The Journal of Computing Sciences in Colleges

Papers of the 29th Annual CCSC Rocky Mountain Conference

October 9th-10th, 2020
Front Range Community College
Westminster, CO

Baochuan Lu, Editor
Southwest Baptist University

Bin Peng, Associate Editor
Park University

Pam Smallwood, Regional Editor
Regis University

Volume 36, Number 2 **October 2020**

The Journal of Computing Sciences in Colleges (ISSN 1937-4771 print, 1937-4763 digital) is published at least six times per year and constitutes the refereed papers of regional conferences sponsored by the Consortium for Computing Sciences in Colleges.

Table of Contents

The Consortium for Computing Sciences in Colleges Board of Directors

Following is a listing of the contact information for the members of the Board of Directors and the Officers of the Consortium for Computing Sciences in Colleges (along with the years of expiration of their terms), as well as members serving CCSC:

University, Dept of Computer Science, Greenville, SC 29613.

Bryan Dixon, Southwestern Representative (2023), (530)898-4864, bcdixon@csuchico.edu, Computer Science Department, California State University, Chico, Chico, CA 95929-0410.

Serving the CCSC: These members are serving in positions as indicated:

Bin "Crystal" Peng, Associate Editor, (816) 584-6884, crystal.peng@park.edu, Park University - Department of Computer Science and Information Systems, 8700 NW River Park Drive, Parkville, MO 64152.

Shereen Khoja, Comptroller, (503)352-2008, shereen@pacificu.edu, MSC 2615, Pacific University, Forest Grove, OR 97116.

Elizabeth Adams, National Partners Chair, adamses@jmu.edu, James Madison University, 11520 Lockhart Place, Silver Spring, MD 20902.

Megan Thomas, Membership System Administrator, (209)667-3584, mthomas@cs.csustan.edu, Dept. of Computer Science, CSU Stanislaus, One University Circle, Turlock, CA 95382.

Deborah Hwang, Webmaster, (812)488-2193, hwang@evansville.edu, Electrical Engr. & Computer Science, University of Evansville, 1800 Lincoln Ave., Evansville, IN 47722.

CCSC National Partners

The Consortium is very happy to have the following as National Partners. If you have the opportunity please thank them for their support of computing in teaching institutions. As National Partners they are invited to participate in our regional conferences. Visit with their representatives there.

Platinum Partner

Turingscraft
Google for Education
GitHub
NSF – National Science Foundation

Silver Partners

zyBooks

Bronze Partners

National Center for Women and Information Technology
Teradata
Mercury Learning and Information
Mercy College

Welcome to the 2020 CCSC Rocky Mountain Conference

Welcome to the 29th annual conference of the Rocky Mountain (RM) Region of the Consortium for Computing Sciences in Colleges. This year, due to COVID-19, we are running our first virtual conference. The CCSC RM region board members are grateful for the authors, presenters, speakers, attendees, and student participating in this year's conference.

This year we received 19 paper submissions on a variety of topics, of which 12 papers were accepted for presentation in the conference. Multiple reviewers, using a double-blind paper review process, reviewed all submitted papers for the conference. The review process resulted in an acceptance rate of 63%. In addition to the paper presentations, there will be three interesting tutorials/workshops out of five total tutorials/workshops submissions. We truly appreciate the time and effort put forth into the reviewing process by all the reviewers. A special thank you goes to co-Submission chair Karina Assister who worked with co-chair, Mohamed Lotfy. Without their dedicated effort, none of this would be possible.

The CCSC RM region board would like to thank our national partners: Turing's Craft, Google for Education, GitHub, the National Science Foundation (NSF), Codio, zyBooks, the National Center for Women Information Technology (NCWIT), TERADATA University Network, Mercury Learning and Information, Mercy College, and the Association for Computing Machinery in-cooperation with SIGCSE. We hope you enjoy the conference and take the opportunity to interact with your colleagues and leave both enthused and motivated. As you plan your scholarly work for the coming year, we invite you to submit a paper, workshop, tutorial, or panel for a future CCSC RM region conference, serve as a reviewer or on the CCSC RM region board. Please encourage your colleagues and students to participate in future CCSC RM region conferences.

Shawn Chiappetta and Matthew Rieck
University of Sioux Falls
Conference Chairs

2020 CCSC Rocky Mountain Conference Steering Committee

Regional Board — 2020 CCSC Rocky Mountain Region

Reviewers — 2020 CCSC Rocky Mountain Conference

Integrating Cloud-based File Storage and NoSQL Databases with Mobile App SQLite*

Conference Tutorial

Robert Sjodin and Mohamed Lotfy
Regis University
Denver, CO 80221
{rsjodin,mlotfy}@regis.edu

Mobile apps typically make use of user data that needs to be persisted across multiple executions of the app. The primary data source can be stored either locally on the device or remotely in the cloud. A combination of the two approaches can be used to back up the data to a secondary source. In this tutorial we will illustrate two operational scenarios:

- The primary data source located on the device, and a backup located in file storage in the cloud.
- The primary data source located in the cloud, with a transient backup stored on the device when remote connectivity is unavailable

The ACM/IEEE Computer Science Curricula 2013[1] added platform-based development (PBD) as a new knowledge area to the Computer Science body of knowledge. CS2013 recommended adding web application development and applying it over a wide range of ecosystems as part of the PBD knowledge area. CS2013 acknowledged the "increasing use of platform-specific programming environments, both at the introductory level and in upper-level electives"[1].

Tutorial Description

In part 1 of this tutorial we will provide a hands-on working example of a mobile app that backs up its user data to cloud-based file storage. In particular, in the tutorial we will illustrate and explain the following concepts[3]:

1. Configuration of the cloud provider:
    (a) creating an account and a location for database backups
    (b) establishing secure access to the remote server
2. Development of the app code and invocation of the cloud API:
    (a) uploading a local file
    (b) downloading a stored file
3. Implementing the following events:
    (a) the saving of the database to the cloud
    (b) the retrieval of the database from the cloud

In part 2 of this tutorial we will illustrate the integration of cloud-based NoSQL technology and mobile phone SQL databases. The remote NoSQL database is used as the primary source for storing the user's application data. The mobile relational database is used as a local, embedded cache when connectivity to the cloud provider is not available. The cache is used to update the cloud NoSQL database when online connectivity is reestablished. The employed cloud service is Google Firebase and the mobile application framework is Android, but the concepts can be applied to any cloud provider and mobile framework. An end-to-end working example is given to illustrate the concepts. In particular, in the tutorial we will illustrate and explain the following concepts[2]:

1. Configuration of the cloud provider:
    (a) creating an account and a location for database backups
    (b) establishing secure access to the remote server
2. Development of the app code and invocation of the cloud API:
    (a) persisting user information to the cloud NoSQL database when remote connectivity is enabled
    (b) persisting user information to the local SQL database when remote connectivity is disabled
    (c) synchronizing the NoSQL and SQL databases when connectivity is reestablished[2]

Tutorial program

Step-by-step implementation and testing of a mobile app that backs up its user data to cloud-based file storage (using Google Firebase as the cloud provider).

Expected outcomes

Attendees will exit the tutorial with a working example of a mobile application that backs up its user data to cloud-based file storage.

Target audience

Any faculty who desires to incorporate cloud-based file storage and NoSQL databases with mobile app SQLite in their courses.

Prerequisites

Attendees should be familiar with OO programming; e.g. Java, C++, Python, etc. It is highly recommended that attendees bring their own laptops with Android Studio installed. No prior experience with Android Studio is required.

References

[1] ACM/IEEE. Computer science curricula 2013 - curriculum guidelines for undergraduate degree programs in computer science. *The Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) IEEE Computer Society*, 2013.

[2] Zach McCormick and Douglas C. Schmidt. Data synchronization patterns in mobile application design. In *Data Synchronization Patterns in Mobile Application Design, proceedings of the Pattern Languages of Programs (PLoP) 2012 conference.*, pages 19–21, 2012.

[3] Pratap P. Nayadkar and Banu L Parne. A survey on different backup and restore techniques used in mobile devices. *International Journal of Computer Science and Information Technologies*, 5(6), 2014.

What I Have Learned About Teaching from Scoring AP Computer Science Exams*

Conference Tutorial

Victoria Eisele
Front Range Community College
Fort Collins, CO 80526
`victoria.eisele@frontrange.edu`

I have been a reader of AP exams for many years. I have seen thousands of exams. Over the years I have come up with some ways that I have modified my teaching to meet student's needs. I learn something new every year. Some I have already implemented and others I will be working on in the next few years. Nothing on this is formal research but observations.

Many people teach what they were taught or just what the book has without thinking about why or what is best for the students. Yes, many people are successful this way but could we help more students be successful.

The topics will be

- Placement of braces
- Vocabulary
- Real numbers
- Multiple Returns
- Setting up a return value
- Integer division
- Use of break in a loop
- Using += etc
- Variable names
- Descending loops not incrementing
- Do while loops
- Enhanced for loops

Discussion Description

In this tutorial we will explore different ways people teach concepts and what seem to lead so more or less understanding. I would like this to be open discussion of why we teach the concept the way we do.

Expected outcomes

Attendees will leave this session thinking about what they are teaching beginners and why.

Target audience

Any faculty who teaches introductory programming classes that wants to look at what they are doing and see if anything could be done to make it better.

Prerequisites

None.

Integrating Mobile Relational Databases and Cloud-based File Storage

Robert Sjodin, Robert Mason and Mohamed Lotfy
Regis University
Denver, CO 80221
`{rsjodin,rmason,mlotfy}@regis.edu`

Abstract

In this paper we discuss the use of cloud-based file storage for backing up and restoring a mobile application database. Google Firebase is employed as the cloud service, SQLite is leveraged on the mobile device and the mobile application framework is Android. Note that the concepts in this paper can be applied to any cloud provider and mobile framework. An end-to-end working example is provided to illustrate the technique and concepts.

1 Introduction

The ACM/IEEE Information Technology 2017 Final Curriculum Report (IT2017) included web and mobile app development in the Integrated Systems Technology (ITE-IST), System Paradigms (ITE-SPA), and Web and Mobile Systems (ITE-WMS) essential IT domains[2]. In addition, the ACM/IEEE Computer Science Curricula 2013 (CS2013) added platform-based development (PBD) as a new knowledge area to the Computer Science body of knowledge. Included in the CS2013 recommendations was mobile application development "both at the introductory level and in upper-level electives" [1]. To address CS2013 and IT2017 recommendations, mobile application development courses, both required and elective, were added to computer science and information technology programs to address the growing demand of these skills in the graduates of these programs.

