, plot resolution and axis scaling, and signal scaling and trigger settings. The visualisation is currently achieved by drawing signal data to an HTML Canvas. The oscilloscope is available for both C++ and Pure Data projects, in order to test, verify and demonstrate system behaviour.

2.2 Advantages for users, and developers

This system has major usability advantages for both beginner and expert users:

1. The system is plug and play.
2. It runs in a browser, which is widely familiar.
3. There is no complex toolchain to learn.
4. Integrated examples for reference and learning.
5. Access to all system functions is preserved.
6. Data visualisation with the built-in oscilloscope.

As a development team, we have the advantage that this

[2]https://nodejs.org
[3]https://ace.c9.io

Figure 1: User interfaces to Bela. Web based IDE Editor (above left) and Scope (below left), including an interactive pin diagram (centre). Hardware peripherals and form factor are compatible with digital making practices (centre). Online community, right, consists of: open source software and hardware development, forum, blog and DMI design repository.

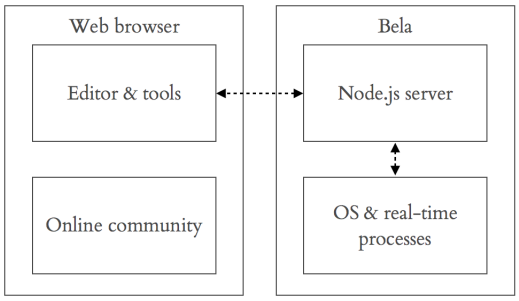

Figure 2: Users connect to Bela through their browsers via a direct network connection, giving them a single interface for both OS and real-time processes. Users additionally participate in online communities as described in Figure 1.

client-server arrangement allows Bela to focus its CPU capacity on the user's application code, while GUI delivery is handled by the browser. Additionally, using web technologies means that adjusting, testing and updating the IDE can be done quickly and iteratively in response to user feedback.

3. INDICATORS OF EFFECTIVENESS

Through our frequent workshops, user surveys, and feedback from those using Bela in education, we can report that we have found that beginners the browser-based IDE is extremely approachable, despite the steep learning curve of embedded audio programming. Further, experienced users appreciate features such as the integrated oscilloscope, and maintaining control of system functions.

The most encouraging heuristic is Bela's repeat customer base. Since launching as a commercial product[4] following our successful Kickstarter campaign, the user community continues to expand in number as well as application [2]. We also find that many institutions that use Bela for education buy more to expand their programs.

[4]http://shop.bela.io

4. CONCLUSION

Web technologies have been central to the development of Bela's browser-based IDE, and have enabled us to develop a system that appeals to users at a variety of levels of experience, and supports them as they gain expertise. This system also allows us to respond to user feedback by updating the IDE easily and often.

It should be noted that implementing a browser-based IDE is not a panacea for all usability challenges, and will not account for a poorly-designed interfaces, inadequate stakeholder knowledge, or other common causes of user dissatisfaction. However, the flexibility of a browser-based system offers significant advantages to users, and makes continuous improvement possible.

5. ACKNOWLEDGMENTS

This research is supported by EPSRC under grants EP/G03723X/1 and EP/L01632X/1 (Centre for Doctoral Training in Media and Arts Technology).

6. REFERENCES

[1] G. Moro, S. A. Bin, R. H. Jack, C. Heinrichs, and A. McPherson. Making High-Performance Embedded Instruments with Bela and Pure Data. In *Proc. International Conference on Live Interfaces*, University of Sussex, 2016.

[2] F. Morreale, G. Moro, A. Chamberlain, S. Benford, and A. McPherson. Building a maker community around an open hardware platform. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 2017.