A Markdown-based Open Toolchain for Writing, Processing, and Publishing Bebras Tasks

Jean-Philippe Pellet[1], Gabriel Parriaux[1], and Nora Anna Escherle[2]

[1] University of Teacher Education, Lausanne, Switzerland
{jean-philippe.pellet,gabriel.parriaux}@hepl.ch
[2] Senarclens, Leu + Partner AG, Zurich, Switzerland
nora.escherle@senarclens.com

Abstract. A considerable energy is put by the international Bebras community into creating tasks for each year's contest. Over each of the past years, more than 100 tasks were worked on and readied. They cover many aspects of computer science in fun, original ways, and are aimed at students from age 6 already to the end of high school, with several difficulty levels. Many of these tasks have high-quality explanations and references in their "It's Informatics" section.
In our opinion, they deserve both (a) a more automated processing workflow than what is currently being done, and (b) a better visibility than they are getting now during (and outside) the contest weeks. In this poster proposal, we propose a new Markdown-based task format and accompanying toolchain as concrete tools for an easier interchange of tasks and processing of task metadata. We show how many current aspects of task validation, selection, editing, and publication can be simplified with that toolchain. We also detail how its architecture and extensibility can be used to further repurpose tasks and republish them off-contest in an open platform for teachers to use them as teaching resources.

Keywords: Bebras tasks · Open toolchain · Teaching resources.

1 Introduction & Context

Bebras tasks go through a long process before they end up reaching students taking the yearly contest. They get conceptualized, written, modified, submitted to local/national task committees, commented on, resubmitted to the international workshop, rated, classified, etc.

Task files, conceptually, are very structured and can be seen as comprising several kinds of data:

- The task question itself, which will be shown to students taking the contest (comprising text and graphics);
- Additional well-defined task sections like the explanation of the correct answer and a section detailing its links to computer science that are typically available only after the contest;

– Comments and discussions on the task, its appropriateness, its formulations, etc.—typically used during the preparation process but not available to students or teachers;
– Metadata such as task ID, age classification and difficulty of the task, categorization with respect to fields of computer science, etc.

Considering the well defined structure, it may seem surprising to know that most tasks are written with LibreOffice, with some others written in HTML, with, as the files show, no way of ensuring a uniform way to get the task files to be formatted in a systematic, consistent way, and no easy way to automatically check the consistency of the metadata. Moreover, both formats allow a lot of liberty as to how tasks are designed and can look: in the actual contest systems displaying the tasks to the participants, more constraints or conventions can apply (e.g. image size, no change of font, no fancy page layouts, etc.).

The Bebras community has recently been split on the format issue. Arguments have been put forward in favor of one or the other solution: LibreOffice files are deemed easier to edit for non-technicians and being the better format based on free software for getting tasks from a potentially larger writer base. HTML files have been put forward as easier to process, being text files: better suited for version-control systems, easier to run queries against, etc.

In this poster, we detail the result of our taking up on a proposal made at the 2017 International Bebras Task Workshop in Brescia that suggested using Markdown as a base format. Markdown is a lightweight markup language designed by tech blogger John Gruber, whose basics can be learned in a matter of minutes and who was favorably compared to traditional word-processing software or other formats like HTML or LaTeX for writing or editing tasks whose complexity is limited [1, 3]. Its main strength is that its content are easy to process as they are traditional plain text files. it has for instance been used for several years in solutions for the creation and publishing of teaching materials in multiple formats from a single source representation [2].

We detail the principle of our implementation in the next section, and, in the poster, also highlight how some parts of the whole Bebras process benefits from these aspects (for instance, automatic task archiving and indexing, PDF brochure generation, or automatic upload to a task database).

The authors would like emphasize that this poster is not meant to be viewed as a scientific contribution to the field of computer science education per se, but rather as the presentation and proposal of tools that may be useful to members of the Bebras community and as a way to foster a discussion around them.

2 Outline of Proposed Format & Toolchain

This list highlights the main points represented in the poster. The whole solution is developed in TypeScript in the `node.js` ecosystem and can easily be adapted for embeddability in the browser.

