Rapid Application Development to Create Proof-of-Concept Software Applications

Margarida Lucas da Silva[1,2], Hugo Silva[3] and Daniel Gonçalves[1,2]
[1]*Instituto Superior Técnico, Av. Rovisco Pais, 1, 1049-001, Lisboa, Portugal*
[2]*INESC-ID, Rua Alves Redol, 9, 1000-029 Lisboa, Portugal*
[3] *Instituto de Telecomunicações, Instituto Superior Técnico,*
Av. Rovisco Pais, 1, Torre Norte - Piso 10, 1049-001, Lisboa, Portugal

Keywords: Rapid Application Development, Rapid Application Customization, Autism Spectrum Disorders.

Abstract: Rapid application development is the best way to test prototypes by giving a solid performance for user's tests, while rapid application customization it is the best approach to easily test the user's needs, such as children with autism spectrum disorders. In this paper we present a framework of a platform designed with these concepts in mind. This platform is a standalone multimedia and rich content software, targeted at students with special needs, that allows to easily expand the functionalities and create proof-of-concept software applications.

1 INTRODUCTION

Software has evolved giving several potential solutions for our everyday problems, but still can be difficult to reuse a good tool and adapt it to a specific need, since the target is typically the general population. This brings a need to have access to a tool that allows to easily develop diferent functionalities and adapt the tool, with end-users in mind. my.Troc@s.net comes as a answer to create fast prototypes and easily create a specific tailored tool. The constant evolution in ICT brought a wide range of new possibilities to help people with learning disabilities access new opportunities for learning, entertainment and personal development. Children with ASD are included in this group, and although many approaches have been tested, few have been adopted or successfully integrated in the daily life of the children, their tutors or caregivers (Putnam and Chong, 2008). Also, it is difficult for tutors to easily adapt the tool for each child, which makes the need to create a tool with two end users. It is still difficult for tutors to become autonomous in the process of managing and adapting the available tools to the needs of children with ASD; we haven't found usability studies specifically focused on this aspect but the tools that we have reviewed to date generally require a backend for content management, and as such, the process is not as easy as it could be. Web technologies and related standards have provided several well-supported means, which can be easily mastered by a broad audience; these technologies can be extended to Augmentative and Alternative Communication (AAC) software development, promoting the involvement of the "user in the loop" due to the simplicity they provide. In this paper we describe a standalone software application for promotion and training of social communication skills in children with ASD. It is based on a framework for rapid application development (RAD), and rapid application customization (RAC). The underlying framework can be generalized for the development of software targeted at people with special needs; the main design principle for this approach was to enable tutors and professionals in the field to fully manipulate the overall contents and look-and-feel autonomously with little training or support. We will describe an application called myTroc@s.net, that takes advantage of the framework presented, but other examples of applications are also available at (Lourenço et al., 2013). Some experimental results with the platform are available in (Lucas da Silva et al., 2011). This paper is organized as follows: Section 3 goes through the main functionalities of Troc@s; Section 4 explains the concept of profiles as a way of achieving the rapid application customization for tutors; Section 5 describes the RAD/RAC framework depicting how we achieve the desired flexibility and adaptability; and finally, Section 5 summarizes the main results and conclusions.

Lucas da Silva M., Silva H. and Gonçalves D..
Rapid Application Development to Create Proof-of-Concept Software Applications.
DOI: 10.5220/0004698202990306
In *Proceedings of the International Conference on Physiological Computing Systems* (PhyCS-2014), pages 299-306
ISBN: 978-989-758-006-2

