

The 9th International Scientific Conference
eLearning and software for Education
Bucharest, April 25-26, 2013
10.12753/2066-026X-13-132

ARROW BRAINSTORMING APPLICATION

Cătălin CHIȚU, Sabin MARCU

Nicolae Grigorescu National College, Natural Sciences and Technologies Departments, No.4 Doftanei Street, Campina,
Prahova County, Romania
catalinchitumail@yahoo.com, sabinmarcu@gmail.com

Abstract: Arrow is a brainstorming application with millions of uses. The starting point for this application was creating an Outliner for the web. An outliner is an application that creates, in essence, nested check lists. However, one of the formats in which an outline can be saved is the OPML format, which can be used for a lot of other things, one of which being mind-mapping. Mind-mapping is a way of displaying ideas as an idea tree, starting from the most basic and general topic or the concept itself, and then the branches go in deeper, analyzing a particular subtopic of its parent. Sounds familiar? This is the idea behind outlines, as well. Starting from this idea, and the two ways of displaying the data, there is no limit to the domain in which the Arrow application can be of use. Education, development, or maybe just domestic to-do lists. Everything goes. Arrow is a web application designed to work in the cloud, connecting people to share ideas, outline projects and imagine the world. Take education for example: A teacher can always bring a summary of the lesson to class structured as a mind-map or an outline. This way, he can keep track of what he's taught already, and what to teach next. At the same time, student can be given access to that outline, and can follow the teacher as he goes through it, and furthermore, can use it to study easier. Possibilities are endless. It all depends on the user!

Keywords: Web Application, Outliner, Mind-Mapping, OPML, Cloud.

I. WHAT IS A BRAINSTORMING?

Brainstorming [1] is a technique often used in any great companies' conference rooms, but fairly less known and used in the rest of the world. This fact is, unfortunately, in our disadvantage. After all, brainstorming is the process that goes on inside everyone's' mind when he/she is presented with a dilemma.

Brainstorming is the graphical representation of an idea under a tree form with notes attached to each information node. This way, the initial idea (or ideas) represent the root node of the tree, and the branches represent explanations, developments of certain ideas or concepts related to (and relevant for) the main idea. The human brain creates these links through the way it is built. We can say without hesitation that this is the way in which any human thinks, in which the human mind processes a given situation and defines the way in which you must handle it.

The result of a brainstorming can be represented in two ways: Mind-Map [2] and Outlining [3].

Mind-Map is the process through which the nodes of a brainstorming are drawn under the form of text containers tied together by symbolic links. An example of a Mind-Map is presented in Figure I.1.

Another method to create a mind-map, much less known, even to the big corporations, is the technique called Outlining. Outlining takes that same tree structure and represents it linearly under the form of nested lists. Thus, after the logical, graphical definition of the idea structure, one can then move on to developing and tracking it through time under an easier form (figure I.2).

