An Architectural Design for Learning Analytics in Remote Education Environments

Completed Research

Sameer Verma
San Francisco State University
sverma@sfsu.edu

Andreas Gros
OLPC San Francisco
andigros@olpcsfl.org

Martin Dluhos
OLE Nepal
martin.dluhos@olenepal.org

Abstract
Improving education through technology has grown steadily. It is an ongoing challenge to implement education technologies in remote environments. It is increasingly difficult to gain feedback from such remote environments as these tend to be disconnected from the Internet. In this paper, we examine a remote education environment implemented through the One Laptop per Child (OLPC) project. We design an architecture and develop a learning analytics platform that allows various stakeholders to gain feedback from education environments across connected, intermittent, and disconnected Internet infrastructures.

Keywords
Learning Analytics, ICT4D, Education, Architecture, Infrastructure, OLPC

Introduction
Improving education through technology has grown steadily in the last 20 years (Bennett, Aguayo, & Field, 2016). We see several examples of this growth, in both the developing world and the developed world. One of the projects that garnered attention was the One Laptop per Child (OLPC) Project, originally proposed by Nicholas Negroponte, founder of Massachusetts Institute of Technology's (MIT) Media Lab (Negroponte, 2005). OLPC has stated as its mission - “To create educational opportunities for the world’s more disadvantaged children by providing each child with a rugged, low-cost, energy efficient laptop with content and software designed for collaborative, and self-empowered learning. The OLPC concept is an education project, rather than a laptop project” (OLPC, 2007). The catalyst for the self-empowered, peer-to-peer learning philosophy is the OLPC’s “$100 Laptop”, (called the XO), which provides a range of communications, collaborative and creative tools for expression (text music, video, graphics) that are the contemporary “toys” for learning (Negroponte, 2005). It is however difficult to know what happens with these laptops once they are deployed in the field. Are these being used at all? If so, when and how? Are the uses curricular or exploratory? Do they have challenges with electricity, Internet access, and educational content? This paper addresses some of these questions through the lens of a learning analytics framework.

Background
This paper is based on ongoing work with a series of OLPC projects in India, Nepal and Jamaica. Specifically of interest are the ongoing abilities of examining how the laptops are used in the field. Methodologies for collecting data leading to such insights vary by project. In most cases we observe that use profile of the technology is gleaned from interviews in the field (Lam, 2006). Other approaches involve the use of metadata from the computer (Siemens, 2013). OLPC did not specifically adhere to a prescribed methodology, but expected each project to generate its own approach to assess implementation and
Architectural Design for Learning Analytics in Remote Environments

sustainability (Rafa, 2010). To achieve a common set of goals, all projects ran on similar hardware and software. The software platform is of specific interest because logs from the operating system and the applications create interesting avenues for generating insight.

One of the key questions that drives the sustainability of these projects is: What do children do with the laptop once they receive it? In several projects of this nature, it is common to deliver the laptops, train the personnel, and leave the sustainability of the project with a local group (McNaughton, Bailey, Verma, & Muschette, 2015). However, apart from qualitative interviews, there isn’t a structured approach to extract any information from these laptops after they have been deployed in the field. Unless conducted appropriately, these interviews simply provide an anecdotal glimpse into the project.

On the other hand, if we have access to the metadata on the laptop, we may be able to gain insights into the laptop’s use. Furthermore, if we had access to the metadata from each laptop, and the ability to analyze the data collectively, we may gain insights into use patterns, scale and scope of laptop use. These questions drive the development of this architectural design and resulting prototypes.

Technology components

We examine the OLPC system as a collection of components designed to work together as one platform. These components are the laptop hardware itself, the operating system, the software platform and the applications.

Hardware

The OLPC XO laptop is a relatively unique hardware platform. Created for the OLPC project, this laptop is of robust design, consisting of a sunlight readable screen, solid state storage, and a ruggedized keyboard. The laptop features built-in Wi-Fi, camera, microphone, speakers, audio and USB ports. The laptop also contains a removable battery. The entire casing of the laptop folds into a robust case that can be carried with a handle.

Operating System

OLPC laptops use a custom version of Red Hat Linux operating system, which provides all the basic drivers for storage, processing, memory, network and keyboard/mouse. Some of the later versions also provide a touchscreen feature, although the system is fully operable without the touchscreen enabled.

Software Platform

The software platform is called Sugar. It is specifically designed for a constructivist learning experience. As per the Sugar Labs (Sugar Labs, 2018) website, “Sugar is a learning platform that reinvents how computers are used for education. Collaboration, reflection, and discovery are integrated directly into the user interface. Sugar promotes "studio thinking" and "reflective practice". Through Sugar’s clarity of design, children and teachers have the opportunity to use computers on their own terms. Students can reshape, reinvent, and reapply both software and content into powerful learning activities. Sugar’s focus on sharing, criticism, and exploration is grounded in the culture of free software (FLOSS).”