2 STATE-OF-THE-ART

In the last decades, a number of software applications have been developed for people with special needs, mostly targeting computer-based learning. These however, have been developed in a more enabling and functional perspective centered in the individual himself (Emiliani et al., 2009) (Helal et al., 2008) Specifically designed for children with ASD there is the Zac Browser[1], which allows autonomous usage of the Internet, together with other tools as activities, games, and videos. This platform is especially designed for recreational and entertainement use, as such, it does not explore the interpersonal communication specificities. Furthermore, tutors are limited at a content manipulation and tool adaptation levels. Boardmaker[2] is a suite of applications that enables the adaptation of learning materials to the needs of each child. The program has a large database of symbols they call Picture Communication Symbols (PCS)™, and allows the users to create any kind of customized material, such as schedules, manual communication boards, books, teaching materials, etc. The material can be used in the computer or printed, making this tool usefull to create materials for teaching, however, this software is designed for the tutor and was not conceived to be used by children. TeachTown[3] is a Computer-Assisted Instructional (CAI) program designed for developmental ages between 2-6 years, specifically to meet needs of ASD population. The program includes a comprehensive curriculum that aligns to standardized measures, which integrated the following learning domains: 1) Language Development; 2) Social and Emotional Skills; 3) Adaptive Skills; 4) Cognitive Skills; 5) Language Arts; and 6) Mathematics. The curriculumdepends on the performance and is adjusted for each student. Tutors can customize the curriculum to meet Individual Educational Plan (IEP) goals. TeachTown also collects data from the children usage and produces reports, to track the student progress. This tool is one of the most complete applications for children with ASD currently available, and extensive research has proven that an ICT tool like TeachTown has beneficial results. However, this tool is not completely customizable, and it is not targetting all the children preferences, which can have poor effects in their attention and engagement. Furthermore, this tool was devised for individual use, and lacks features that can promote communication and social interaction skills. Although not specifically designed for children with ASD, another tool of possible application is the Grid 2[4]. This is a customizable solution that replaces the conventional keyboard and mouse by a pointing device applied in a virtual keyboard, through which it is possible to use any software on the computer. However, this is a platform aimed to a different target population, with special needs in terms of accessibility.

3 FUNCTIONALITY

my.Troc@s.net was designed with the purpose of encouraging socialization, facilitating interaction among children with ASD, and also allowing tutors to easily customize the contents and overall look-and-feel of the platform.

The application can be fully customized with icons, background images and the contents that allow each child to be more effective while using it. Children can even use their own drawings or favourite images in a semiotics approach to better associate each feature of the software with the underlying function. Tutors have full control over the software and they can intervene at virtually all levels. They can act at the content level, where contents are structured and organized under the application folder tree, just as any other document is organized in regular computer work. But they can also act all the way up to the presentation level, where with basic training they are able to manage the screen and navigation structure by simply editing the HTML files.

All the activities are targeted at communication skills training. They are based on multimedia content: images, videos, audio, and stories that children can autonomously browse through and learn from. In addition there is also a message board, and content preference sharing options, these being the main contribution of the platform to the field, when compared to existing work.

In the message board children can post messages to their colleagues. The sender child has a buddy list with names and photos of all counterparts, from which the receiver colleague can be chosen. The message can then be written either through text, picture exchange (Bondy and Frost, 2001), or a combination of both methods. When the receiver colleague opens the message board, he will immediately be able to see all of the messages sent to him by others. Figure 1 shows the message board screen.

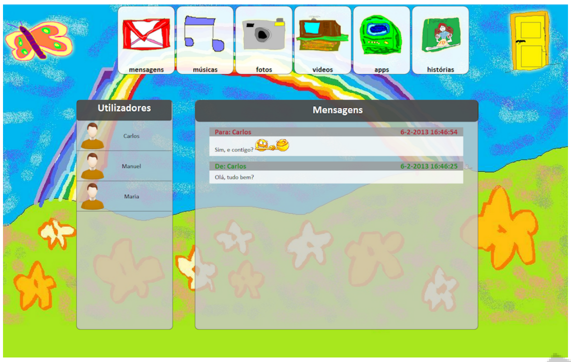
For content preference sharing, associated to each multimedia item (image, sound or video), there is a positive and a negative preference option. These al-

[1] http://zacbrowser.com

[2] http://www.mayer-johnson.com/boardmaker-software

[3] http://web.teachtown.com/

[4] http://www.sensorysoftware.com/thegrid2.html



Para: Carlos 6-2-2013 16:46:54
Sim, e contigo?

De: Carlos 6-2-2013 16:46:25
Olá, tudo bem?

Figure 1: Message history screen of myTroc@s.net.

