Developing remote procedure calls for NetsBlox to facilitate climate change lessons in K-12 STEM education

Ismail Habibi, Gordon Stein, and Akos Ledeczi

The Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37212

KEYWORDS. Computer science education, climate change, block-based programming, K-12 education, STEM education

BRIEF. Climate change services were implemented into NetsBlox, as linking computer science to outside fields of science is an effective method of teaching programming.

ABSTRACT. Computer science is having a profound effect on the daily lives of billions of people around the globe. Due to this increased noteworthiness in recent years, it is exceedingly important to educate the next generation of students about computation. NetsBlox, a block based programming language built on Snap!, is a cloud-based environment which introduces distributed computing abstractions, allowing users to access external databases through various services using remote procedure calls (RPCs). In order to support this educational endeavor, four new services have been added to NetsBlox: Historical Weather Data, Carbon Dioxide Measurements from Mauna Loa, Hurricane Information, and Historical Ocean Temperature Data. Each of these new services are focused on climate change, as connecting computer science education with other aspects of life is beneficial for retaining computing concepts. After the main services, it was necessary to design and program the specific RPCs each service is comprised of, which would fetch various information from the corresponding database of each given service. The addition of these services makes it possible for users to create programs that both facilitate computer science learning and help demonstrate the effects of climate change.

INTRODUCTION.

Computer science has steadily gained in importance over the years, transforming many fields of work and our daily lives. Following from this increased need for knowledge of computer science concepts, it is necessary to make certain that there are appropriate resources to introduce and teach computational thinking to students starting from a young age. Block-based programming languages have typically been the optimal starting point for learning computing for novice students because using a set of predetermined blocks allows the students to focus on the more logical and computational side of computer science rather than memorizing the subtle syntax differences between various languages [1].

It has been demonstrated that linking computer science to real-world experiences is beneficial to learning the subject, as the they can play an important role in “grounding” the abstract programming concepts by illustrating how they can be used to model real-life phenomena [2]. The advantage of teaching programming as an interdisciplinary effort that involves topics other than computer science has been explored before in the context of physics [3]. Rather than being the focal point of the lessons, programming was used as a medium to convey the different aspects of physics that were a part of the study curriculum. In contrast, the current project aims to use external fields of science in order to enforce the retention of computer science knowledge, rather than the inverse.

To accomplish this goal, an educational environment called NetsBlox was used. NetsBlox is a block-based programming language and corresponding cloud-based environment that introduces distributed computing abstractions to the block-based programming paradigm [4]. This new functionality allows the creation of different NetsBlox services that communicate with third-party databases and online services, such as Google Maps, historical earthquake data from the US Geological Survey, astronomy imagery from the Sloan Digital Sky Server, the Open Movie DB, and many others, to retrieve various types of information. A NetsBlox service consists of a set of remote procedure calls (RPCs). Using RPCs, a user is able to send requests to an external web server, where the requests are interpreted and the desired information is returned and displayed within NetsBlox. The functionality afforded by NetsBlox opens up many possibilities in terms of what can be done within the environment and what kinds of programs can be created with the language.

Based on previous research showing that linking computer science education to other aspects of life is effective in retention of knowledge [2], climate change was decided on as the central theme of this project. Climate change is an important subject in the field of environmental science and has a personal connection with many students. Four different services were developed in this project: Historical Weather Data, Carbon Dioxide Measurements from Mauna Loa, Hurricane Information, and Historical Ocean Temperature Data. Each of these services encapsulate factors and effects of climate change and can all be used to develop programs within NetsBlox that illustrate the relationships among them. Two programs were created using the added services as part of this project. These pre-made programs can serve as a model for students to base their own future applications of off, fostering the effectiveness of NetsBlox-focused computer science education.

METHODS.

Using JavaScript, four services were implemented into NetsBlox for this project, each of which consist of multiple components. The heart of the service is what connects to the specified database, either an online web service or a data file, and includes multiple RPCs. These RPCs carry out specific functions within that particular service’s data. These RPCs also accept input parameters. The user must supply valid input values for the RPCs that require them, which limit the search through the data to get more specific results. Narrowing down the request through this method is effective to get the information that is useful to the user, such as displaying a table with only the desired values included or finding and displaying one value based on the other parameters passed into the RPC.