– Tasks are written in the popular Markdown format. These are plain text files. The title and the order of appearance of the Markdown sections in a task

file is prescribed. Some Markdown extensions are provided (embedded LaTeX, image placement, etc.) as extensions to the readily available `markdown-it` parser.
- Task metadata like age classification and categorization is defined in a structured way in what is known as Markdown "front matter" at the beginning of the Markdown file. It is specified with YAML. It has a corresponding validation schema. YAML was chosen because it is more easily user-editable than JSON and is already used in similar "Markdown + metadata" projects.
- All of the above rules can be checked easily by some "linter-like" tool reporting errors and warnings. This tool operates on text files, uses existing Markdown and YAML parser, and is easy to integrate in an existing workflow: it can be e.g. used from the command line, producing compiler-like output with line and column information for errors and warnings. The tool can be used to validate submitted tasks from third parties and report reader-friendly feedback.
- From the base format, easy to derive are the following files:
  - HTML, in which the base Markdown is rendered with custom CSS that matches the look and feel of the LibreOffice files currently used
  - PDF, generated from the HTML file, paginated, numbered and with a chaptering
  - JSON, in order for easy importability in a DB or for powerful file-based processing and transformations with the `jq` tool
  - TeX, with automatic wrapping of numbers in inline-math contexts and other goodies—especially useful for systematically creating after-contest task brochures
  - CSV to export summarized data extracted from the metadata of tasks
  - other derived formats with potentially filtered output, in an effort to better reuse these tasks in teaching context outside the contest itself
- Editors can use any text editor, however, we provide a plugin for Visual Studio Code with a user-friendly user interface for rendering the Markdown tasks, automatically running the linter tool, thus highlights warnings and errors in the metadata, also honoring the Markdown extensions and rendering the metadata, and showing UI buttons for the described conversion tools.

3 Tradeoffs

Any change to well established practices meets resistance, and a change of the main editing tool used to write tasks also comes with tradeoffs. From initial discussion as well as feedback from reviewers, these include the following points. In the following points, the word "users" refers to writers or editors of Bebras tasks, or anyone in the subsequent country-specific workflows that have to process the task files.

- User-friendliness is often cited as a plus for LibreOffice, as virtually all users are expected to be familiar with some level of familiarity with a typical word-processing software. Fewer users are expected to be familiar with raw text

files and VS Code. We note, again, that VS Code does offer a live rendering of Markdown files: what users would have to learn, however, is the Markdown syntax.
- Cross-platform availability has been cited as a plus for LibreOffice, but HTML and Markdown edition is actually even a more open approach, since virtually any platform has some form of text editor, and real-time preview of tasks via VS Code will be available on three major platforms (Windows, macOS and various Linux flavors). Once VS Code is installed, of course, no internet connectivity is required to work on the tasks. Actually, LibreOffice may even be at a loss here in the cross-platform comparaison, as multiple users have reported usability problems with basic scrolling in a simple document on macOS[3], and its look and feel does not feel platform-native.
- HTML files can embed JavaScript code which can be used to provide an initial implementation of interactive tasks. LibreOffice and Markdown tasks cannot do this. But since the VS Code preview is a live HTML view, it would be relatively easy to add to the metadata a list of scripts to load in that VS Code preview if the community deems it necessary.

4 Outlook

We think that the proposed format and open toolchain can bring interesting perspectives to the community, improving existing processes with the two current formats in HTML and LibreOffice. This solution offers convenient ways to automate the processing of the tasks all along the process, from the creation to the use in various contexts (online, paper-based, query, etc.). It remains low-tech for the end user (mere text editor in the simplest case), while providing command-line tools and extensibility for power users. Relying on this, we may expect teams involved in the Bebras community to be more efficient and have more opportunities to exploit the content of Bebras tasks.

References

1. Ovadia, S.: Markdown for librarians and academics. Behavioral & Social Sciences Librarian **33**(2), 120–124 (2014)
2. Roganov, E., Roganova, N., Aleksandrov, A., Ukolova, A.: Web portal for dynamic creation and publication of teaching materials in multiple formats from a single source representation. In: AIP Conference Proceedings (2017)
3. Seo, J.Y., McCurry, S.: LaTeX is not easy: Creating accessible scientific documents with R Markdown. In: Proceedings of the 34th Annual Assistive Technology Conference (2019)

[3] https://bugs.documentfoundation.org/show_bug.cgi?id=113104