Teaching mobile application development concepts in undergraduate computer science and IT courses needs to introduce students to the storage and retrieval of the user data through the use of cloud computing in the context of Database as a Service (DaaS). [5] provided a survey of mobile phone backup and restore techniques. The system level options available on various devices, including iPhone, Palm, Windows Mobile, Blackberry, Symbian, and Android were highlighted[5]. [7] provided an overview of Android backup and restore capabilities. The notions of full backups, incremental backups, differential backups, and mirror backups using Sprit, Titanium, and My Backup Pro were explained[7]. [6] provided an instructional example for backing up and restoring RLE compressed data on an Android platform. [8] provided an end-to-end, working example that illustrated the use of cloud computing in the context of a mobile application development undergraduate course. The provided example involved an Android mobile app that provided the front-end user interface and a cloud NoSQL real-time database, hosted in Google Firebase[8]. In addition, Android's official online site focuses on device upgrades and app re-installation[3].

Mobile phones are being used increasingly more to store user data that is deemed important, if not critical, to end users. The application data is typically stored in an SQLite database that's embedded in the device[9]. Various techniques have been developed to back up the data; invariably, the solution involves a system-level approach that targets the user's identify, their settings, and their suite of installed apps. Mobile apps typically make use of user data that needs to be persisted across multiple executions of the app. The data is often stored on the user's handheld device, in a database embedded in the app. If the app is uninstalled, or if the device is destroyed, the data is lost. To resolve this problem, the data can be backed up on an external remote server, as shown in Figure 1.

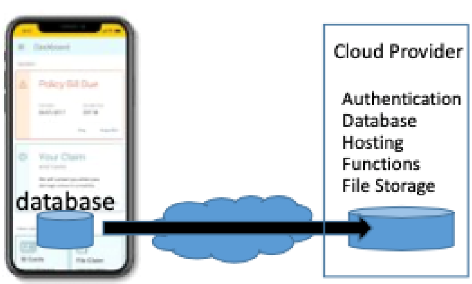


Figure 1: User Application Data Backed Up in the Cloud

If the data needs to be restored, it can be downloaded and reinstalled on the device. The concept is simple enough, but details need to be worked out, including:

- Configuring the cloud provider
  - creating an account, and a location for database backups
  - establishing secure access to the remote server

- Developing the application code; invocation of the cloud API for
  - uploading a local file
  - downloading a stored file

- Identifying, choosing and implementing the events that initiate
  - the saving of the database to the cloud
  - the retrieval of the database from the cloud

The above concepts are taught at Regis University in an undergraduate course entitled "Mobile and Enterprise Computing" . The course has eight topics, with the last three dedicated to Google Firebase. The concept of uploading the app's database to Firebase is addressed in the last topic and is explained below in the following sections. We begin with the configuration of the cloud provider –Google Firebase[4].

2 Google Firebase - File Storage

All cloud providers support the notion of storing and retrieving files. This capability can be used to back up a database that is embedded in a mobile app. The console for Google Firebase File Storage is shown in Figure 2 below. It can be accessed at www.firebase.google.com.

In Figure 2, screenshot of the Firebase Storage console, the "red" annotations highlight the following:

1. The "Files" tab is selected.
2. The "+" icon is used to create folders/directories for storing files.
3. The folder "database" is created to store database backups.

The security access rules are specified under the "Rules" tab. Initially, we set the security settings to allow public read/write access. Once the upload-/download functionality is completely tested, we change the security settings to restrict access to only authorized users. The initial public security rules are displayed in Figure 3.

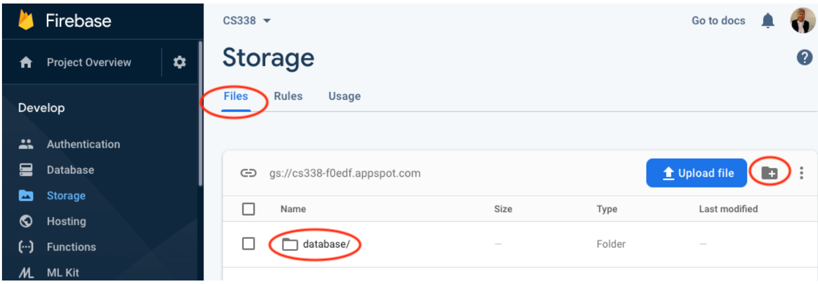


Figure 2: Google Firebase Console for File Storage

```
rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write;
    }
  }
}
```

Figure 3: The Firebase Security Rules for File Storage

3 The Code –Exporting and Importing the DB

The Android app uploads the database content via the Firebase API. The required libraries are linked in the app via the Gradle configuration file, as shown in Figure 4.

```
implementation 'com.google.firebase:firebase-core:17.4.0'
implementation 'com.google.firebase:firebase-storage:19.1.1'
```

Figure 4: Firebase libraries identified in app's Gradle configuration file

Once the Firebase libraries are added to the Android project, code is developed to upload and download the database. Two methods are created:

- exportSQLiteDatabase(String backupname)
  - exports the app's SQLite database to a file stored in Google Firebase

- importSQLiteDatabase(String backupname)
  - imports the app's SQL database from a file stored in Google Firebase

The details of the two methods are shown below in Figures 5 and 6.

```
public void exportDB(String backupname) {
    Log.i(TAG, msg: "*** exportSQLiteDatabase");
    try {
        String dbPath = "/data/data/com.example.my3612app/databases/contacts.db";
        Uri file = Uri.fromFile(new File(dbPath));
        FirebaseStorage firebaseStorage = FirebaseStorage.getInstance();
        StorageReference storageRef = firebaseStorage.getReference();
        StorageReference ref = storageRef.child("database/" + backupname);
        ref.putFile(file).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                Log.i(TAG, msg: "*** onSuccessListener");
            }
        }).addOnFailureListener((e) → {
                Log.i(TAG, msg: "*** onFailureListener");
        });
        Log.i(TAG, msg: "exiting exportSQLiteDatabase");
    } catch (Exception e) {
        Log.i(TAG, msg: "EXCEPTION: " + e.getMessage());
    }
}
```

Figure 5: Code to Export the SQLite Database to Firebase

The above methods can be executed from numerous contexts, including the following scenarios:

- manually by the user, from an application menu selection,
- after each update to the local database,
- with each termination of the app by the user,
- periodically, as dictated by the user in an application setting.

The above scenarios are not mutually exclusive; one or more can be implemented in a given application. Specifically, all of them can be implemented in the same application. As an example, we will focus on the first scenario, when the user triggers the event through a menu selection of the app.

4 Execution Scenario - menu selections

The Android app provides two UI menu selections: "Export Database" and "Import Database" ; the purpose of each selection is straightforward:

- "Export Database" triggers the execution of method exportDB
- "Import Database" triggers the execution of method importDB

The code for "Export Database" does the following:

```
public void importDB(String backupname) {
    Log.i(TAG, msg: "*** importSQLiteDatabase");
    try {
        String dbPath = "/data/data/com.example.my3612app/databases/contacts.db";
        File file = new File(dbPath);
        FirebaseStorage firebaseStorage = FirebaseStorage.getInstance();
        StorageReference storageRef = firebaseStorage.getReference();
        StorageReference ref = storageRef.child("database/" + backupname);
        ref.getFile(file).addOnSuccessListener(
                new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
                    @Override
                    public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {
                        try {
                            Log.i(TAG, msg: "*** onSuccess COMPLETED");
                        } catch (Exception e) {
                            Log.i(TAG, msg: "*** onSuccess EXCEPTION: " + e.getMessage());
                        }
                    }
                }
        ).addOnFailureListener((e) -> {
                Log.i(TAG, msg: "*** onFailure EXCEPTION: " + e.getMessage());
        });
    } catch (Exception e) {
        Log.i(TAG, msg: "EXCEPTION: " + e.getMessage());
    }
}
```

Figure 6: Code to Import the SQLite Database from Firebase

- Queries the current date and time (to be used as the file name)
- Invokes the method exportDB (with the date and time as the filename)
- Upon success, stores the filename for future use
- Notifies the user of success/failure.

And the code for "Import Database" does the following:

- Queries the filename (date and time) from the user
- Invokes the method importDB (with the date and time as the filename)
- Notifies the user of success/failure

The UI sequence for exporting and importing the database is illustrated in Figure 7 below.

5 Conclusions

Cloud file storage is a logical and straightforward solution for backing up mobile application data. In this paper we illustrated the backup and restoration of an SQLite database using Google Firebase from an Android perspective, but the concept can be applied to any mobile framework and cloud provider. The

Figure 7: Export and Import of SQLite Database using Firebase

capability is internal to the app and does not require any 3rd party software. The concepts presented in this paper have been taught for two years—six online and two face-to-face courses—in an undergraduate course entitled "Mobile and Enterprise Computing". The cloud file storage concepts have been well received and executed by the students, in both classroom and online settings. The students gained hands-on experience in developing mobile applications using cloud-based storage and were able to articulate the advantages/disadvantages of cloud versus local storage.

References

[1] ACM/IEEE. Computer science curricula 2013 - Curriculum guidelines for undergraduate degree programs in computer science. *The Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) IEEE Computer Society*, 2013.

[2] ACM/IEEE. Information technology curricula 2017 - Curriculum guidelines for baccalaureate degree programs in information technology. *Task Group on Information Technology Curricula, Association for Computing Machinery (ACM) IEEE Computer Society*, 2017.

[3] Android Developers. Data backup overview, 2019. `https://developer.android.com/guide/topics/data/backup`.

[4] Firebase. Firebase realtime database, 2020. `https://firebase.google.com/products/realtime-database`.

[5] Pratap P. Nayadkar and Banu L Parne. A survey on different backup and restore techniques used in mobile devices. *International Journal of Computer Science and Information Technologies*, 5(6), 2014.

[6] Medhavi S. Shriwas, Neetesh Gupta, and Amit Sinhal. Efficient method for backup and restore data in Android. In *2013 International Conference on Communication Systems and Network Technologies*, pages 693–697. IEEE, 2013.

[7] Medhavi S. Shriwas and Amit Sinhal. Backup and restore data in Android. *International Journal of Pure and Applied Research in Engineering and Technology*, 2013.

[8] Robert Sjodin and Mohamed Lotfy. Integrating cloud computing into the curriculum. *Journal of Computing Sciences in Colleges*, 35(2):29–38, 2019.

[9] SQLite. About SQLite, 2020. `http://sqlite.com/about.html`.