An integral component of creating the new services in NetsBlox was writing a unique file parser for each of the desired data sources. All of the files being used for these services were formatted in non-standard methods, making existing parsers incompatible with them. In this case, a parser is a computer program that reads a given file and returns only the information that the user requests. A custom parser was created specifically for the files of each of the services to read the desired columns of a file, returning a new file containing exclusively that information.

After creating the parser, the individual RPCs were developed. RPCs function as specialized calls to the data source that is being accessed by the parent service. Most of the RPCs developed for the Carbon Dioxide Measurements from Mauna Loa, Historical Ocean Data, and Hurricane Tracking Information services returned modified versions

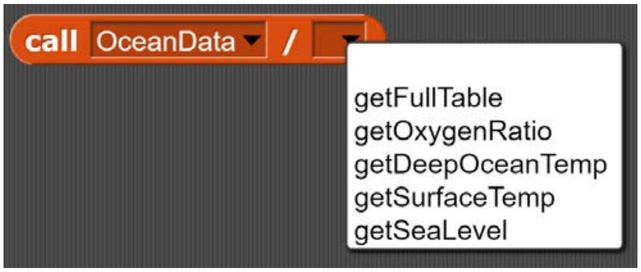

Figure 1. The available RPCs for the Ocean Data service.

of the initial parsed data file, which displayed different parts or columns of the table, as shown in Figure 1. However, some of the RPCs also returned specific number or string values if the correct parameters were entered, allowing for the user to enter two parameters and get back a single answer.

The customizable parser structure written for this project was modified for use for each of these three services. However, the parser was not functional with the final service that was to be added in this project, Historical Weather Data, as the database with the desired information was contained within an external application program interface, or API, on a different web server. Due to this, it was necessary to take an entirely different approach to creating this service.

In order to create the final service, it was necessary to write a program that would access the information within the National Oceanic and Atmospheric Administration (NOAA) Climate Data Online Web Service. NOAA collects information through the use of various stations scattered across the United States, all of which monitor weather conditions. With the way the NOAA database was currently set up, users would have to find a specific station before being able to see the weather data for a given location. In order to make using this service most convenient for the end user, it was decided that the best way for the user to access this information would be through RPCs that require only location parameters in the form of latitude and longitude coordinates, which would automatically find the nearest station and return the weather information, all wrapped up in one RPC within NetsBlox.

Working with the station-based NOAA database involved many components more complex than those of the first three services that were created. First, an RPC was created that displayed all of the possible data types available from the database. All of the NOAA stations that had ever been in use were then compiled into a JSON file, a filetype often used to represent data in modern computer programs. The next task was to create an RPC that accepted location parameters in the form of latitude and longitude coordinates and accessed the new file. The purpose of this RPC, called getNearestStation, was to find a station that was the nearest to the entered location so that the user could bypass having to manually enter a station ID themselves. After some testing, this RPC was completed and correctly found the nearest station to the coordinates entered into the parameter fields.

The purpose of the next RPC within the Historical Weather service, called getLocalWeatherInformation, was to take in a station ID as a parameter and return the information pertaining to that location. Unfortunately, this RPC proved to be too difficult to complete within the allotted development period.

RESULTS.