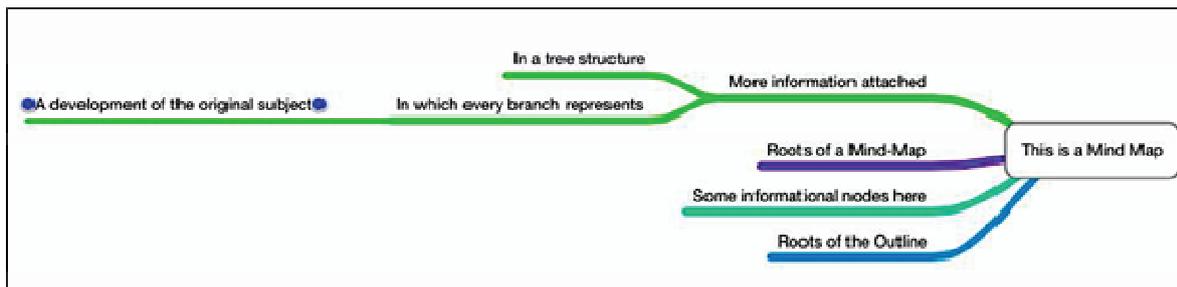


Figure I.1. Example of a Mind Map

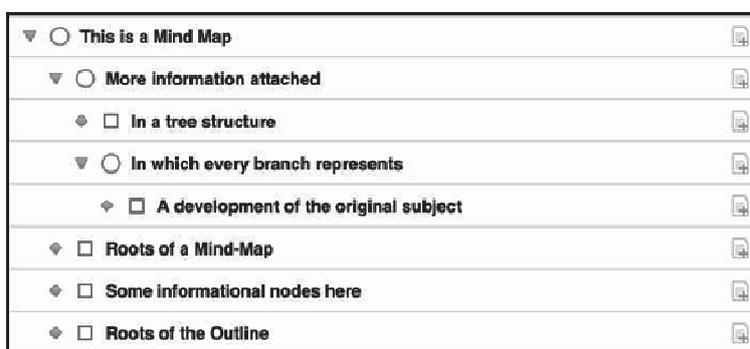


Figure I.2. Previous example in outline form

In the case of large scale projects, for example, an efficient process would be to start off with a Mind-Map based Brainstorming, with starting point the primary idea and fully developing the concept, in a radial fashion, later on converting the Mind-Map into an Outline, and, thus, tracking the progress in developing the project on the base of the original concept.

Even more useful is its application in Education. A teacher can build its lesson for today basing it on a Mind-Map, and giving his students (even those who are absent on medical basis but want to participate) access to it, or even building the brainstorming on the fly with his students.

II. EXAMPLE

Differentiated training represents the complex activity of adapting teaching components to the different nature of student's personalities. These can be typologically grouped or considered individually with rigorously individualized particularities.

Individualized training is subordinated to differentiated training and represents the action of adapting the whole training-educational process to the individual particularities of each student. Great attention is paid by both differentiated and individualized training to individualized activities with small groups or individually realized. (Figure II.1).

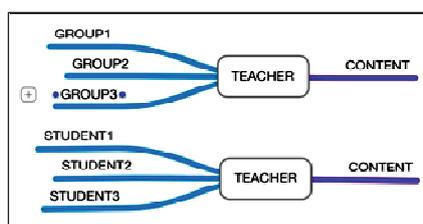


Figure II.1. Individualized and differentiated training [33]

Obtaining competences on students is the result of training objectives, through formed abilities, skills and proficiencies. A proposition of an implicit relationship between these terms is represented in Figure II.2.



Figure II.2. Relation between educational factors [33]

In the purpose of differentiated treating of content, we can exemplify a laboratory experiment with the objective of determining the elastic constant of a rubber band.

The experiment fits in the "Variation theorems and conservation laws in mechanics" teaching unit. The corresponding conceptual map is represented in Figure II.3 [33, 34].