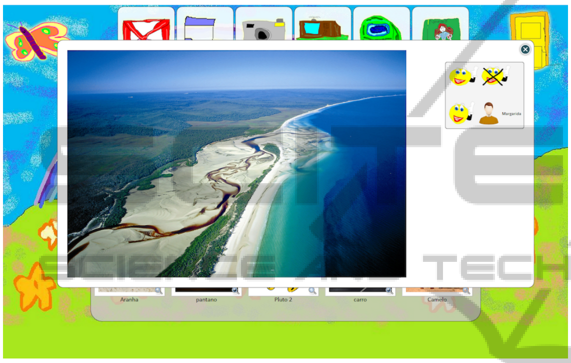


Figure 2: Image detail with the content preference sharing pane.

low the child to express and share with its counterparts at any time, his/her preference regarding a given individual element. This information is associated with the content and shown every time it is accessed, allowing all children to see the opinion of their colleagues. Figure 2 illustrates one of the tasks of the platform, with the content preference sharing pane on the right side of the screen.

4 PROFILES

As a way to empower the tutors we created a possibility of having user-tuned content as a way of further exploring the individual differences of children with ASD. This problem is not exploited enough in order to improve the children engagment while using software that can be beneficial for them. In this section we will detail our proposed methodology to the problem of profile management.

4.1 Concept

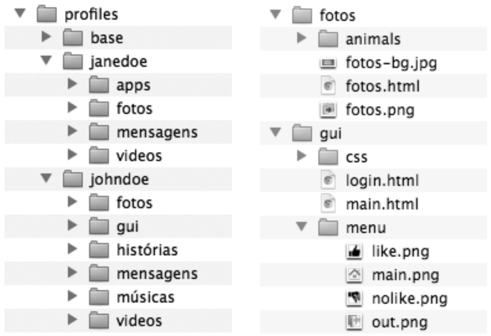
The profile concept arises from the need, shown by tutors to create customized layouts and content for the children. As in content management, we needed the process to be straightforward and familiar to the tutor, and to create a mechanism that allowed them to easily customize the whole platform, taking into account different user profiles. We decided to support this feature also on a folder structure of the Operating System (OS), since it was a fast and easy way to handle large amounts of files in bulk, and also because it involves procedures that the tutor are familiarized with. Using the “My Documents” metaphor available in all Operating Systems, we created profiles as if they were the “My Documents” of each user. This allows tutors to manage all the files individually, or to create shared profiles, without having a limited number of profiles per user. This method is usually familiar to all tutors and simplifies the process of managing content when compared to the typical backoffice or in-program settings management approaches.

By creating profiles, the platform can be fully customized to a user with icons, background images and the contents that allow each child to be more effective while using it. Children can even use their own drawings or favorite images to better associate and manage the semiotics of each feature of the software with the underlying function. With myTroc@s.net children can have their own individual profile on the platform, and visualize the application in the most engaging way for them.

Tutors can act at the content level, where contents are structured and organized under the application folder tree, just as any other document is organized in regular computer work, as we can see in Figure 3(a). But they can also act all the way up to the presentation level, where with the right training and proficiency they are able to manage the screen layout and navigation structure by simply editing the HTML files (Figure 3(b)). The previous customization approach provided by Troc@s was still preserved; if the tutor chooses to do so, he can use the *base* profile provided by the myTroc@s.net framework, and simply customize the platform in a one-size-fits-all approach where all of the children access the same customization settings.

The application reads all folders and file names, and uses them as content titles. This way, the application is not bound to a main language, by using everything that is specified by the tutor in the folder and filenames. The activities presented to the child in the platform are determined by the subfolders in the profile folders assigned to each child, so if for example the *pictures* folder didn't exist in the profile folder, then it wouldn't be available as an activity in myTroc@s.net. Also, the *base* profile is available as default for all children.

The tutor can create as many profile as he/she



(a) myTroc@s.net content folder tree for a given profile.

(b) A few of the system files used by myTroc@s.net.

Figure 3: FMS-based profiles, content and layout organization.

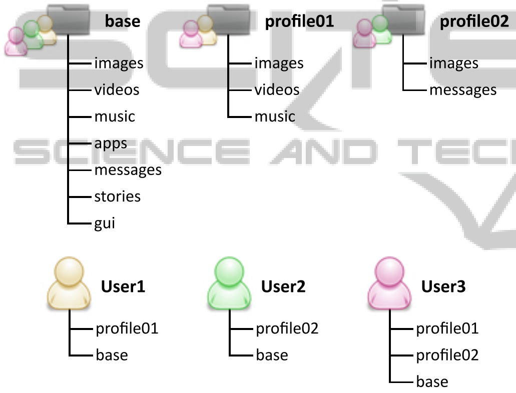


Figure 4: Example of the profiles structure in myTroc@s.net.

want's, and assign them to any user; in turn, each user can have an unlimited number of profiles, which allows a better organization of contents. For example if the tutor wants to share a class trip pictures, but not all children in the school went to that trip, he can create a profile for the class trip group, make the pictures available there, and only the corresponding children will have access to those pictures.

