

# Learning from Learning Objects and their Repositories to Create Sustainable Educational App Environments

Carsten Ullrich, Ruimin Shen  
*Department of Computer Science*  
*Shanghai Jiao Tong University*  
*Shanghai, China*

*Email: ullrich\_c@sjtu.edu.cn, rmshen@sjtu.edu.cn*

Kerstin Borau  
*School of Continuing Education*  
*Shanghai Jiao Tong University*  
*Shanghai, China*  
*kerstin\_borau@onlinesjtu.com*

**Abstract**—The last years have seen an increased usage of apps, applications that are downloaded from a centralized repository and run in a particular environment. However, smaller app environments face the major challenge of attracting sufficient awareness and usage by third-party developers and users to become sustainable. This is a typical chicken-and-egg problem. Without apps, few people will be using the environment. Few users lead to limited incentive for app developers. We analyze work done in a very similar domain, namely learning object repositories (LORs), to learn about factors that inhibit participation in app environments. We also present an approach to overcome these problems, which aims at lowering the barriers of participation for app production as much as possible. The framework was partly implemented and is being applied at a higher education institution, where teachers and teaching assistants created over a hundred of apps.

**Keywords**-apps; markets; web search; metadata; education;

## I. MOTIVATION

The last years have seen an increasing popularity of apps, applications that are downloaded from a centralized repository (*market* or *store*) and run in a particular environment, such as OpenSocial containers like iGoogle, and device-specific environment such as iOS. The advantage over stand-alone software and web pages is that apps running within such an environment can access information such as location, and transfer data from one app to another. This allows for additional functionality not easily provided by other means. The European FP7 project ROLE investigates apps for education in the context of Personal Learning Environments (PLEs) whose functionality are composed by its users out of individual apps instead of being centrally defined by the learning institution. The apps are available in a dedicated educational store.

Smaller vendors and especially research-related app environments face the major challenge of attracting sufficient awareness and usage by third-party developers and users to become sustainable. This is a typical chicken-and-egg problem. Without apps, few people will be using the environment. Few users lead to limited incentive for app developers. Several factors contribute to this problem. The first one is the

difficulty of creating and publishing apps, which is almost impossible to do for the average user. The second factor is the problem of adequate description of apps, i. e., app metadata and its authoring. While general-purpose markets offer too few metadata, which makes finding appropriate apps difficult, the detailed metadata used in LORs overtaxes its potential users. The third factor is the difficulty of integrating existing resources inside the app environments. In education for instance, many web pages exist that train or teach specific topics and which would make sense to be available inside an environment in a way that makes use of its additional functionality.

In this paper, we present an approach that lowers the barriers of participation for app production significantly. This research was guided by the motivation to apply the AAA slogan, which states that “Anyone can say Anything about Any topic” [2], to apps and their environments. The slogan illustrates that one reason why the Web has become such a success is that fact that anyone can contribute, no central entity decides on what is published. Today’s markets are centralized and closed, which raises significant barriers to participation. The goal of our work is to lower the barriers as much as possible to enable contribution by anybody and unforeseen usage of apps and app metadata.

## II. LESSONS TAUGHT BY LEARNING OBJECTS

The similarities between app stores and LORs warrant a deeper investigation. Both store self-contained entities designed for interaction with a user and described in a specific way to make them findable. LORs have a long tradition in research in technology-enhanced learning and by taking into account the problems usage of learning objects and LORs faces, lessons can be learned regarding app stores.

A LOR uses metadata to describe its learning objects and exposes the metadata to users adding and searching for learning objects. The most frequently used standard is LOM, which describes learning objects through a number of almost 60 properties. Such detailed metadata is both difficult to author and difficult to use in search [3]. This is especially relevant regarding the importance of enabling

participation with as low barriers as possible as exemplified by Connexions, a repository with substantial growth [4]. Also, given that even LOM with its focus on education was the result of years-long effort and compromise, it is unrealistic to expect any agreement for apps, which come from a multitude of domains.