Integrating Cloud-based NoSQL Technology and Mobile Phone Relational Databases*

Robert Sjodin, Robert Mason and Mohamed Lotfy
Regis University
Denver, CO 80221
{rsjodin,rmason,mlotfy}@regis.edu

Abstract

In this paper we discuss the integration of cloud-based NoSQL technology and mobile phone relational databases. The remote NoSQL database is used as the primary source for storing the user's application data. The mobile relational database is used as a local, embedded cache when connectivity to the cloud provider is not available. The cache is used to update the cloud NoSQL database when online connectivity is reestablished. The employed cloud service is Google Firebase and the mobile application framework is Android, but the concept can be applied to any cloud provider and mobile framework. An end-to-end working example is given to illustrate the technique and concepts.

1 Introduction

Teaching mobile application development concepts in undergraduate computer science and IT courses need to introduce students to the storage and retrieval of the mobile application user data through the use of cloud computing in the context of Database as a Service (DaaS). Both the ACM/IEEE Computer Science Curricula 2013 (CS2013) and the ACM/IEEE Information Technology 2017 Final Curriculum Report (IT2017) included web and mobile app development as required knowledge areas. CS2013 added platform-based development

(PBD) as a new knowledge area to the Computer Science body of knowledge. Included in the CS2013 recommendations was mobile application development "both at the introductory level and in upper-level electives" [1]. IT2017 deemed the Integrated Systems Technology (ITE-IST), System Paradigms (ITE-SPA), and Web and Mobile Systems (ITE-WMS) essential IT domains[2]. To address CS2013 and IT2017 recommendations, mobile application development courses, both required and elective, were added to computer science and information technology programs to address the growing demand of these skills in the graduates of these programs.

Mobile phone application data can be stored remotely in the cloud or locally on the device. The advantage of remote data storage is its large data capacity and its inherent backup capability; its disadvantage is occasional service disruption. Conversely, the advantage of local device storage is continuous data availability; its disadvantage is its limited storage capacity and no inherent backup capability. The two scenarios are depicted in Figure 1.

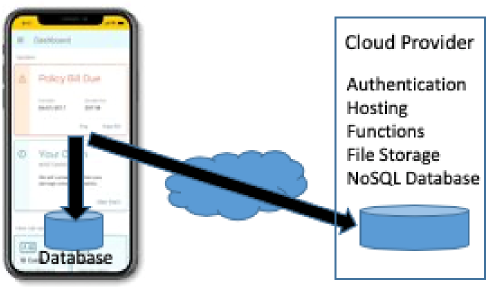


Figure 1: Local and Cloud Data Storage

The above two approaches to data storage are not mutually exclusive; they can be used together. Consider the following scenario:

- The remote, cloud storage can be used as the primary data source.
- The local database can be used as a data cache, to enable functionality when remote connectivity is unavailable. Then when connectivity is reestablished, the cache can be used to update the cloud database.

The main disadvantage of remote data storage is the occasional interruption of service when connectivity is lost. When this happens, the remote data is unavailable until the connection is restored. The user can continue to use the app, but they cannot access the remote server. In this case, the local SQLite database can be used to cache any changes the user makes with the app[8]. Then, when connectivity is restored, the app can update the cloud NoSQL database using the cache embedded in the device[6]. The following sections of this paper expand on the above scenario. In particular,

- The NoSQL cloud database will be used as the primary database.
- The embedded SQLite database will be used as a cache when the cloud NoSQL database is unavailable.

A literature review of mobile application data storage is provided below.

- An overview of mobile app data storage was provided by [5]. Three types of apps regarding data storage were identified: online, offline, and synchronized. [5] characterization of a synchronized app describes the scenario discussed in this paper.
- The architectural patterns associated with data synchronization were addressed by [4], including the following patterns: asynchronous, synchronous, partial storage, complete storage, full transfer, timestamp, and mathematical transfer. Using [4] pattern names, the technique discussed in this paper can be classified as an asynchronous, complete storage, mathematical transfer.
- [7] provided a case study of an application that uses the Firebase NoSQL database to store data from a mobile app. The application provided Create, Read, Update, and Delete (CRUD) functionality.

We begin with the configuration of the Google Firebase[3] Realtime Database, a NoSQL database.

2 Cloud and Application Configuration

The application uses the Firebase NoSQL Realtime database as its primary data source for storing user notes. The security rules are initially set to allow read/write public access. Then once the app is working with public access, the rules will be restricted to authenticated users. The public security rules are displayed below in Figure 2.

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

Figure 2: The Firebase Realtime Database Security Rules

The Android app will interface with the Firebase Realtime Database via the Firebase API. The required libraries are specified in the project's Gradle file, as displayed in Figure 3 below.

```
implementation 'com.google.firebase:firebase-core:17.4.0'
implementation 'com.google.firebase:firebase-database:19.3.0'
```

Figure 3: Firebase Libraries Identified in app's Gradle Configuration File

With the cloud and app configured, we proceed with development of the application code.

3 Cloud NoSQL and Mobile SQLite Databases

The mobile app allows the user to create, retrieve, update and delete user notes with the Firebase NoSQL database when Internet connectivity is enabled. To support offline functionality, we use an SQLite table that replicates the attributes of the NoSQL database, plus the following additional properties:

- id –an auto incremented primary key for each transaction
- transtype –the transaction type: create, update, or delete
- timestamp –the time at which the transaction was submitted.

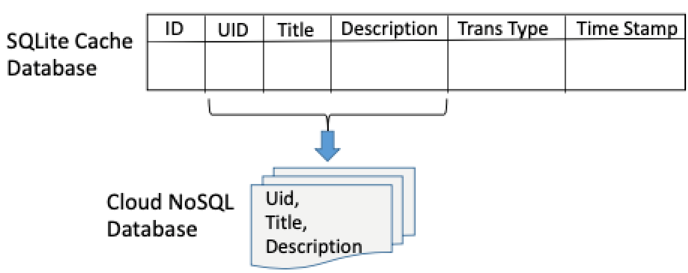


Figure 4: The Relationship of the Cache SQLite and Cloud NoSQL Databases

With the databases defined, we can now turn our attention to the system-level events and application code that interfaces with the above databases.

4 The App Code –Remote and Local Storage

The primary data source of the mobile app is a NoSQL database hosted by Google Firebase. Access to the database requires Internet connectivity. In particular,

- When Internet connectivity is present, the app will use the NoSQL database
- When Internet connectivity is absent, the app will use the embedded SQLite database.

Furthermore, when the app transitions from being disconnected to connected, the local cached transactions will be uploaded to the NoSQL database. The events associated with these scenarios are illustrated in Figure 5.

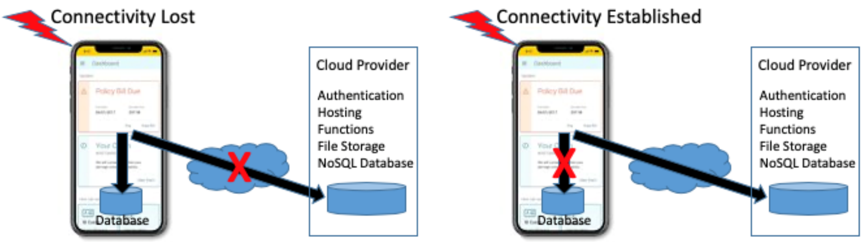


Figure 5: Connection and Disconnection Events

The key to implementing the scenarios depicted in Figure 4 is the detection of two events:

- Connectivity Lost
- Connectivity Reestablished

The above events are programmatically available on an Android device via the Connectivity Manager API (application programming interface). To respond to these events, we create an Android Broadcast Receiver, and then register it with the host OS. The code for the Broadcast Receiver and its registration are displayed below in Figures 6 and 7.

When network connectivity is disabled, all subsequent transactions are cached in the local SQLite database. Then, when network connectivity is reestablished, the cached data that includes all subsequent transactions, is uploaded to the cloud NoSQL database.

5 Conclusions

In this paper we illustrated the integration of cloud NoSQL databases and embedded mobile SQLite databases. The cloud NoSQL database is used as the primary data source for the application; the embedded SQLite database is used

```java
public static class MyNetworkChangeReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(final Context context, final Intent intent) {
        Log.i(TAG, msg: "*** MyNetworkChangeReceiver ENTER ***");
        boolean isConnected = isConnectedToNetwork(context);
        Toast.makeText(context, text: "*** Connection: " + isConnected + " ***", Toast.LENGTH_LONG).show();
        if (isConnected) {
            useCloudDb();
        } else {
            useCacheDB();
        }
        Log.i(TAG, msg: "*** MyNetworkChangeReceiver EXIT ***");
    }

    public boolean isConnectedToNetwork(Context context) {
        ConnectivityManager connectivityManager =
                (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY_SERVICE);
        boolean isConnected = false;
        if (connectivityManager != null) {
            NetworkInfo activeNetwork = connectivityManager.getActiveNetworkInfo();
            isConnected = (activeNetwork != null) && (activeNetwork.isConnectedOrConnecting());
        }
        return isConnected;
    }
}
```

Figure 6: Network Change Broadcast Receiver

```java
mynbr = new MyNetworkChangeReceiver();
intentFilter = new IntentFilter();
intentFilter.addAction("android.net.conn.CONNECTIVITY_CHANGE");
intentFilter.addAction("android.net.wifi.WIFI_STATE_CHANGED");
registerReceiver(mynbr, intentFilter);
```

Figure 7: Registration of Network Change Broadcast Receiver

as a cache for storing application data when cellular connectivity is unavailable, and for updating the NoSQL database when connectivity is reestablished. To test the app, the connection/disconnection events were initiated from the device's Settings panel. The events were monitored in the running app via the framework's "Connection Manager" . Disconnecting and reconnecting the app resulted in the NoSQL database being successful updated with data from the local cache. A workshop at the RMCCRM conference will provide hands-on experience in implementing this scenario.

References

[1] ACM/IEEE. Computer science curricula 2013 - Curriculum guidelines for undergraduate degree programs in computer science. *The Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) IEEE Computer Society*, 2013.

[2] ACM/IEEE. Information technology curricula 2017 - Curriculum guidelines for baccalaureate degree programs in information technology. *Task Group on Information Technology Curricula, Association for Computing Machinery (ACM) IEEE Computer Society*, 2017.

[3] Firebase. Firebase realtime database, 2020. `https://firebase.google.com/products/realtime-database`.

[4] Zach McCormick and Douglas C. Schmidt. Data synchronization patterns in mobile application design. In *Data Synchronization Patterns in Mobile Application Design, proceedings of the Pattern Languages of Programs (PLoP) 2012 conference.*, pages 19–21, 2012.

[5] Robert C. Nickerson and Francois B. Mourato-Dussault. Selecting a stored data approach for mobile apps. *Journal of Theoretical and Applied Electronic Commerce Research*, 11(3).