4.2 Management

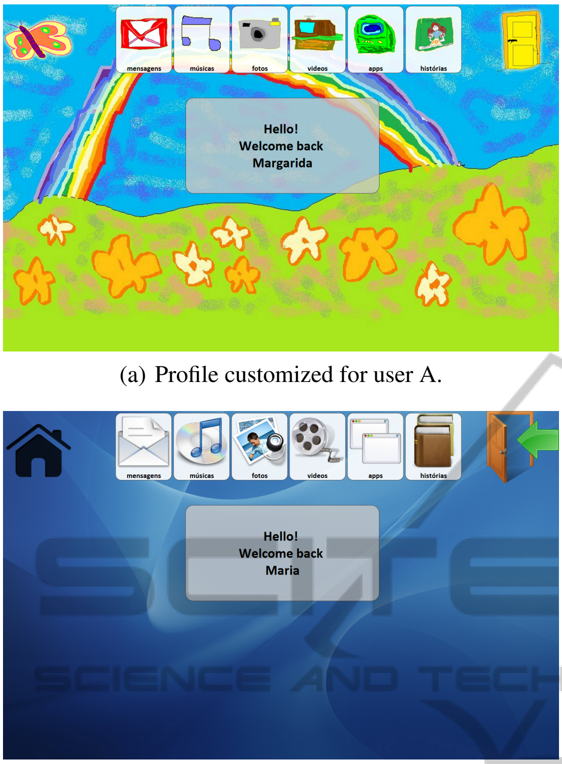
Profiles can be used individually, or shared with any number of users; for example, if the tutor has children from different age groups that can share a few of the activities (e.g. apps), he can create a common profile for each age group and selectively assign it to each child. When adding a profile to a child, the platform gives priority to the leftmost profile specified, and so on. This means that the content listed in the first profile specified, will be the first to appear in the content list view.

Profiles are specified in the a *settings.json* file that is associated with each user, and where the tutor only has to write the profile name with quotes, separating profiles with a comma, for example: *"profile":["profile02", "profile01"]*. In this example, the application first lists the *profile02* content, then the *profile01* content, and finally, all content in the base profile, which as the default profile, does not need to be specified in the profile lists, and is automatically available for all users.

In Figure 4 we can see an example on how the profiles are applied; the *base* profile is available for all children, and profiles *profile01* and *profile02* are only individually and selectively assigned to a few children. In each profile it is possible to override the content folders that one wants to customize; in *profile01* images, videos and music are being customized, while in *profile02* only images and messages are being customized. In each folder the tutor can change the menu icon, the background image, or the content; to do so, he/she just needs to copy and paste any image on the corresponding folder and rename it:

- For the menu icon, the image must have the same name as the folder. For example, the folder is called *images*, then the icon must be called *images*. The file extension can be of any image kind (e.g. jpg, png, bmp);
- For the background image, the image must have the same name as the folderwith the *-bg* suffix. For example, if the folder is called *images*, then the background image must be called *images-bg*, and again the file extension can be of any file image.

We can see examples of this structure in Figure 3(b). We also have a *gui* folder, where the user can change the *like* and *no like* icons (shown in the preferences sharing pane), the *main* icon (that the child uses to return to the main activity selection screen), the *out* icon (that the child uses to close the platform), and the HTML and CSS of the presentation layer, in case the tutors learn how to adjust with it. All content available in a a given user profile will only the accessible to the user(s) that have that corresponding profile assigned. Using profiles allows the children to have myTroc@s.net completely prepared for them, following their preferences. Figure 5 illustrates a customization of myTroc@s.net with two different profiles.



(a) Profile customized for user A.

(b) Profile customized for user B.

Figure 5: Example of content customization in myTroc@s.net.

5 FRAMEWORK

The framework was designed with both the end-users and the people behind application management in mind, targeting the seamless delivery of rich multimedia content, while providing high customization levels, easy content management and maintenance by non-technologically proficient individuals. Figure 6 presents an overview of the framework, which will describe throughout the following sections.