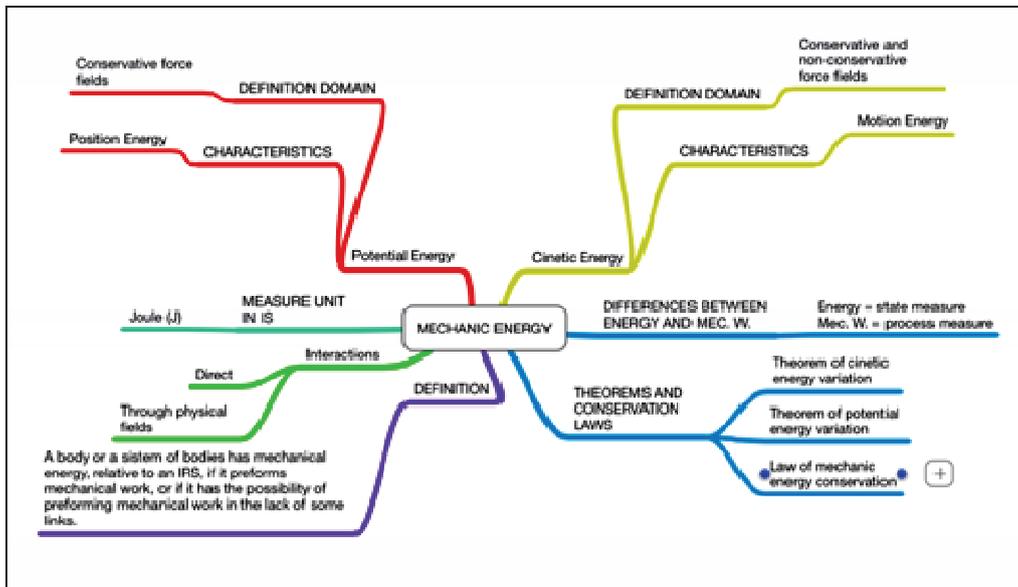


Figure II.3. Conceptual Map [33, 34]

III. THE ARROW APPLICATION

3.1 Defining the application

The ARROW application is, first of all, a WEB application [5]. That means it is universal, it can be accessed on any kind of device, at any given moment, with no differences whatsoever (well, maybe not Microsoft IE [6]). Any modern browser is compatible and certain browsers support even partial and even complete integration with the operating system. On any desktop unit, Mozilla Firefox [7] and Google Chrome [8] support partial integration with the operating system under the form of the so-called "packaged apps", as Mozilla Firefox OS [9] and Google Chrome OS [10] have as native applications web applications, so full integration.

Arrow is an application built to facilitate collaboratively, in real time, the brainstorming process, supporting both mind-mapping and outlining. Collaboration is made online, in real time,

much like the Google Docs [11] application (full synchronization and marking of the current modifications for all participants), but anonymously (each connected user receives a unique ID, which he can share with other people, thus connecting them).

The application uses a file type called OPML [4] (Outline Processor Markup Language) to save the nodes of a Brainstorming and, on the base of it, drawing the tree in a Mind-Map or drawing the lists of an Outline. At the same time, the application uses the Drag and Drop API of HTML5 [12] to integrate with the operating system, the use mode is described in the next section.

The application has a sidebar-content layout present and common to all device types. The design is responsive (it changes based on different device resolutions). Also, it is size-dependant, not vendor or device-dependant (that means that it reacts to size change rather than targeting a specific device and creating an entire design for it). An example of the application design is presented in Figure III.1.

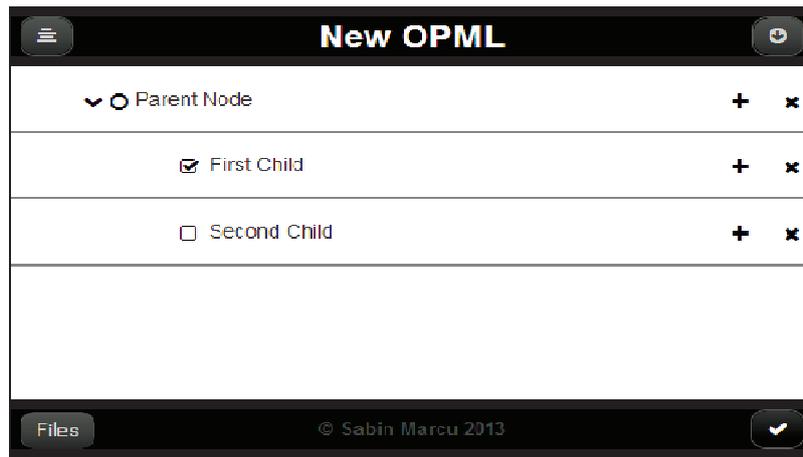


Figure III.1. An outline view of the application

Also, the application uses a proprietary system for connectivity that can run independent of the application compilation and distribution server (main server). This means that one can install the server on one machine in an intranet and connect clients to that specific machine. The server issues a unique ID for each client connected. This ID is then used to connect two clients (or more) together (the first sends the ID to the second client via any other means, and the second connects to the first client's ID). An example of the connection interface is presented in Figure III.2.