In this project, the developed services and the programs created using them serve as the primary results. As such, after the completion of the Carbon Dioxide, Ocean Temperatures, and Hurricane Tracker services, two example programs were created within NetsBlox to showcase the capabilities of the services. These demo applications function as a potential guide to new programmers using NetsBlox who wish to try out the various network-compatible services that the environment

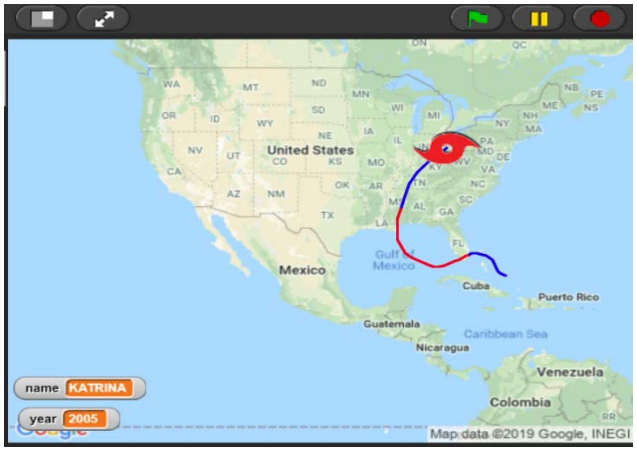
name KATRINA
year 2005

Figure 2. Program developed within NetsBlox that tracks any given hurricane using both the Google Maps and Hurricane Tracking services. Pictured is the path of Hurricane Katrina (2005).

offers. In order to be the most accessible for novice users, each of these custom programs was tailored specifically to a single service, as dividing the focus between multiple services at a time would likely be confusing to a new user trying to learn the functions of particular services and blocks of code.

The first program that was created within NetsBlox utilized the Hurricane Tracker RPC, illustrated in Figure 2. This application worked alongside the existing Google Maps RPC in order to map out the path of any given Pacific or Atlantic basin hurricane. Upon initializing the application, the user would be prompted to input a year between 1950 and 2018. Once a valid year was entered, a list of hurricanes that occurred within that year would be displayed. Using this list if necessary, the user would then input the name of the hurricane that they wish to plot. After an authentic hurricane was typed in, the sprite would travel across the displayed map in the same way that the original hurricane traveled across the area. Along the way, a line would be drawn behind the sprite as it progressed across the hurricane's track: blue for the time that the forming hurricane was still classified as a tropical storm, red for when the storm became a full-fledged hurricane. Additionally, the user can change how much of the map is displayed by zooming in and out with the plus (+) and minus (-) keys respectively and can pan across the map using the arrow keys.

The next application developed utilized the Carbon Dioxide Measurements service, which contains a database of interpolated and seasonal carbon dioxide concentrations from March 1958 to April 2019 on top of the Mauna Loa volcano. This service worked in tandem with NetsBlox's Chart service in order to create a graph of the change in CO2 concentrations in ppm over the years between 1958 and 2019, displayed in Figure 3. This program simplifies the process of graphing out either the interpolated or seasonal concentrations of carbon dioxide in Hawaii over the past 60 years.

Although a program created specifically for the Ocean Temperature Data service was not developed in this project, this service still holds much potential to showcase relationships between different climate change factors, such as generating a graph tracking the increase in sea levels over time.

DISCUSSION.

The selection of which services to implement into NetsBlox was informed by previous studies that concluded that linking computer science to other fields of science and facets of life is beneficial to the retention of integral programming concepts [2]. Climate change was chosen because it is a very important issue that permeates the daily

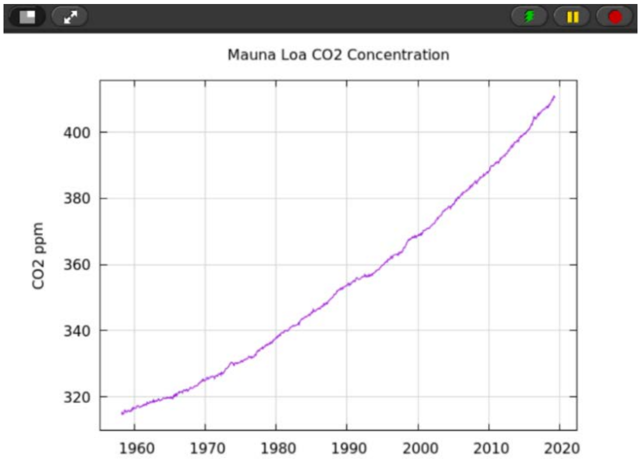

Figure 3. A graph charting the rise of carbon dioxide seasonal concentrations in Hawaii's Mauna Loa volcano from 1958 through 2019, created within NetsBlox using the Chart and Carbon Dioxide Measurements services.

lives of billions of people across the world and has effects that are experienced by every individual on a personal level to some degree.