5.1 File Management and Data Persistence

In myTroc@s.net we used a Model-View-Controller (MVC) development approach; we implemented a Python Websocket (Wang et al., 2013) (Flanagan, 2011) server, which takes advantage of the advanced connectivity capabilities introduced by HTML5. In this approach, we use the browser just as a front end for content presentation and user interface, and all of the automation, OS-level operations, and computing tasks are supported by the Python programming language and framework. We chose the use of WebSockets since they are a very straightforward way of interfacing the browser with the “outside world”, and because it has the advantage of providing a full-duplex communication channel that can be used for asynchronous interaction between the browser and Python (in our case). Through this infrastructure, the presentation layer implemented at the browser level, sends requests to the Python back-end using a Message Passing Protocol (MPP), for Python to manipulate the file system, read all files from the platform, automatically generate thumbnails, among other operations.

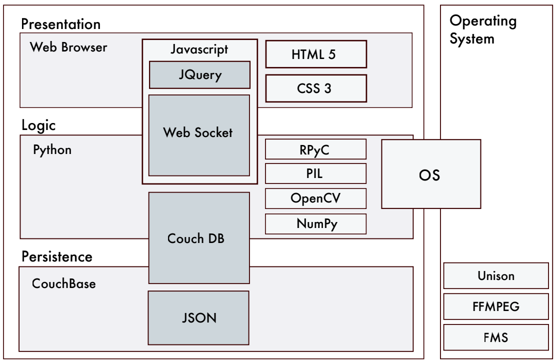


Figure 6: Framework of the myTroc@s.net.

For data persistency we used the Couch DB[5] database engine, which is also managed by our Python back-end. We selected Couch DB due to the fact that it is a full featured database management system, with the advantage of being document oriented and natively structured in JavaScript Object Notation (JSON); as previously described, JSON is directly supported by JavaScript, providing a streamlined way of storing information produced at the presentation layer in the database and vice versa. All preferences (*like*/*dislike*) assigned to the content and messages sent by the child are saved in the database as a JSON string associated to each user. myTroc@s.net also saves logs with the user navigation, files loaded, messages sent/received, likes, and amount of time spent in the platform. All events that occur in the platform are monitored by RPyC - Remote Python Call[6], which listens for all user interaction events, triggers the corresponding actions (e.g. new message warning or new file detection) and saves the event description in logs.

5.2 Networking

Another feature incorporated in myTroc@s.net is the support for networked and parallel operation, which

[5] http://couchdb.apache.org

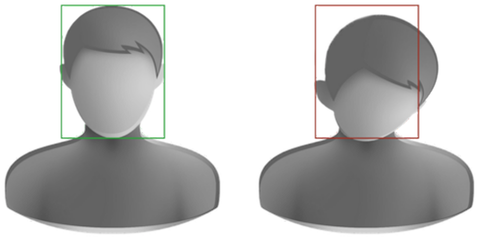
[6] http://rpyc.readthedocs.org

enables the automated synchronization of files across computers, and also the simultaneous and near real-time message exchange between children as any other chat client (unlike the monolithic message board approach followed in Troc@s). This gives enables the child to access his/her customized content from every networked computer, and also to communicate with other children and tutors using the myTroc@s.net platform in other parts of the school or even in other schools, since the platform synchronizes content and messages across the network, allowing children to commute and communicate with others outside of the physical space of the school. To support this feature in our platform we recurred to the Unison File Synchronizer[7], allowing the tutor to populate myTroc@s.net in his personal computer and automatically propagate the changes to the individual computers of each child in a Dropbox[8]-like fashion. This functionality was merely instrumental for the purpose of our study, since it makes the customization process more convenient for the tutors, however, it also sets an important background for the real-world deployment of the outcomes of our work as well as for future work to be developed.

5.3 Attention Detection

To detect the user attention while using myTroc@s.net, which will be used to evaluate the child's engagement in the platform, we used the Python OpenCV wrapper[9]. For this feature the myTroc@s.net framework uses the computer's camera, together with a Haar Feature-based Cascade Classifier for Object Detection for automated detection of the face of the child. Using the frontal and profile face cascades, if the child is not in front of the computer, or not looking at the screen, the platform detects it, and records the distraction state in the log file; on the other hand, when the child returns to the computer or looks again at the screen, the attention state is detected and recorded in the log file. Figure 7 illustrates the attention and distraction cases.