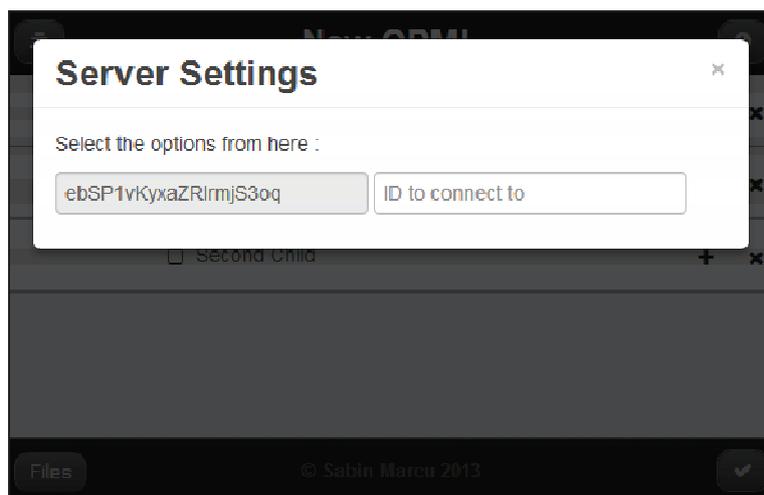


Figure III.2. Connectivity settings for one client

The application has different controls between desktop apps and touch-based devices: for desktop browsers, it relies on context menus to do some actions, while on mobile devices buttons

appear to make it easier to realize certain tasks. The example in Figure III.1 (and in the background of Figure III.2) is running in mobile mode: The two buttons on the right of each Outline node represent Add and Delete (adds a new node underneath this one and delete the current one).

To save data on any client (not necessarily on a desktop) the application uses Local Storage (at the moment: an upgrade to Indexed DB is planned soon). The data saved is in OPML format (so it can be re-read exactly as it was inputted in the first place). Also, the last opened OPML (Outline and/or Mind-Map) is saved, and an index of records. An example of a data structure (with Google Chrome Inspector) is presented in Figure III.3.

Key	Value
lastOPML	0
opmls	["New OPML","Big Ideas"]
opmls.Big Ideas	<opml version='1.0'><head><title>Big Ideas</title></head><body>...
opmls.New OPML	<opml version='1.0'><head><title>New OPML</title></head><body>...

Figure III.3. An example Local Storage layout for the application

A lot of other applications use OPML as a file type. For example, for IOS: Omni Outliner, Outliner, Simple Mind, Mind Node and many others. Drop box integration is planned in the near future to connect to all of these, in an easier way. Also, Google Drive integration is planned for sometime next year, when everything else is thoroughly tested.

3.2 Technologies used

Arrow is built on the Node JS platform on the server side, and HTML5 [15] with JavaScript [16] on the client side.

On the server side, I have used a proprietary packaging system extended from Sam Stephenson's "Stitch" [17] project, which compiles into one file all logic elements, visual styles, HTML elements and JSON [18] data, minifying at the same time the sources, taking care that there are no comprehension errors. That being said, the application has, on the client side, a unique, single source, with a very small size, which makes the accessing and using of the application on any electronic device, anywhere possible.

On the client side I have used Coffee Script [19, 32] to write the logic and interface elements, Stylus [20] and Nib [21] on the visual styles side, Eco [22], Jade [23], Markdown [24] and HTML5 for the interface part. As auxiliary libraries I have used Angular JS for the bidirectional links between the abstract data and the useful information that the user can access, Twitter Bootstrap [25] for the modal windows and other auxiliary elements and j Query [26] for DOM [27] manipulation.

To communicate between the client and the server (to connect to other clients) the application uses the Node JS [13] module called "Socket. IO" [14], and the actual communication with one of my own modules which extends the functionality of Socket.IO, making the connections between clients anonymous.

Coffee Script [32] is a programming language inspired by Ruby [28] and Python [29] (mainly), which compiles into vanilla JavaScript, avoiding many of the well-known problems of the language.