[6] NoSQL. What is NoSQL?, 2020. `https://aws.amazon.com/nosql/`.

[7] Robert Sjodin and Mohamed Lotfy. Integrating cloud computing into the curriculum. *Journal of Computing Sciences in Colleges*, 35(2):29–38, 2019.

[8] SQLite. About SQLite, 2020. `http://sqlite.com/about.html`.

Results of using a Multi-Programming Language Approach to Decrease Drop-Fail Rates in CS1*

Ed Lindoo
Regis University
Denver, CO 80221
elindoo@regis.edu

Abstract

In late 2018, we presented a paper [7] pointing out the high drop-out and failure rates (D/F) in our two introductory programming courses within the CIS department at Regis University. The study performed over a three-year period found D/F rates in excess of 60%. High drop and failure rates are, of course, a big conňcern for institutions, instructors and students because there are a lot of ramifications for each of these. One of the things we pointed out through our research is that even though languages such as Java, C and C++ are popular, there has been much debate about the suitability of these languages for education, especially when introducing programming to novices.

In the previous paper we presented the idea of going back to the basics, BASIC programming that is, but not with traditional programs like QBASIC or MS-BASIC. Instead we introduced a product called Xojo (pronounced Zo-Jo). Upon implementing this, we decided to take it a step further. Following the work of Hong et. al. [6] a multi-programming environment was implemented in which we teach BASIC and Java at the same time, comparing and contrasting the two languages. Since introducing this in the fall of 2018, students went from a 38% pass rate to an 76% pass rate in this multi-programming course, where Xojo and Java are taught together. While this is a great success, it should be noted that those who went on to the follow-up Java course passed with a 75% success

rate, as opposed to 47% previously. The audience for this paper includes computing faculty that are planning, designing, developing, revising or implementing a new course that introduces programming concepts to novice programmers.

1 Introduction

The challenge at our university, specifically within the College of Computer Information Sciences (CC&IS), was to reduce the high drop-fail (D/F) rates that we have seen over the past three years. Figure 1 shows these rates for CIS and CS majors for the two-and-a-half-year period starting in August 2015, ending with the January 2018 term.

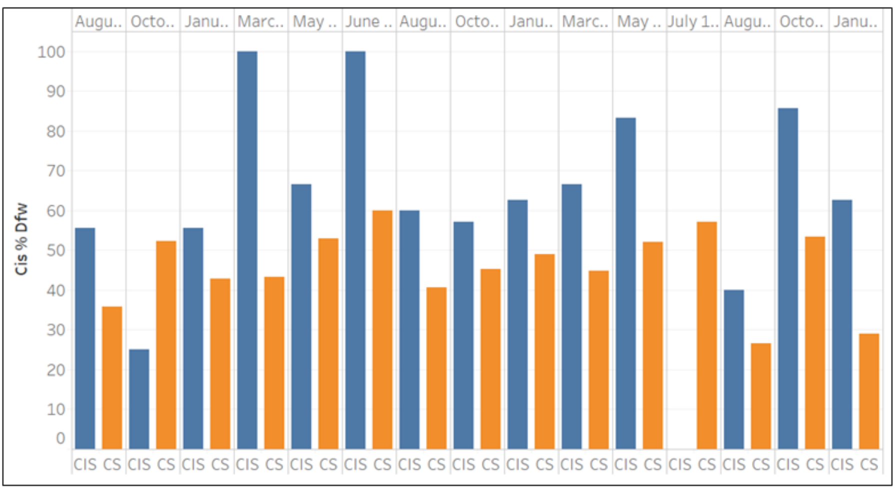


Figure 1: CS vs. CIS Drop-Fail Rates

As shown in Figure 1, CIS majors struggle considerably more than their CS counterparts. When summed, CIS majors average a 62% DF rate while CS majors have averaged a 42% DF rate. Combined, the two programs average approximately a 50% DF rate, a rate that we cannot sustain. Therefore, the challenge was to develop an introduction to programming course for CIS majors that will help them better retain what they learn, write better code and have a good understanding of the programming language (PL) that will be used in their next course, Java.

To fix any problem, one needs to understand it, thus we performed extensive research on the successes and failures of intro to programming courses. We quickly came to realize that students' success or failure in the introductory programming course often determines their majoring in computer science and related fields, yet many schools are challenged with high D/F rates [5, 7, 8]. For this very reason our first paper looked at why this is happening, and while there is no single reason, we found a number of factors that contribute to failure. For example, during the past four decades, many languages have been used for teaching introductory programming. Pears, et al. [9] provided a well-researched survey on the literature in teaching introductory computer courses and found that, C, Java, C++ and Python top the list of the most widely used programming languages. The Tiobe programming community index [4] (https://www.tiobe.com/tiobe-index/) also provides this information for the industry, is updated monthly, and interestingly has changed little over the years. Even though these are the most popular in business, Pears et al. [9] concluded that the best choice of a programming language (PL) is elusive, and in fact, a popular PL (ie: C, Java or Python) is not necessarily the best for teaching intro to programming courses. Of course, the language choice is usually made locally, based on factors such as faculty preference, industry relevance, technical aspects of the language, and the availability of useful tools and materials. However, we found in research that this process has become increasingly cumbersome for professors as the number of languages has grown.

One of the important things we pointed out in our previous paper is that Alturki [1] found that most institutions are starting students out in Java, C, C++, Python or Ruby. However, Barland [3] a professor at Radford, argues that C/C++ and Java are poor choices to learn as a first language. He went on to explain that, "Programming is a difficult task, learned over months and years. Object-oriented programming (the "++" part of "C++") is a more advanced topic which is important for larger programs but is best taught after the fundamentals have been learned." Barland [3] pointed out that most teachers realize that they should not distract from a topic by teaching advanced details to a beginner, yet that is exactly what's happening in these intro classes if we throw C++ or Java at them. As Malik [8] found in their intro to programming course, programming exercises were too difficult and as a result they were seeing about a 30% drop rate. Malik [8] also pointed out that the usual approach in teaching programming is to start with the syntax of a programming language (usually Java or a flavor of C) and move on to the associated semantics. This has proven to not be the best approach as motivation can quickly drop.

In the College of Computer and Information Sciences at this authors university, intro to programming and a second class in Java is required to continue in the CIS program. If a student drops out, fails, or passes with difficulty, Malik [8] found that it is unlikely that the student will enroll in a follow-up course.

This, of course, represents a huge loss of revenue to the University, and with high D/F rates, not a model that can be sustained. As a result, the goal of our previous paper was to find a way to improve these D/F rates. Easier said than done, but as Alturki [1] pointed out, researchers have found that failure is not always associated with cognitive ability, but often with motivation and teaching style, both in industry and education. Likewise, Álvarez [2] found that motivation in programming courses is directly related to student satisfaction. Because of this we decided to focus on motivation.

2 Research

2.1 Motivational and Demotivational Factors

s Alturki [1] explained, probably the biggest challenge in learning programming is to acquire different sets of skills at the same time. He found that new students had to learn both syntax and semantics of a programming language, while at the same time develop problem solving skills. To further compound this, most programming courses require students to study theoretical concepts and practice these concepts while designing and developing programs. Therefore, novice programmers must learn multiple concepts and apply these in a practical manner concurrently, often pushing students into overload [8]. Pillay and Jugoo [10] conducted a study to identify the relationship between student characteristics and performance in a first course in procedural programming. In this study, they found that for students who have a strong Mathematics background there is a positive correlation between the students' problem-solving ability and their programming performance in the course. In their 2016 study, Álvarez and Larrañaga [2] found that the biggest problem novice programmers face is their lack of program solving skills (much of which involve math). This produces high drop-out and failure rates in programming courses. [2] Though we did not pursue this in our efforts, we point it out because most institutions including ours do not have a math class as a pre-req to the intro to programming course.

In our previous paper we presented the idea of going back to the basics, BASIC programming that is, but not with traditional programs like QBASIC or MS-BASIC. Instead we introduced a product called Xojo (pronounced Zo-Jo), which is a free programming language that is quite like Visual Basic. As we started down that path, however, we came to realize that we're really just teaching another language, BASIC. Yes, a much easier language to teach (and learn) than "C" for example, but our ultimate goal is preparing students for the next class, which is Java. Time for something untraditional! Through further research, we found that Hong et al. [6] set out to make programming more interesting and more relevant to real-world problems. To accomplish this, their hypothesis was that "because children raised in an immigrant family are able to master two very different languages in their early childhood, first-time programming students should be able to learn multiple PLs at the same time." [6]. Hong et al. [6] also felt that just as a bilingual child can better appreciate two languages, programming students can also appreciate the designs and strengths of different PLs by way of a comparative study. Using control groups, the study [6] concluded that students can learn multiple PLs simultaneously without having a negative effect on their learning. They also found that students in a multiple PL environment learned and retained better and wrote better code than those students in a single PL environment. Specifically, within Hong's [6] experimental group, they observed that once a student came up with a solution in one PL, the student was often able to easily convert the solution to another in a different PL.

2.2 Teaching Multiple Programming Languages

Rather than teach one language, followed by another, like Hong [6] we proposed to teach multiple languages in our Introduction to Programming course. As previous authors [1, 3, 5, 9] have found, languages such as C/C++ and Java are poor choices to learn as a first language. Given the research presented, and to provide students with a more meaningful learning experience, a new course (CIS-275) was created within the CIS department to incorporate two programming languages, Xojo and Java. Xojo on the surface is a drag and drop programming environment with base code in BASIC. With Xojo, many different types of apps can be built, including web-based apps, console (or command line) apps, iOS apps and just released Android apps. Xojo is an object-oriented environment and is quite similar to VB.NET.

Research proved [6] that students can grasp multiple programming languages, thus we introduced our students to Xojo and Java. By using multiple programming languages our hypothesis was that the course will be more motivational, and students should be able to master multiple languages within the course. Overall, the goal was to greatly reduce these DF rates, and this was accomplished well beyond our initial goals. In the three terms completed since introducing this new method, on average, students went from a 38% pass rate to an 76% pass rate in this multi-programming course, where Xojo and Java are taught together. While this is a great success, it should be noted that those who went on to the follow-up Java course passed with a 75% success rate, as opposed to 47% previously. Also, of note, there were no students in the intro course that made it to the end and failed. In fact, all who did fail either never showed up for class or stopped contributing after the first week. This may be indicative of first courses that students take, but we did not pursue this observation.