For the tests, a log system was implemented in the myTroc@s.net platform, which saved the current user, together with the timestamps in which we looked at the screen, and in which he looked away. Taking into account a physiological head movement frequency limit of 0.5Hz (Calais-Germain, 2007; Berthoz et al., 1992), the OpenCV was setup to only check for the head position every 2 seconds.



(a) The child is in front of the computer and looking at the screen.

(b) The child is in front of the computer but is not looking at the screen.

Figure 7: Attention detection while using the myTroc@s.net platform.

5.4 Logs

In order to record the actions and interaction patterns of children while using the myTroc@s.net platform, our framework was fitted with a logging mechanism which enables us to derive objective measurements that can be used to characterize the experience of the child when using the platform. For each user, a new log file is created every time he logs in, and the current date and time is assigned as the file name. The framework produces 6 different types of logs through which we can audit the child's behavior; per user session it records:

1. **Apps:** Saves the date and time in which an external app was launched, and saves the full path of the loaded applications.
2. **Face Detection:** Each line has two columns, and represents a change in interaction status from attentive to distracted and vice-versa, the timestamp has the time in seconds since the Epoch and is saved on the left column, while on the right column, a Boolean value is stored, where 0 is the state in which the child looks away, and the 1 is the state in which the child looks back at the screen (attentive).
3. **Likes:** Saves the date and time of every *like/dislike* selection that the child made using the preference sharing pane presented along with each content item, and lists the full path of the chosen content item together with the type of selection (whether it was a *like* or *dislike*).
4. **Activities:** Saves the date and time in which a given activity was loaded, with the indication of the type of activity that was loaded and what section did the child visit during the session.
5. **Interaction:** Saves the date and time of nearly all human-computer interaction events generated by the child through the peripherals, namely the

[7] http://www.cis.upenn.edu/ bcpierce/unison

[8] https://www.dropbox.com/

[9] http://opencv.willowgarage.com/documentation/python

mouse events (move up, down, click, double click, and scroll events, the *x* and *y* coordinates of the pointer, and the left, right or middle button clicks), and the keyboard events (key up, down, and press, as well as the character code).

6. **Messages:** Saves the date and time in which the message was sent, together with the indication of the recipient peer, the number of characters in the message, and the number of pictures included in the message.

These logs are created to record every activity of the child while using the the platform, and will be used in our study to quantify the children's communicative and behavioral patterns. Preference sharing and Messages are social interaction activities, directly related with the communication skills, and as such it is important to track this activity individually, and measure it, to see how it evolves over time, as well as, and compare it to other children.

The Apps show us the type of application that children are using, which helpful to give some insight about their preferences and capacities. In our study, Apps are particularly related with story telling activities that the tutors configure in the platform, in particular with the Microsoft PhotoStory and with the Digital Library from the National Reading Plan[10]; both are directly related with communicative interaction patterns developed by the children.

Activities notes all activity of a child in the platform, which can be helpful to understand the children behaviour and focus of attention while using the platform (e.g. if the child spends some time watching photos, or videos and no time listening to music). Interaction is a log that allows us to get more detail about the behavior of the children in each activity (e.g. see if the child was doing anything during that period of time, or analyze how he/she navigates in the content).

The Face Detection log enables us to determine if the child was looking at the computer screen or not, thus providing further behavioral information which would not be accessible otherwise. This enables us to draw some conclusions about their attention levels while using the platform.

Combining all information available in the logs, we can have an objective and quantitative perception of the child's behaviour and perform group analysis in order to determine the effect of user-tuned content in the educational strategies and outcomes of children with ASD.

6 CONCLUSIONS

In this paper we present a novel methodology materialized by a platform and framework called myTroc@s.net, which emerged from the real-world needs of special education teachers. Our approach is focused on customization as a way of potentiating the capabilities of each child and improve their overall outcomes, by introducing the concept of profiles as a way of enabling the full customization of the platform for each child, together with a RAD/RAC framework designed to streamline the customization process for the tutors. Besides the children, we also focused our attention on the caregivers, since they are the ones that interact directly with the children, and prepare the individual profiles for the children.

The biggest novelty in this approach is the way we take advantage of several tools and known metaphors, and model their purpose, to create a rapid application that can fully answer two user types needs. In this case this simple application can be a simple content management for large amounts of files with a small learning curve, and also a learning tool for the children.