Stylus is a descriptive language inspired from CSS [30] and Python which eliminates many of the structural rigors of CSS. The language supports including files and compiling them in a single CSS file. Nib is a library of Stylus which makes prefixing and correcting the experimental properties of CSS3 [31] possible, to ensure the compatibility between browsers (including Internet Explorer - not that it would matter too much).

Eco and Jade are two templating engines, which based on some arguments create a large amount of HTML elements, straight from JavaScript. Eco has a syntax similar to Coffee Script, but the elements that are not related to logic are HTML, so a lot of information which could be abstracted. Jade has a much lighter syntax, containing 0% HTML (except when the programmer thinks they are necessary) but, unfortunately, the same structure proves sometime to be unsuited for the application.

Markdown is a text programming language which compiles into HTML5, but is created via abstracting the language, to make the document itself more readable, and much easier to write and edit. Markdown is used only in the case of static documents (landing page, help page, etc).

IV. CONCLUSIONS

The world is changing. The tendency is to go towards the Web. That is why all our efforts should go in that direction. The Arrow application is meant to be more than just a simple responsive reflexive web brainstorming application. It's meant to be a beacon, a light to guide future applications on the way to the future. It is meant to reach the public in a way that it has never been reached thus far, or only in a shallow manner.

Brainstorming itself is becoming a big deal. It has millions of uses, and is unknown to more than just a million people. Arrow is an attempt to make it more accessible. Also, brainstorming has a huge importance especially in the field of education, because it molds onto every student's thinking pattern. It is a perfect fit.

References

- [1] <http://en.wikipedia.org/wiki/Brainstorming>
- [2] http://en.wikipedia.org/wiki/Mind_Map
- [3] [http://en.wikipedia.org/wiki/Outline_\(list\)](http://en.wikipedia.org/wiki/Outline_(list))
- [4] <http://en.wikipedia.org/wiki/OPML>
- [5] http://en.wikipedia.org/wiki/Web_application
- [6] <http://window.microsoft.com/ro-ro/internet-explorer/>
- [7] <http://www.mozilla.org/en-US/firefox/new>
- [8] <http://www.google.com/chrome>
- [9] <http://support.mozilla.org/en-US/kb/how-set-firefox-os-desktop-build>
- [10] <http://www.chromium.org/chromium-os>
- [11] <http://docs.google.com/>
- [12] http://developer.mozilla.org/en-US/docs/DragDrop/Drag_and_Drop
- [13] <http://nodejs.org/>
- [14] <http://socket.io/>
- [15] <http://www.w3schools.com/html5/>
- [16] <http://en.wikipedia.org/wiki/JavaScript>
- [17] <http://github.com/sstephenson/stitch/>
- [18] <http://www.json.org/>
- [19] <http://coffeescript.org/>
- [20] <http://learnboost.github.com/stylus>
- [21] <http://visionmedia.github.com/nib>
- [22] <http://github.com/sstephenson/eco>
- [23] <http://jade-lang.com/>
- [24] <http://daringfireball.net/projects/markdown>
- [25] <http://twitter.github.com/bootsrap>
- [26] <http://jquery.com/>
- [27] http://en.wikipedia.org/wiki/Document_Object_Model
- [28] <http://www.ruby-lang.org/>
- [29] <http://www.python.org/>
- [30] <http://www.w3schools.com/css/>
- [31] <http://www.w3schools.com/css3/>
- [32] Stefan, A.; Catalin, C.; Alexandru Catalin, T.; Sabin, M. (2012). Elements of Interdisciplinary Instruction in the Natural Science Curricula. In *eLSE 2012, Proceedings of the The 8th International Scientific Conference eLearning and Software for Education*, Volume II, pp. 86-91, University Press
- [33] Catalin, C. (2011). *The Doctoral Thesis*, University of Bucharest, Faculty of Physics, 2011, p. 56.
- [34] Georgeta, I.; Emil, B.; Catalin, C. (2010). Interactive Training at Physics Classes through Conceptual Map. In *EDULEARN 2010, Barcelona, Proceedings of the International Conference on Education and New Learning Technologies*.

Copyright of eLearning & Software for Education is the property of "Carol I" National Defence University and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.