Besides metadata, the other major lesson for apps from learning objects concerns interoperability. Different standards exist that cover use cases ranging from integrating content (SCORM), to complete tools (IMS LTI). As major problem with all these solutions is that their complexity has massively hindered uptake. Independently, the W3C Web Applications Working Group is developing standards for a unified view on apps. At the time being these standards are in an early stage with limited support by environments, and in competition with OpenSocial.

### III. FRAMEWORK

This section describes the two main components of the proposed framework, namely the metadata that describes the apps and the execution wrappers that enable their usage in the corresponding environments. To fulfill the goal of creating a sustainable app store, the framework has to enable participation so that anybody can create and describe an app, not only developers or the original authors of a resource. Also, any metadata needs to be easy to understand, and employed mechanism easy to use while being extensible to new environments and domains.

#### A. Metadata

The purpose of the app metadata focused on in this article is to define a minimal viable semantic of an app that describes its possible usages. More specifically, the metadata describes the scope of the application so that it can be found by users based on what it does, describes, teaches, shows, using terms from the domain it is from. The proposed metadata partly builds on the W3C Widget Interfaces, but simplifies and adapts it.

The metadata is as follows: the property *id* is a URI that uniquely identifies the app. Its title is specified in the *name* property. The *category* represents the scope of the app on a very high, abstract level. The main information regarding the scope and topic of an app is encoded in the property *subject*. The values of the property *subject* are taken from Wikipedia and thus provide a precise and fine-grained description of the app. As an encyclopedia, Wikipedia aims at covering all areas of human knowledge. It is therefore safe to assume that topics that an app covers will be representable.

The next property is *instance*. The same app can have different instances that run in different environments (e.g., a Twitter app can be used in OpenSocial containers, but also on an Android phone). The sub-elements of *instance* describe these instances: the *type* specifies the environment of the instance, and the *href* attribute specifies the location

where the instance can be accessed. For instance, for an OpenSocial app *href* points to the XML gadget code. The persons who authored the apps are given in the property *creator*. Each instance can have a *description*. The technical metadata, which is specific to the environment (such as dimensions of an app, required permission) is stored the way required by the environment.

Since we also target the automatic conversion of existing web resources into apps, the metadata specifies the URL of the original *source* (an URL) and its *creator*.

For publishing the metadata, we employ RDF. RDFa can be used to encode the metadata in the original page, in case page authors want to describe the app functionality of their creations. In addition, any other party can author descriptions for existing web pages and publish the metadata on their own sites, store it in triple stores, or upload it to app stores which support such functionality.

This open creation and publishing of metadata following the example of the AAA slogan achieves two goals. First, apps are no longer only findable in centralized app stores, but can be found and indexed the same way any other content on the Web with specialized information describing its functionality. Second, everybody can author metadata that describes existing web pages as apps.

#### B. App Wrapper

The metadata allows describing apps and also existing web sites that offer functionality somebody deems to be useful in an app environment. Now, we look into how to transform these web pages into applications. Such a framework is relevant because today few apps are available compared to the pages that make up the web. If it were possible to use existing content and services in the platform, such that this content would support the functionality of the environment to provide added value, then with moderate effort a significant amount of resources were suddenly available.

The main component in this framework is the *wrapper*. It integrates an existing web resource into an environment and extends it with functionality. Each wrapper has two components: First, the *display* makes the content of the resource accessible to the environment so that it can be rendered. Second, the *extender* adds functionality to the embedded content. The precise functionality of the extender depends on the environment.

Our framework transforms web resources into apps by instantiating a *wrapper template*. A template is an app which requires the replacement of several placeholders with concrete values to become a functional app. The display and extender are part of this template. Since the templates are environment specific, each targeted environment requires its own wrapper template. From the view of the environment an instantiated wrapper works like any other app and its usage does not require any special treatment.

In our approach, the app producer uses a tool (the *generator*) to generate the app metadata and the app for a given environment. For each app, the generator requires the producer to input parts of the app metadata, namely the URL of the original web resource, the categories of the generated app, and the producer name. The generator then parses the resource located at the URL, extracts the resource title and author information (if available), and uses it to automatically generate values for the remaining app metadata. Afterward, the generator instantiates the app template to generate the complete app. Thus, the output of the process is the app and app metadata. Finally, the user (or the generator) uploads the app at the specified URL and publishes the metadata.