These goals helped us shape the application and create a framework that can easily aggregate more functionalities, and deliver a simple application using web technologies that allow an easy full manipulation. myTroc@s.net is an example of an application that could be easily created using the framework presented, by taking advantage of web technologies and the folder system to easily format an application, ease to use by all end users.

Our future work first will focus on improving the current application, so we can further improve the user experience. To help the tutors with content customization, it would be convenient to have the possibility of embedding YouTube videos in myTroc@s.net; most tutors said that if they had this functionality, they would have further customized the videos activity, since YouTube allows them to easily search and visualize video content. Still concerning the content customization, this process could be improved by creating a drag and drop page feature to manage files and folders across profiles through the myTroc@s.net interface, or by providing feedback about the effect that the changes to the File Management System (FMS) have on the profile configuration.

In order to allow a more sustainable and continued engagement between the users and the platform, and also to bring together parents and children during the time they are away from each other (e.g. at school), one next step will be to create a mobile version of the platform, to be used in smartphones and

[10] http://www.planonacionaldeleitura.gov.pt

tablet devices, enabling parents and tutors to interact with the children anywhere (and vice-versa).

Besides, given the natural curiosity of the child toward the touch-based interfaces, mobile devices can also facilitate the interaction between the child and the platform, further enhancing the motivation of the child towards its use. Due to our web-based software architecture, most of the platform can actually already be used in mobile devices, provided that the Python websocket server and Couch DB database are hosted in a remote server.

To facilitate the interaction between the child and the Troc@s-enabled devices, we are now preparing the framework to respond to physical objects. With this feature, children will be able to use their toys and other objects with which they are familiar with, to control and navigate in the platform. By introducing wireless RFID tags in each object, a reader in the computer will be able to detect a given object that the child is holding nearby, and change the behavior of the platform accordingly, enabling an easier navigation and increased interactivity.

ACKNOWLEDGEMENTS

This work has been partially funded by Fundação Portugal Telecom, ASUS Portugal, and by the Fundação para a Ciência e Tecnologia (FCT) under the grant SFRH/BD/65248/2009 and project PEst-OE/EEI/LA0021/2013.

REFERENCES

Berthoz, A., Vidal, P. P., and Graf, W., editors (1992). *The Head-Neck Sensory Motor System*. Oxford University Press, USA, 1 edition.

Bondy, A. and Frost, L. (2001). The Picture Exchange Communication System. *Behavior Modification*, 25(5):725–744.

Calais-Germain, B. (2007). *Anatomy of Movement (Revised Edition)*. Eastland Press, revised edition.

Emiliani, P. L., Burzagli, L., Como, A., Gabbanini, F., and Salminen, A. L. (2009). *Assistive Technology from Adapted Equipment to Inclusive Environments: AAATE 2009, Volume 25 Assistive Technology Research Series*. IOS Press, 1 edition.

Flanagan, D. (2011). *JavaScript: The Definitive Guide: Activate Your Web Pages (Definitive Guides)*. O'Reilly Media, 6th edition.

Helal, A., Mokhtari, M., and Abdulrazak, B. (2008). *The Engineering Handbook of Smart Technology for Aging, Disability and Independence*. Wiley-Interscience, 1 edition.

Lourenço, A., Silva, H. P., Carreiras, C., Priscila Alves, A., and Fred, A. L. (2013). A web-based platform for biosignal visualization and annotation. pages 1–28.

Lucas da Silva, M., Simões, C., Gonçalves, D., Guerreiro, T., Silva, H., and Botelho, F. (2011). TROCAS: Communication Skills Development in Children with Autism Spectrum Disorders via ICT Human-Computer Interaction INTERACT 2011. volume 6949 of *Lecture Notes in Computer Science*, chapter 103, pages 644–647. Springer Berlin / Heidelberg, Berlin, Heidelberg.

Putnam, C. and Chong, L. (2008). Software and technologies designed for people with autism: what do users want? In *Proceedings of the 10th ACM International Conference on Computers and Accessibility (ASSETS)*, pages 3–10, New York, NY, USA. ACM.

Wang, V., Salim, F., and Moskovits, P. (2013). *The Definitive Guide to HTML5 WebSocket*. Apress, 1 edition.