In our informal experiment we found that assigning/offering both BASIC and Java simultaneously was beneficial to students understanding in that it helps them appreciate the differences among individual language designs. For example, for the same problem, we observed that when given a specific assignment in BASIC, students quickly and easily adapted it to Java. We also observed the existence of significant synergy in learning multiple PLs among the students. In our study we found that students learned to write better code in one PL by learning the other at the same time. One piece of clear evidence is that students in the new class started to write clearer Java code with proper syntax after they learned to code in BASIC. Our choice to teach BASIC and Java simultaneously has also proven enlightening. As shown in figure 2, we point out to students that BASIC and Java have many code similarities, and in the case of these statements the only real difference is the semicolon at the end of each line of Java.

BASIC Code	Java Code

Dim myName As String
myName = "Elvis Presley"
Dim oneNumber As Integer
oneNumber = 5
Dim oneNumber As Double
oneNumber = 5.3
Dim thisIsAwesome As Boolean
thisIsAwesome = True
 	
Note in Xojo variables can also be declared and initialized in one line.
Dim myName as String = “Elvis Presley”
Dim onNumber as Integer = 5
	String myName;
	myName = "Elvis Presley"; //don't forget the semicolon.
	int oneNumber;
	oneNumber = 5;
	double oneNumber;
	oneNumber = 5.3;
	boolean thisIsAwesome;
	thisIsAwesome = true;
	
	Just as in Xojo, Java variables can also be declared and initialized in one line.
	String myName = “Elvis Presley”;
	Int oneNumber = 5;

Figure 2: BASIC Code vs. Java Code

In one early exercise for example, students learn how an if-then-else works. This is a simple program that takes input from a form and presents the results in a message box.

Listing 1: BASIC code

```
Dim myAge As Integer
myage = cdbl(age.text) 'cdbl converts string to Integer
If myAge > 17 Then
  MsgBox "You are old enough to vote"
Else
  MsgBox "You are not yet old enough to vote"
End If
```

Next students write basically the same program in Java.

Listing 2: Equivalent Java code

```
public class NewClass
{
   public static void main(String[] args)
   {
      Scanner input = new Scanner(System.in);
      // Prompt the user to enter age
      System.out.print("Enter␣an␣age␣");
      double myage = input.nextDouble();

      if (myage > 17)
      {
         System.out.println("You␣are␣old␣enough␣to␣vote");
      }
      else
      {
         System.out.println("You␣are␣not␣yet␣old␣enough␣to␣vote"
                           + myage);
      }
    }
}
```

The "If" statement is similar, however the rest is a bit different, and it is in these early programming examples that we explain the use of brackets, comments, and semicolons in Java. By the time we get to week 6 of the course, students are not only following the Xojo manual and writing pretty sophisticated BASIC code such as sending an email, but they also do it in Java.

Listing 3: Java email code

```
public class SendEmailApp {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws IOException,
                      AddressException, MessagingException {
        //Setup system properties
        Properties properties = System.getProperties();
        properties.put("mail.smtp.starttls.enable", "true");
        properties.put("mail.smtp.auth", "true");
        properties.put("mail.smtp.host", "smtp.gmail.com");
        properties.put("mail.smtp.port", "587");
        String senderEmail = "XXX@gmail.com";
        String senderPassword = "xxx";

        //Instantiate mail session and
        //  compose email including subject, receipient & content
        Session mailSession =
```

```
            Session.getDefaultInstance(properties, null);
        MimeMessage mailMsg = new MimeMessage(mailSession);
        mailMsg.addRecipient(Message.RecipientType.TO,
            new InternetAddress("XXX@some.edu"));
        mailMsg.setSubject("Test java mail - Test");
        mailMsg.setContent("CIS 275 - This is test email.",
            "text/html");

        //Connection  to sender email & send
        Transport transport = mailSession.getTransport("smtp");
        transport.connect("smtp.gmail.com",senderEmail,
            senderPassword);
        transport.sendMessage(mailMsg, mailMsg.getAllRecipients());
        transport.close();
    }
}
```

As we move through each week in the course, our focus is on programming concepts and required syntax, with an emphasis on the comparison of Xojo and Java code in the visual environment. The goal is to prep the students for their next course which is purely a Java based course. For the Java piece, we introduce students to the Apache Netbeans visual environment. Similar to Xojo, Netbeans is a visual, drag and drop environment with Java as the underlying code. Building upon weekly concepts in both Xojo and Netbeans, students write simple, but meaningful, real-world programs in both Xojo and Netbeans. We believe this simple concept is what kept students motivated in the course, while at the same time preparing them for their future programming course.

3 Results

Since going to this new format, the class has run three semesters with the following results in the fall of 2018, spring 2019 and fall 2019.

As shown in figure 3 , we've gone from a previous passing average of 38% for CIS students to an average of 76.6% or more than double.

It should be noted that of the 17% (average) who failed the course, all stopped attending after the first week. Why they did not withdraw is unknown.

In the course that follows, CIS-375 (Java) we now see a passing rate of about 75% (figure 4) as opposed to 47% previously.

4 Conclusion

In this paper we presented an approach that teaches multiple programming languages simultaneously and comparatively to beginning programming students.

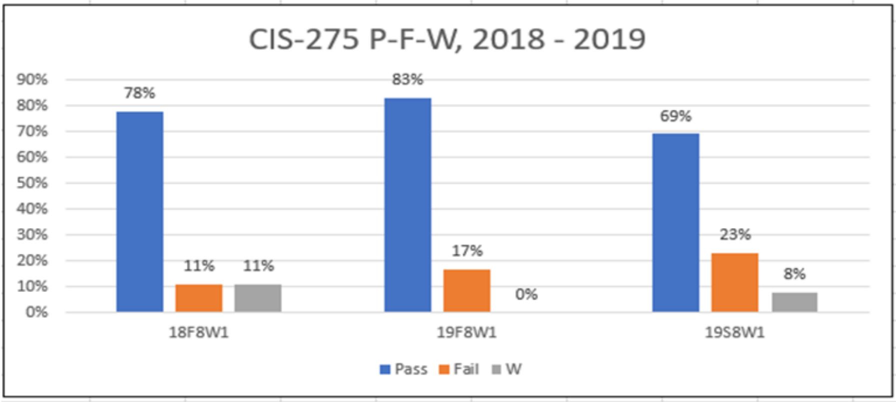


Figure 3: CIS 275 Pass-Fail-Withdraw Rates

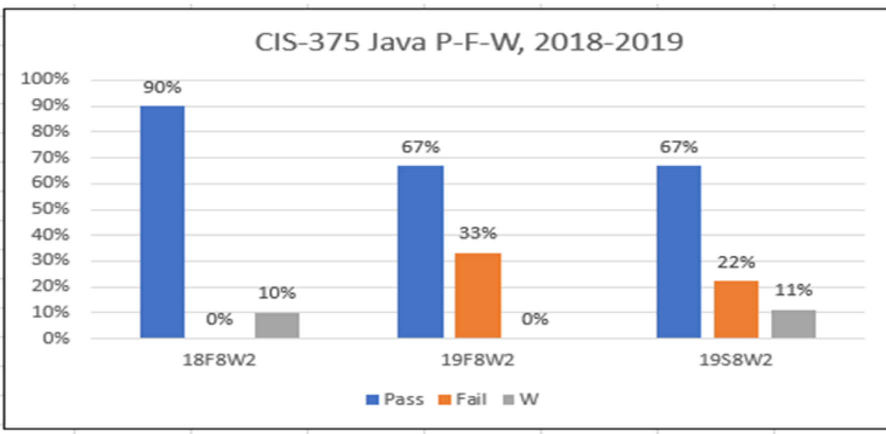


Figure 4: CIS 375 Pass-Fail-Withdraw Rates

Based on our previous proposal to go back to the basics, BASIC programming, we experimented with this approach by teaching BASIC and Java simultaneously. Going into this project there was concern that we might make matters worse. For example, is it possible for students to learn multiple programming languages simultaneously, or will this turn out to be a flop? The results overwhelming show that learning two programming languages simultaneously is not only possible, but also greatly reduced our D/F rates.

We have proven that when students are exposed to multiple PLs early, they can handle it without difficulty. Since our introduction of the two-language concept, we have seen that there is a clear benefit: learning multiple PLs simultaneously without loss of learning gains. Through this process we have found the need to focus more on teaching fundamental programming concepts rather than teaching distinct language syntax structures and features, and only teach these after students have a grip on the fundamentals.

Simplifying the course structure by reducing complexity has proven to have a positive effect on student performance, motivation and retention. Students focus more on the fundamental concepts in programming. By introducing these concepts in the Xojo environment, as well as contrasting Xojo programs to Java programs, students don't get bogged down in low-level programming languages that should be introduced later in a CS or CIS curriculum. If fact, as previously noted, we have a proven success rate of 76.6% versus 38% in the intro to programming course, and a 75% success rate of those who went from the intro course to Java.

References

[1] Raad A Alturki et al. Measuring and improving student performance in an introductory programming course. *Informatics in Education-An International Journal*, 15(2):183–204, 2016.

[2] Ainhoa Álvarez and Mikel Larrañaga. Experiences incorporating lego mindstorms robots in the basic programming syllabus: lessons learned. *Journal of Intelligent & Robotic Systems*, 81(1):117–129, 2016.

[3] Ian Barland. Why C and C++ are awful programming languages, 2017. `https://www.radford.edu/ibarland/Manifestoes/whyC++isBad.shtml`.

[4] TIOBE Software BV. TIOBE programming index, 2020. `https://www.tiobe.com/tiobe-index`.

[5] Elizabeth Simão Carvalho. Analyzing the quality of students interaction in a distance learning object-oriented programming discipline. *Interdisciplinary Journal of e-Skills and Lifelong Learning*, 11:085–099, 2015.

[6] Gongbing Hong, Jenq-Foung Yao, Chris Michael, and Lisa Phillips. A multilingual and comparative approach to teaching introductory computer programming. *Journal of Computing Sciences in Colleges*, 33(4):4–12, 2018.

[7] Ed Lindoo. Back to the basics. in an effort to improve student retention in intro to programming classes. *Journal of Computing Sciences in Colleges*, 34(2):72–79, 2018.

[8] Sohail Iqbal Malik and Jo Coldwell-Neilson. A model for teaching an introductory programming course using ADRI. *Education and Information Technologies*, 22(3):1089–1120, 2017.

[9] Arnold Pears, Stephen Seidman, Lauri Malmi, Linda Mannila, Elizabeth Adams, Jens Bennedsen, Marie Devlin, and James Paterson. A survey of literature on the teaching of introductory programming. In *Working group reports on ITiCSE on Innovation and technology in computer science education*, pages 204–223. 2007.

[10] Nelishia Pillay and Vikash R Jugoo. An investigation into student characteristics affecting novice programming performance. *ACM SIGCSE Bulletin*, 37(4):107–110, 2005.