The Sugar learning environment is pre-installed on the OLPC laptop. Sugar uses specific HCI guidelines for all the applications (“Human Interface Guidelines - Sugar Labs” 2016).

Applications

A Sugar application is called “Activity” in Sugar terminology. Each Activity is written in either Python or JavaScript. All Activities written for Sugar adhere to a specification which allows the Activity to write its metadata to a datastore. Metadata created by each application is stored in the local datastore on the laptop (figure 1). The state of this datastore is made available to the user through an application called the Journal.
**Metadata**

The Journal allows for exploring the metadata (such as filename, timestamp, last modified, icons, collaboration, etc) in a journal-like format. This metadata can also be exported in various formats, such as XML, JSON, and CSV. Sugar also provides an application that can look at the aggregates of this metadata on each laptop. This aggregate provides a glimpse into how often the laptop is used, what kinds of applications are used on the laptop, how often collaboration occurs and other related insights. Note that this application is limited to a specific laptop only (figure 2) and do not provide insight across the classroom or school.

![Metadata stored in Journal](image)

**Figure 1: Metadata stored in Journal**
Several projects run automated backups of each laptop, primarily for the purpose of re-imaging corrupt drives. This backup happens automatically on the Wi-Fi network and stores compressed backups for each laptop on a server that runs on the local network. This server, called the XS school server (McNaughton, Bailey, Verma, & Muschette, 2015), typically runs in the Principal’s office or some other secure location on the grounds of the school where the laptops are deployed. It was not the original intention of the backup utility, but we were able to extract metadata from the backups from the XS server rather than trying to collect metadata from each laptop on the network.

**Learning Analytics framework**

The original goal of this effort was to analyze metadata obtained from a collection of laptops at a given deployment. In this case, it was a specific pilot study being conducted by the Center of Excellence at the University of the West Indies, Jamaica. The laptops were deployed at a local school. While we could examine this through the lens of a data analytics framework, given the specific nature of use of these laptops, we found it more appropriate to use a learning analytics framework. Cooper (2013) discusses a number of communities from which learning analytics draws techniques. These include statistics, business intelligence, web analytics, operational research, artificial intelligence, data mining, and social network analysis. For the purposes of our project, we only avail techniques from statistics and web analytics.

Chatti et al. (2012) suggest a framework that helps us define the various stages of the proposed Learning Analytics Platform (figure 3). This framework suggests that in a given system, we need to process data in four stages: Establish an activity that we would like to measure (typically within an application), collect data about the measurement, analyze collected data, and report the outcome to the stakeholder(s).
Architectural design

Using the learning analytics model from Chatti et al. (2014), we build the architectural design for our learning analytics platform following the four prescribed stages:

Measure

Various Sugar applications provide different variables available for measurement. The Sugar Journal records certain variables as metadata (figure 4). These are the name of the activity, a unique activity ID, the mime type, modification time, scope of collaboration, timestamp, default title, title set by user, and a user ID.

Collect

The metadata are stored on the XO laptop locally. The XS server backs up this datastore from each laptop seamlessly over the Wi-Fi network. As stated earlier, this backup system was originally designed to re-image corrupt laptop images, but in our case it can be used for purposes of extracting metadata. It is important to note that in this backup process, when we extract the metadata from these files, we anonymize and de-identify the data.

Analyze

There were several attempts to analyze metadata based on backups on the school server (McNaughton, Bailey, Verma, & Muschette, 2015). These were done in a variety of different ways, but none of these methods pointed to a streamlined effort to go beyond extracting the metadata. In many cases the metadata were analyzed in spreadsheets after the extraction was done separately using Python or Ruby. Initially we used Python and R to analyze the data, but we were able to find a more streamlined way to analyze the data in the database platform itself (details in the next section).

Figure 3: Learning Analytics Process. (Chatti et al. 2012)
Reporting of aggregates from these data sets was also based on different approaches in each project. In some cases these were pie charts and bar graphs generated in Ruby or JavaScript, while in other cases these were analyzed using Excel or R. We based our approach on these methods, but decided to extend the reporting as an ongoing approach (figure 5). In our approach, the reporting reflects the latest datasets as synchronized from the laptop to the school server. Additionally, the teachers and administrators do not have to run the analysis themselves. Instead, the analytics platform streamlines this process automatically.

Constraints

The nature of OLPC deployments bring with it some interesting constraints. In many cases the project deployments were remote and not connected to the Internet (Verma & Ryan, 2017). In other cases the connectivity was available, but was intermittent. In very few cases, access to the Internet was reliable and fast.