The advantages of the distributed generation approach are that the app and metadata are separated, and not just available within an app store. While the app can be uploaded at existing stores, it is still available outside of it, and its metadata can be used to make it available in other ways, e. g., indexed by app search engines. The disadvantage is that the producer has to upload the app and the metadata to a server.

#### IV. APPS GENERATION IN HIGHER EDUCATION

The above framework was partly implemented and evaluated in the School of Continuing Education (SOCE) at Shanghai Jiao Tong University (SJTU), an institution implementing blended learning. We investigated the usage of a PLE inside the LMS Moodle to augment the learning experience by providing additional activities. However, uptake of the PLE was slow, mostly due to the small amount of apps available in the beginning. We therefore implemented the described framework with the goal of quickly reaching a critical mass of widgets to attract teachers to the PLE. Implementing the framework required a concrete wrapper template. For SOCE, it consists of an OpenSocial app (an XML file) with placeholders. For the app generation process, the producer provides the URL of a web resource, its categories, and contact information to a production tool, which parses the resource, generates the metadata and instantiates the wrapper template to form the completed app. The metadata and app are then uploaded to a server (and also the ROLE widgetstore), and thereby available to everyone. Authoring happens through an admin-triggered batch process that converts several resources on one go, as well as through an authoring tool used by teachers.

After implementation, we generated about 420 apps in five courses Within two semesters. All these apps are reused web resources. It would have been impossible to manually author this amount in such limited time.

In order to have a quantitative measure to compare our approach to existing LORs, we performed a log-file analysis of the server the apps were uploaded to and counted the number of uploaded apps. The average growth rate in existing LORs is about 1 or 2 objects per day [5]. Our

logs show that in the two semesters (about eight months) after the framework was made available about 420 apps were produced. Albeit this amounts to a slower growth rate, in our case the production was done by a very limited amount of producers, namely three teachers and their assistants for five courses. Now that the framework has matured, we will make it available to additional lecturers.

#### V. CONCLUSION AND FUTURE WORK

The work described in this paper has presented a way how enable the reach of a critical mass of apps to gain self-sustaining traction. The presented approach is no alternative to dedicated apps fully using an environment's functionality, but it levels the playing ground by making the valuable resources available in the Web also available in the app environment. The approach builds on the AAA principle of the Web. Instead of creating close and monolithic markets, it allows anybody to produce apps and it makes the data about the app and the app itself available. A simple but effective semantic description of the app using the world knowledge represented in Wikipedia allows characterizing very precisely the functionality and scope of an app. The framework has been partly implemented and employed in a test-bed where teachers and teaching assistants, with little to no knowledge in programming and knowledge engineering produced more than 400 apps.

A number of issues need to be addressed by further work, among them the question of quality insurance, trust and rights management in such a decentralized system. A lot of research is currently invested in solving these questions for the Web as a decentralized system as a whole and we will investigate which methods are applicable in our context.

#### ACKNOWLEDGMENTS

This project was supported by FP7/2007-2013 under grant agreement no 231396 (ROLE project).

#### REFERENCES

- [1] S. Wilson, O. Liber, M. Johnson, P. Beauvoir, P. Sharples, and C. D. Milligan, "Personal learning environments: Challenging the dominant design of educational systems," in *Proceedings of the EC-TEL06 Workshops*, 2006, pp. 173–182.
- [2] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann Publishers Inc., 2008.
- [3] J. Najjar and E. Duval, "Actual use of learning objects and metadata - an empirical analysis," *TCDL Bulletin*, vol. 2, no. 2, 2006.
- [4] X. Ochoa, "Connexions: a social and successful anomaly among learning object repositories," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 1, pp. 11–22, 2010.
- [5] X. Ochoa and E. Duval, "Quantitative analysis of learning object repositories," *IEEE Transactions on Learning Technologies*, vol. 2, pp. 226–238, 2009.