An Empirical Study of How Novice Programmers Search the Web for Help*

Ryan Michaels[1], *Naveen Tula*[2], *Susie Ramisetty-Mikler*[2], *Dmitry Nurmuradov*[2], *Renee Bryce*[2]
[1] *Computer Science Department, St Edwards University*
3001 S Congress Ave, Austin, TX 78704
rmichael@stedwards.edu
[2] *Computer Science and Engineering Department,*
University of North Texas
1155 Union Cir, Denton, TX 76203
{Susie.Ramisetty-Mikler, Renee.Bryce}@unt.edu

Abstract

Students often use the web as a resource to help solve problems that they encounter on programming assignments. This work examines how students use the web to search for help. We analyze data from 444 student surveys and 41 focus group participants. The data reveals that as assignments grow more complex, student reliance on instructor provided material decreases and usage of Internet resources increases. Efficiency of searches did not similarly increase, with students spending 7.5 minutes to define their query and visiting 3.6 websites on average for help. The findings of this study reveal how students use the web to search for help on programming assignments, including how they format queries, the amount of search time, the average number of sites that they visit, and specific sites that they visit. The results guide future work to help students to more effectively search the web for help when they independently work on programming assignments.

1 Introduction

Computer science enrollment has grown in recent years, but retention of students is still an important issue. This work examines how students search for help when they leave the classroom to independently work on programming assignments. Students are required to learn not only new concepts, but the programming languages to implement solutions. Many students struggle to instantiate solutions to problems because they do not fully understand the concepts and further struggle to understand a programming language. In this work, we first conduct a student survey to identify the resources that students use and the perceived value of those resources. Based on the results of our survey, we focus on one particular resource – the web, and the online searches students perform. A focus group study then provides deeper insight into search strategies that students employ when working on programming assignments.

This paper is organized as follows: Section 2 discuses related work, Section 3 describes the methodology, Section 4 presents and discusses the results of the study. Section 5 discusses threats to validity. Section 6 presents conclusions.

2 Related Work

Most students encounter bugs while working on programming assignments. Fixing bugs in programming assignments has always been a major task and part of the learning process. A significant body of literature exists to classify the types of bugs that students encounter [2, 7, 8, 10] and ways to improve debugging efficiency [4, 6, 12]. Chmiel et al. [3] report that students who learn multiple debugging techniques spend less time resolving programming bugs [3].

Several approaches examine student coding patterns and bugs. Fenwick et al. [5] use their ClockIt Data Logger to track student coding patterns and analyze the behavior of students in Computer Science 1 (CS1) and Computer Science (CS2) courses. They observe that students who start early and work on their assignments regularly score higher grades. In our study, we do not focus on the amount of time that students spend on programming assignments, but rather the amount of time that they spend seeking help from online resources.

Alqadi and Meletic examine how novice programmers observe and fix bugs [1]. They perform two studies in which students are given code seeded with faults. They measure whether students find the errors and the effectiveness of their solution. Program size and years of programming experience correlate with student success to detect and fix bugs. Our work differs as we do not provide students with code, but rather track how they independently search for help for their own programming assignments across our curriculum.

Metzger [13] reports that the Internet is a major source of assistance and identifies several problems students experience when using the internet for help. Nasehi et al. [14] report that formulating a question is the first step toward finding relevant solutions to a question. Moreover, defining the areas of the problem that a student finds difficult helps to narrow the necessary search terms. Our work differs as we examine how students in different class years use the web to search for help on programming assignments.

Searching the web is a challenging process for many people [17]. This is particularly problematic for introductory programming students. As frustration accumulates, students may view coding as a difficult process and give up early if they do not find a solution quickly [15]. Giving up on a solution to their programming problem results in low grades and decreased satisfaction with the course [9]. On the other hand, many students succeed and we need to better understand resources that they use to independently solve their own programming problems.

3 Experimental Setup

This study analyzes data from student surveys and focus group participants. Test subjects are undergraduate students enrolled in computer science courses. We examine the following research questions:

> **RQ1:** What are the factors that measure the student's efficiency in solving the programming problems?
>
> **RQ2:** Where do students find help for programming problems?

3.1 Data Collection

3.1.1 Survey

We distributed a survey to undergraduate computer science students at the University of North Texas and received 444 responses. The survey assessed student efficiency measured by: 1) Search Parameter Time (SPT): time taken to define search parameters for the problem, 2) Solution Website Time (SWT): time taken to find the solution website, and 3) Total Number of Web Pages (TNWP): the total number of web pages they visited. The survey also tracks the type of resources used by the students, if students used the focus group website (not required for survey participants), and how students decided to search. Students also were asked the usefulness of a website, their demographics information, and expected grade in the course.

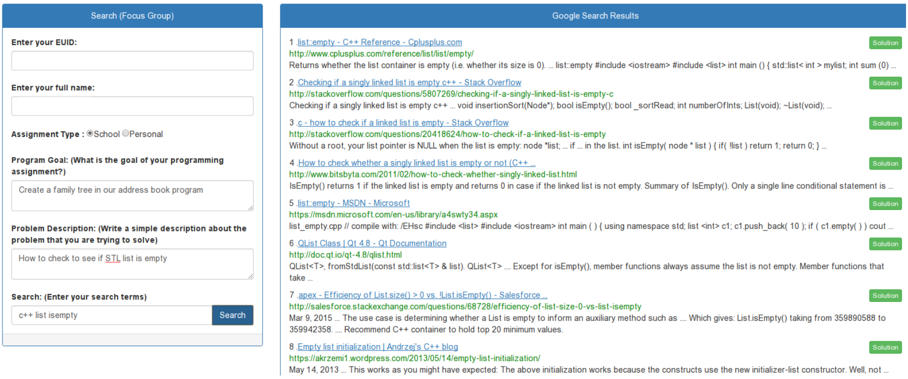


Figure 1: The *FG search engine*: The left side contains query entry fields. The right side shows the query results.

3.1.2 Focus Groups

We invited all students in our department to participate in the focus group. Students were accepted on a first-come/first-serve basis. A total of 50 students were selected and 41 students completed all of the requirements, including 11 freshmen, 14 sophomores, 10 juniors, and 6 seniors.

Each participant attended an orientation and installed the *Focus Group (FG) search engine* web browser plug-in. The *FG search engine* records data about student searches for help on programming assignments. Figure 1 shows a sample screenshot of the FG search engine to record student search patterns and website URLs. The engine retrieves information from Google using NodeJS and displays the results in a similar way to the Google search engine. The engine also captures: student name, assignment goal, problem description, and search terms. Once a student finds information to help solve their problem, they use the "solution button" on the right side of each search result to rate the usefulness of the respective URL.

3.2 Data Analysis

To assess student's efficiency in solving programming problems (outcome of interest) we measure three indicators: SPT, SWT, and TNWP. We analyze survey data using SPSS statistics [16]. Data cleaning was performed to ensure accuracy and completeness, to identify missing data patterns, and to regroup levels when extreme values were noted. Bi-variate associations [11] between the three outcome variables, predictor variables and demographics were tested.

T-tests [11] were used to test the mean difference between a two-level categorical variable and a continuous variable. One-way ANOVA [11] is used if the categorical variable is more than two-levels. Bi-variate associations between independent and dependent variables were investigated using cross tabulations with chi-square option [11] (for categorical variables) and significance level set at 0.05. Significance in mean differences were examined for class year and SPT, SWT, TNWP, forums, example code and tutorials.

To identify significant predictors, linear regression models were run separately for three outcome variables of interest, namely SWT, SPT, and TNWP. Based on the results from bi-variate analyses, regression models included demographic and predictor variables that were associated with the outcomes. The results were corrected for the influence of potential confounders, in this case gender and class level. A p-value<0.05 was considered statistically significant.

4 Results and Discussion

The results from the surveys and focus groups complement each other to reveal insight into how students use the web to search for help with programming bugs. The survey results provide a higher level overview with data from 444 students. The focus group provides deeper insight from 41 students.

Table 1: Mean and Standard Deviation of Bivariate Associations of Predictors and Outcome Variables

Predictors	**Num Students**	**Outcome Variables (mean; SD±)**		
		SPT (min.)	SWT (min.)	TNWP (min.)
Resources Used				
Textbook **Yes**/No	**173**/271	**9.2 (9.9)**/6.4 (6.9)	**3.2 (1.3)**/2.6 (1.2)	**4.0 (2.6)**/3.4 (1.5)
Class Notes **Yes**/No	**244**/200	**8.3 (8.9)**/6.6 (7.5)	**3.0 (1.3)**/2.7 (1.3)	**3.6 (1.7)**/3.7 (2.4)
Search Engine **Yes**/No	**306**/138	**7.2 (8.2)**/8.3 (8.7)	**2.8 (1.3)**/3.0 (1.4)	**3.7 (2.0)**/3.4 (2.1)
Specific Website **Yes**/No	**164**/280	**7.5 (8.4)**/7.5 (8.3)	**2.7 (1.3)**/3.0 (1.3)	**3.7 (2.2)**/3.6 (2.0)
Lecture Slides **Yes**/No	**315**/129	**7.9 (8.6)**/6.6 (7.8)	**2.9 (1.3)**/2.7 (1.4)	**3.6 (1.8)**/3.7 (2.5)
Decision to Search by				
Assignment	102	8.7 (9.7)	3.0 (1.4)	3.7 (2.6)
Error	275	7.0 (7.6)	2.7 (1.2)	3.5 (1.7)
Used Websites for a Portion of Assignments				
Less than a Half	111	5.4 (7.4)	2.7 (1.4)	2.9 (1.9)
Half	93	7.1 (7.6)	2.9 (1.1)	3.6 (1.7)
Most	160	9.0 (8.8)	2.8 (1.3)	3.9 (2.3)
All	80	7.6 (9.1)	3.1 (1.3)	3.8 (1.7)

4.1 Survey

A total of 444 students responded to the survey, of which 77% were males. The results of our statistical analysis of the surveys are displayed in Tables 1

- 4. The mean time taken to define search parameters for the problem was 7.5 minutes (SD±8.3), the mean response time to find the solution website was between 5 to 10 minutes, and the mean number of web pages students visited before finding the solution was 3.6 (SD±2.0).