While assessing the need for analytics, we found that not only was the need for analytics pertinent for various stakeholders, it was also different in the way the data would have to be reported. Primarily, there is the need at the deployment (the edge of the network) by stakeholders such as the child, the parents, the teachers and the administrators. For instance, the parents would ask for reports on their child’s activities across the day, the week, and the academic term. However, the teacher would want similar reports, but aggregated across the entire class. We also found that administrators such as the Principal wanted reports, but these were more logistical in nature. They wanted to know how many laptops showed up on the Wi-Fi network everyday, to what extent did they use the Internet access, and to what extent were the laptops failing. The need for reports by the sponsors of a project or government institutions such as the Ministry of Education (the core of the network) was strong, but served very different purposes.

In order to provide reporting, both at the core and and the edge, we needed a distributed architecture to handle all the four stages of Learning Analytics and service the needs of the core and the edge with one system.

```python
#write header row
```

Figure 4: Metadata structure in Python
Learning Analytics Platform

Design parameters of the learning analytics platform were based on a set of requirements. We arrived at these requirements by interviewing various stakeholders (McNaughton, Bailey, Verma, & Muschette, 2015).

Mobile responsive

We found that there were several stakeholders who wanted access to various reports from the learning analytics platform. To keep things simple and accessible from OLPC laptops, regular computers, and mobile devices, we decided to build a mobile responsive interface using web technologies.

Intermittent connectivity

Managing intermittent connectivity was by far the most difficult challenge. While the extremes are easier to handle (i.e. either you have no connectivity, in which case there are no expectations to connect, or you have good connectivity in which case connecting is not a problem), the biggest problem was with intermittent connectivity. We needed something that could not only provide access intermittently, but also be reliable in its approach to synchronizing data from the school site at the edge to core locations on the cloud.

Database

We decided to address the problem of connectivity by building the database back end on a platform that allows for such intermittent connectivity. We chose the CouchDB database platform as opposed to some of the other technologies that rely on a more robust Internet connection (Anderson, Lehnardt, & Slater, 2010). Couch is an acronym for Cluster Of Unreliable Commodity Hardware. A CouchDB database is made up of entities called “documents”. Each document has its own data and schema. Document metadata has revision information, so that diffs can be merged when databases are disconnected. Conflicts are resolved in the
application as opposed to the database. The documents are organized through “views” which are lenses for aggregation, filtering and computation. We serialize the metadata from each XO laptop and store the resulting JSON string as a “document” in CouchDB.

**Aggregation**

We also wanted the analysis and aggregation of data to be done within the database platform. Given that CouchDB supports an aggregation feature, we were able to use the built-in features to not only analyze the data but also aggregate it into descriptive statistics.

**Visualization**

The next step was to report this aggregation as relevant to the various stakeholders. External viewers can be written in Python, JavaScript, PHP and other language platforms (Lawson, 2013). To do so, we chose to use the Highcharts JavaScript framework (Highcharts, 2014) that leverages HTML5 and CSS to provide visual representation of the aggregates by CouchDB (figure 6).

Using the architectural design described above, we were able to run an instance of CouchDB on the XS school server (at the edge of the network). This may be a location (say, a school) that has intermittent or stable Internet connectivity. Using replication features of CouchDB, we were able to incrementally replicate the database from the school location to a central location on the cloud. In this case we utilized cloud computing services of Cloudant, an IBM cloud service that runs CouchDB (Cloudant, 2014). The instances of CouchDB at each of the schools replicated to the central CouchDB at Cloudant as and when Internet connectivity became available. By using this additional feature of CouchDB replication, we were able to gather analytics across multiple schools and present those in comparative modes. This feature is of specific interest to government agencies.

**Conclusion**

The Learning Analytics Platform is able to take the measured data and collect it from the OLPC XO laptops, analyze it to provide aggregate data, and report the information through appropriate visualizations. While we are currently working with the second stage prototype, we have started to gather some feedback from the stakeholders.

Some notable characteristics are the seamless nature of measuring, collecting, analyzing and reporting, without any specific human intervention at the school site. All reporting is web-based, so it does not require any specific operating system or application other than a web browser. The eventual consistency feature of CouchDB helps in bridging the gap of poor Internet connectivity to remote locations where OLPC serves its constituents.

There are a handful of limitations with the current prototype. The visualization reflects the data as they are collected on the XS school server. We do not have robust mechanisms in place to adjust for drifting time clocks. We do have procedures to effectively anonymize and de-identify the data at its source, but the segmentation of reporting based on the stakeholder requires more work.

The software for this project is developed under the GNU GPL v3 license and is hosted on GitHub (Dluhos, 2015). We encourage contributors to extend the features of this platform across all the stages of measurement, collection, analysis and reporting. We would also like to extend the ability of this platform to systems that may not conform to the OLPC project and Sugar specifications.
REFERENCES