These new services that have been developed for NetsBlox provide new ways for beginner programmers to learn core computer science concepts and logic while linking their programs to real world issues. The service CO2 Measurements would be very useful in tandem with the Historical Weather Data service for tracking both the carbon dioxide output of an area and the temperature at the same time. The Historical Ocean Temperature Data service could be used alongside the Hurricane Tracking Service to monitor changes in factors in a specific location of the ocean alongside the formation and paths of hurricanes near that location.

With the existence of these services within NetsBlox, students will have access to various climate-related databases that can be used when creating their own programs. The two existing programs explicated in the previous section are examples that illustrate the possible applications of the added services to the students. Using these premade programs as models, students will be capable of using the new services to create programs that may reveal relationships between different factors influencing climate.

While the successful implementation of the Hawaii Carbon Dioxide service, Hurricane Tracker, and Ocean Data service all came to fruition, the Historic Weather Data service has not been fully added due to study constraints. Due to this, there are only three functional services at the time of writing.

A potential enhancement that could improve NetsBlox's functionality is adding in more services that encompass more of the topic of climate change and the factors that affect it. These new services would address meteorological phenomena that influence climate patterns, such as wind patterns or ocean currents. Another future direction is to create templates that set up an outline of how the new services are used so that the users can more easily familiarize themselves with their functions. Following from this future direction comes a similar potential improvement on a larger scale: creating more full-fledged programs that illustrate all of the capabilities of each new service, such as the newly created Hurricane Tracker and Hawaii Carbon Dioxide Graphing programs.

Another future direction is fully completing the fourth service in this project, Historical Weather Data. As weather is a very important aspect of climate change, it is critical that this service be finished. Because the service already has the capability to seek out the nearest weather station to an inputted coordinate point, it should not be exceedingly difficult to implement a system of retrieving specific weather information from each station.

With all of the aforementioned applications, these services represent a large step forward in computer science education. By integrating the basic premise of teaching K-12 STEM students computer science with database functionality, these student users will be able to learn about the basics of programming along with how databases and information requests work, which is an ever-important skill to have in recent years with the growth of cloud-based data systems. This opens up the course for a more advanced curriculum while retaining the inherent simplicity and ease-of-use of block-based coding languages. Additionally, the theme of climate change that these services are based around also symbolizes another departure from traditional computer science education methods. Usually, tools for teaching computer science rely exclusively on programming-based examples to illustrate concepts. However, with the implementation of an entirely different field of science, students using NetsBlox and its various services experience hands-on how programming can be indispensable in many fields and can be used to model many miscellaneous phenomena in the natural world.

ACKNOWLEDGMENTS.

We would like to thank Dr. Menton Deweese for advising us throughout the project and providing indispensable feedback on all components of it. She was also very helpful in organizing the results of the work that was put into the lab throughout the summer.

REFERENCES.

1. C. Kelleher, R. Pausch, Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*. **37**, 83-137 (2005).
2. D. N. Perkins, R. Simmons, Patterns of misunderstanding: An integrative model for science, math, and programming. *Review of Educational Research*. **58**, 303-326 (1988).
3. N. Hutchins, G. Biswas, M. Maroti, A. Ledeczi, B. Broll, A design-based approach to a classroom-centered OELE. Paper presented at the International Conference on Artificial Intelligence in Education, London, United Kingdom, June 2018.
4. B. Broll, A. Ledeczi, H. Zare, D. N. Do, J. Sallai, P. Volgyesi, M. Maroti, L. Brown, C. Vanags, A visual programming environment for introducing distributed computing to secondary education. *Journal of Parallel and Distributed Computing*. **118**, 189-200 (2018).

Ismail Habibi is a student at Martin Luther King Jr. Magnet High School in Nashville, TN; he participated in the School for Science and Math at Vanderbilt University.