RQ1: What are the factors that measure the student's efficiency in solving the programming problems? Regression analysis was performed for SPT, SWT, and TWNP. The regression model predicting SPT included textbook use, web help, decisions made based on the assignment or programming error, whether they visited the web for help on assignments, and helpfulness of sites with forums. Gender and class year were included as control. Only the textbook use (beta=0.13; p=0.015) and web help (beta=0.14; p=0.011) significantly predicted search parameter time after adjusting for class level. The regression model for SWT included textbook use and decisions made as those were significantly associated at bivariate level. Only textbook use significantly predicted (beta=0.21; p<0.001) after adjusting for gender and class year. Regression analysis for TNWP suggested that textbook use (beta=0.14; p=0.01) and whether they had web help for assignments (beta=0.16; p=0.004) significantly predicted SPT after adjusting for gender and class year. These results are shown in Tables 1- 3.

Table 2: Significance Tests (T-Test/F-Test) of Bivariate Associations Between Class Year, Resources Used, and Other Predictors and Outcome Variables

Predictors	**Outcome Variables (significance)**		
	SPT	SWT	TNWP
Class Year	F=1.35	F=0.78	F=1.36
Resources Used			
Textbook	T=3.1**	T=4.4***	T=2.8**
Class Notes	T=1.9	T=1.7	0.28
Search Engine	T=1.2	T=1.0	T=1.5
Specific Website	T=0.6	T=1.4	T=0.5
Class Lecture Slides	T=1.2	T=1.3	T=0.6
Decision to Search by	T=3.0	T=3.8	T=0.7
Used Websites for a Portion of Assignments	F=3.0*	F=1.3	F=4.3**

* p<0.05, ** p<0.01, *** p<0.001

RQ2: Where do students find help for programming problems? Students reported resource usage at the following rates: 71% used class lecture slides, 69% used search engines, 55% used class notes, 39% used textbooks, 37% used specific websites, and 7% used other resources. Student responses reflect the fact that a resource could be used alone or in combination with other resources. Focusing specifically on the web: 36% of students used web searches for most of their assignments, 25% for less than half of their assignments, 21% for half of their assignments, and 18% for all of their assignments. When searching for a solution, 62% searched by a specific error, 23% by an assignment topic, and 15% by other criteria. Students who reported using their textbook as a resource averaged a higher SPT and TWNP than those who did not.

Table 3: Bivariate Associations Between Class Year and Other Variables

	Freshm.	**Soph.**	**Junior**	**Senior**	χ^2 **(df)**
Resources Used (%)					
Textbook	36.5	41.1	38.0	40.0	0.53 (3)
Class Notes	65.0	57.3	50.0	45.0	7.6* (3)
Search Engine	63.5	64.5	60.0	87.1	17.3*** (3)
Specific Website	43.5	38.0	28.0	35.3	3.4 (3)
Class Lecture Slides	83.5	69.4	76.0	57.0	14.6** (3)
Decision to Search by (%)					13.0** (3)
Assignment	25.0	37.4	48.0	22.4	
Error	75.0	62.6	52.0	77.6	
Used Websites for a Portion of Assignments (%)					10.6** (9)
Less than a Half	26.0	24.2	34.0	19.0	
A Half	28.0	20.2	18.0	16.5	
Most	34.0	36.3	32.0	41.0	
All	12.0	19.4	16.0	23.5	

* p<0.05, ** p<0.01, *** p<0.001

4.2 Focus Group

The Focus group results reveal data about actual student searches and the student's perceived value of sites that they visited. Out of the 450 searches, 35% of students visited the website stackoverflow.com, 27% visited cplusplus.com, 4% visited tutorials.com, 2% visited cprogramming.com, and 2% visited YouTube. The remaining sites were all visited by 1% or less of student searches. The majority of students found their solutions on stackoverflow.com (35%) and cplusplus.com (27%). The high frequency of usage encourages us to explore and recommend these sites to future students.

RQ1: What are the factors that measure the student's efficiency in solving the programming problems? We classify efficiency in the focus group by analyzing their searches in relation to type of bugs and the efficiency measures above. Classification of programming bugs is done based on the assignment goal, problem description and search term used in the web browser plug-in. A breakdown of bugs per year is shown in Table 5. The most common bugs change as students progress through the program. As an example, freshman and sophomores face many bugs related to classes and functions, i.e., passing variables to functions, improper object instantiation, etc. These bugs occur in lower frequency for juniors and seniors. When juniors and seniors searched for syntax related bugs, their query composition time was reduced compared to freshman and sophomores.

The efficiency increase may be partially attributed to their previous experience of searching for similar issues. We believe that by offering instruction in both how to effectively structure a query and what constitutes a good result may help students.

RQ2: Where do students find help for programming problems? We classify the usage of forum and tutorial websites by the different class year students. Table 6 shows the percentage of focus group students who used the forums and tutorial websites. Usage of tutorial sites by freshman and sophomores (59% and 62%) outweigh their usage by juniors and seniors (41% and 27%). Freshman and juniors use forums (37% and 31%) at a lower rate than juniors and seniors (50% and 65%). The percentage of other resources is lower for freshmen (4%) and ranges from 8% to 9% for upper class years. The high usage of tutorial sites by freshman and sophomores reveals that they are often concerned with functionality and syntax of a programming language. Debriefings with the students at the end of the semester revealed that many juniors and seniors seek tutorial style sites when they were work in new languages or with new APIs. However juniors and seniors were primarily concerned with forum sites that offer assistance for specific problems.

5 Threats to Validity

Our survey was sent to 713 majors in our department and 444 completed the survey. While this is a significant number of participants, bias exists as some students chose not to complete the survey. Our focus group included students who signed up on a first come/first serve basis and initially consisted of 50 students, with only 41 students who completed the requirements and were included in the data set for this paper. Our future work will attempt to increase the sample sizes. The survey participants are all from one university. Results may differ at other institutions and in courses with different programming languages and technologies. Different problems may yield different search queries and results. A longer term study may yield additional insights.

Table 4: Bivariate Associations Between Class Year and Helpfulness of Different Website Categories

Websites	**Mean (SD±)**				**F-value**
	Freshm.	**Soph.**	**Junior**	**Senior**	
Forums	3.59 (1.1)	3.51 (1.0)	3.82 (1.1)	4.06 (0.9)	5.5***
Sites with Code Examples	3.76 (1.1)	3.61 (0.98)	3.68 (1.1)	4.01 (1.0)	2.5
Tutorial Sites	3.64 (1.2)	3.71 (1.1)	3.48 (1.1)	4.06 (1.1)	3.4**

* p<0.05, ** p<0.01, *** p<0.001

Table 5: Errors by Classification and Class Year as Percentages

Error Type	**Freshman**	**Sophomore**	**Junior**	**Senior**
Arrays & Vectors	8	12	13	8
Classes & Functions	28	24	8	11
Computer Environment	2	4	7	26
File I/O	3	12	10	13
Logic	11	5	14	21
Loops and switch statements	16	1	4	5
Pointer	8	5	1	0
Strings	8	12	11	8
Syntax	15	24	31	8

Table 6: Website Categories Used by Students in Different Class Years

Class Year	**Forums(%)**	**Tutorials(%)**	**Other(%)**
Freshman	37	59	4
Sophomore	31	62	8
Junior	50	41	9
Senior	65	27	8

6 Conclusion & Future Work

The results show that undergraduate students commonly use the web to search for help with programming bugs. Surveys reveal that 75% of the students use the web for help with at least half of their programming assignments. While web usage is common across all levels, student preferences of resources and websites varies. As courses progress, usage of class notes decrease and usage of web resources increase. When using a search engine, the mean time taken to define search parameters was 7.5 minutes and the mean number of web pages visited for solutions was 3.6. The most helpful websites include those with tutorials, examples of code, and forums that discuss similar questions to an assignment.

In the focus group, students mainly use a few websites to search for solutions to their programming bugs. As class year level progresses, the *search by the error* is given more preference compared to *search by topic*. The use of forums increases in junior and senior years. Usage of tutorial sites increased (in terms of efficiency) by junior and senior students.

Future work will examine whether teaching students structured examples of queries and sharing useful vs non-useful websites from our studies may improve their effectiveness of searching the web for help on programming assignments. Further studies will extend this work to examine search patterns of professional programmers and factors that influence their search efficiency.

References

[1] Basma S Alqadi and Jonathan I Maletic. An empirical study of debugging patterns among novices programmers. In *ACM SIGCSE*, pages 15–20, 2017.

[2] Renée C Bryce, Alison Cooley, Amy Hansen, and Nare Hayrapetyan. A one year empirical study of student programming bugs. In *Frontiers in Education Conference (FIE), 2010 IEEE*, pages F1G–1. IEEE, 2010.

[3] Ryan Chmiel and Michael C Loui. Debugging: from novice to expert. In *ACM SIGCSE Bulletin*, volume 36, pages 17–21. ACM, 2004.

[4] Du Chuntao. Empirical study on college students' debugging abilities in computer programming. In *Intl. Conf. on Information Science and Engineering (ICISE)*, pages 3319–3322. IEEE, 2009.

[5] James B Fenwick Jr, Cindy Norris, Frank E Barry, Josh Rountree, Cole J Spicer, and Scott D Cheek. Another look at the behaviors of novice programmers. In *ACM SIGCSE Bulletin*, volume 41, pages 296–300. ACM, 2009.

[6] Sue Fitzgerald, Renée McCauley, Brian Hanks, Laurie Murphy, Beth Simon, and Carol Zander. Debugging from the student perspective. *Education, IEEE Transactions on*, 53(3):390–396, 2010.

[7] Iris Gaber and Amir Kirsh. The effect of reporting known issues on students' work. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 74–79, 2018.

[8] Morgan Hall, Keri Laughter, Jessica Brown, Chelynn Day, Christopher Thatcher, and Renee Bryce. An empirical study of programming bugs in cs1, cs2, and cs3 homework submissions. *Journal of Computing Sciences in Colleges*, 28(2):87–94, 2012.

[9] Hua-Li Jian and Frode Eika Sandnes. Taiwanese and Norwegian engineering students' self-image of academic abilities, grades and course satisfaction. In *Intl. Conf. on Industrial Engineering and Engineering Management (IEEM)*, pages 993–996. IEEE, 2009.

[10] IIT Kanpur. List of common bugs and programming practices to avoid them. [Accessed: June 5, 2020].

[11] Ann Lehman, Norm O'Rourke, Larry Hatcher, and Edward Stepanski. *JMP for Basic Univariate and Multivariate Statistics: Methods for Researchers and Social Scientists*. SAS Institute, 2013.

[12] Y. Lin, C. Wu, T. Hou, Y. Lin, F. Yang, and C. Chang. Tracking students' cognitive processes during program debugging —an eye-movement approach. *IEEE Transactions on Education*, 59(3):175–186, 2015.

[13] Miriam J Metzger, Andrew J Flanagin, and Lara Zwarun. College student web use, perceptions of information credibility, and verification behavior. *Computers & Education*, 41(3):271–290, 2003.

[14] S. Nasehi, J. Sillito, F. Maurer, and C. Burns. What makes a good code example?: A study of programming q&a in stackoverflow. In *Intl. Conf. on Software Maintenance (ICSM)*, pages 25–34. IEEE, 2012.

[15] Ricardo Queirós. *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming.* IGI Global, 1 edition, 11 2014.

[16] IBM Corporation. SPSS Statistics Standard. [Accessed: June 4, 2020].

[17] Amanda Spink and Bernard J Jansen. *Web search: Public searching of the Web*, volume 6. Springer Science & Business Media, 2006.

The Japanese Fifth Generation Computing Project: A Brief Overview*

J. Paul Myers, Jr.[1], *Kayako Yamakoshi*[1,2]
[1]*Department of Computer Science*
Trinity University
San Antonio, TX 78212
pmyers@trinity.edu

[2]*Department of Engineering Management*
(Current Master's Student)
Duke University
Durham, NC 27708
kayamakos@gmail.com

Abstract

The history of computing and AI includes the largely forgotten Japanese Fifth Generation Computing Project in parallel knowledge-based AI that was announced at a sensitive time, capturing the imagination of the West (and the U.S.). After worldwide domination of the personal electronics and automotive industries, this Japanese Project was perceived to be a threat to achieve domination in the computing industry. While this was not actually the intent, the perceptions and fears led to large-scale responses throughout the U.S. and Europe. This Project is an interesting and worthy topic to be included in any history class/seminar in our field or for supplemental reading in an AI or Database class.

1 Introduction

Fading from our discipline's collective memory, even in Japan, is a fascinating and influential moment in computing history: The Japanese Fifth Generation

Computing Project of the 1980's (alternatively known as the Fifth Generation Computer Systems Project, and abbreviated herein as 5Gen). This "moment" spanned more than a decade: from some early planning meetings in 1978, to its announcement in 1981 and its formal beginning in 1982, to its projected end in 1993, and then to a few added years of activity to 1995. The most active work was conducted 1983-1993, a decade that proved to be important for computing.

5Gen was orchestrated and well-funded by the Japanese Ministry of International Trade and Industry (MITI) and delegated to its research subsidiary, the Institute for New Generation Computer Technology (ICOT), directed by Kazuo Fuchi. MITI created the Project as a coordinated effort involving industry, academia, and government.

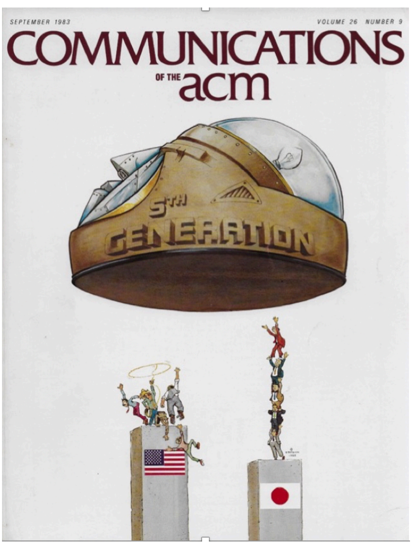
One of the most interesting aspects of 5Gen is how it was initially perceived in the West. The fact that it was to be a highly cooperative project of all sectors, combined with the then recent Japanese domination of the personal electronics and automotive industries, created a perfect storm of anxiety that the West (or U.S.) might lose its primary stance at the hands of the Japanese in yet another industry: computing. Indeed, in their book, artificial intelligence experts Edward Feigenbaum and Pamela McCorduck wrote [5]:

> We are writing this book because we are worried. But we are also basically optimistic. Americans invented this technology! If only we could focus our efforts, we should have little trouble dominating the second computer age as we dominated the first. We have a two- or three-year lead; that's large in the world of high technology. But we are squandering our lead at the rate of one day per day.

This book was written in 1983 as one response to 5Gen. Note the underlying sense of alarm, if not panic ("worried", "squandering"). Note the language ("dominating"). This was a heady time, long before any notion of the later dot-com booms and busts.

In the 1980's we referred to several generations of computing. The first four were machine/hardware-based (the vacuum tube, the transistor, the IC "chip" , and VLSI/microprocessor). The proposed fifth, while containing a hardware component emphasizing parallelism, was to be primarily software-based: pervasive artificial intelligence and inference processing using a relatively new computing paradigm known as logic programming (LP: symbolic logic as an approach to computation) and its then relatively obscure programming language exemplar, Prolog (PROgrammable LOGic). The 5Gen Project then was to develop robust knowledge information processing systems (KIPS), far exceeding the capabilities of the extant expert systems of the time. And the intent was for a high-performance personal Prolog machine with a "beautiful appearance" [15]. ICOT's initial vision was that the user would communicate intelligently with the computer using natural conversational language and images. We don't think that it's too far-fetched to say that the ultimate vision, thirty-five years ago, was something akin to the smartphone of today.

Attempting to match the announced Japanese cooperation among government, industry, and academia, the U.S. founded an umbrella corporation, MCC (Microelectronics and Computer Technology Corporation), to respond to the perceived threat. That the initiative was in fact perceived as a threat is evident in the Feigenbaum & McCorduck quote above and the following cover of the September 1983 *Communications of the ACM*, the flagship publication of the America-based leading international organization for computing. This image was also published by the ACM as a poster, yet another indication of the concern [2]:



The MCC partook in several activities and generated some spin-offs such as Cycorp (under Douglas Lenat; Cyc, creating an intuition and common sense knowledge base, is reputedly the world's longest-lived AI project). The MCC operations ended in 2000 and dissolved in 2004, well after the conclusion of the Japanese 5Gen.

In addition to the MCC, other responses to 5Gen emerged, mostly in Western countries: for example, Alvey (UK), SICS (Sweden), ESPRIT and ECRC (Europe). Moshe Vardi, in a private Communication [13], indicated that a study of the world-wide responses to 5Gen would be relevant [16]. To what extent the aforementioned fears were (or were not) warranted will also be taken up in this upcoming assessment.

Finally, here it is worth noting that research and development in AI has been punctuated by what some refer to as "AI Winters," periods of diminished attention/funding in AI. The intense focus on AI by both the Japanese 5Gen and the many Western responses could be seen to have led to a significant AI Spring; indeed, McCorduck wrote, "In the 1990's, shoots of green broke through the wintry AI soil" [9]. So, given the current importance of AI and the wealth of responses to 5Gen across many nations, this Japanese Project should not fade into historical obscurity.

2 Fifth Generation Software & Hardware

To accomplish the 5Gen goals, a great deal of emphasis was placed on parallelism for a relational knowledge base (KB) machine. KL1 (Kernel Language 1), a concurrent-Prolog-based LP language, was developed because Prolog does not presuppose a von Neumann (bottleneck) architecture and is inherently amenable to parallel computation. Moreover, it is a truism that in Prolog the KB = the program! Regarding standard relational databases, Fuchi wrote that now is the time for a "bold proposal" to develop "knowledge information processors" with a nontraditional architecture and presenting a more natural human-computer interface [7]: "At present it is common that databases and programming languages belong to different systems. This is not a desirable situation. Their unification appears to be quite feasible." The ideal is that the program = the data. Dahl [3] notes that

> "the logic program serving to define the data serves at the same time to compute it. In conventional database systems these two functions (description and retrieval ⋯) are performed by separate components, generally based on different formalisms. Also the fact that the operational aspects are automatically taken care of by an interpreter eliminates the need for a host programming language ⋯The user need only be concerned with the declarative semantics of [the] database."

This is demonstrated in any standard Prolog textbook, such as [1].

The key to fulfilling these ideals was the fact that the large apparatus of logical systems (axiomatic, natural deduction, ⋯) can be replaced with just one rule of inference: resolution. This necessitates listing all of a KB's statements in a specified format called Horn clausal form (see next page). Horn clauses constitute a normal - standard, uniform - form for FOPL (first-order predicate logic) in that there is an algorithm to convert an arbitrary FOPL statement into a set of Horn clauses [1]. All clauses are universally quantified (obviating the need for resolution to deal with quantification); and there is a particularly elegant procedure called Skolemization, to replace all existentially quantified variables with either constants or functions. Horn clauses are a very natural way of asserting knowledge and relationships; hence, the conversion algorithm rarely needs to be applied. Once in clausal form, resolution operates by a kind of cancelling of like terms on both sides of an implication, resulting in a new statement; a simple example:

$$\textit{Resolution}\ [\text{RES}]\quad \frac{\begin{array}{r} A \wedge \cancel{B} \wedge C \wedge D \rightarrow S \\ C \wedge E \rightarrow \cancel{B} \end{array}}{\therefore A \wedge\ C \wedge D \wedge E \rightarrow S}$$

Complication arises in that the terms to be cancelled (B here) must be *unified*: all variables/constants within the terms must be made to be identical. There are unification algorithms for this purpose, but they can be computationally intensive. For more technical details, e.g., for a classroom setting, on this and also how a LP/resolution program/KB performs an actual computation, see ([1], [8], [10]).

In the late 1970's Japanese researchers and committees met to determine new information technologies for the 1990's. Of the two main proposals (architecture oriented adaptive systems and software-oriented systems based on new languages), the latter was adopted and eventually became 5Gen. LP was seen to solve the problem of merging knowledge processing and parallel processing. A parallel inference machine (PIM), with its own operating system (PIMOS), was developed and was expected to have far-ranging applications such as VLSI CAD, image processing, genome analysis, legal reasoning, theorem-proving, Go, and others. We should note that the then AI lingua franca, LISP, was rejected in favor of LP. Prolog was to be a "kernel language, ranging from implementing the system itself up through the applications layers" : an unprecedented goal [6]. PIMOS was the world's first LP-based parallel OS.

3 Project Motivations

The reasons, explicit and implicit, for this Japanese endeavor were multidimensional. Japan is not resource self-sufficient given its geography and population density; therefore, it must rely on production, information, and knowledge. Entering into the world of computing as a major player was an advantage in terms of economics and prestige. Feigenbaum & McCorduck [5] posit a number of factors. For example, the Japanese thought that with the development of 5Gen, they would be maximizing their use of energy, improving society's ability to cope with an aging population, and expanding human abilities, from physical labor to intelligence. Indeed, Feigenbaum & McCorduck cite Sozeburo Okamatsu, a MITI official telling an American journalist: "Because we have only limited resources, we need a Japanese technological lead to earn money for food, oil, and coal. Until recently we chased foreign technology, but this time we'll pioneer a second computer revolution. If we don't, we won't survive."

5Gen was established as a counter to "chasing foreign technology." To some extent, the boldness of the vision was regarded as strange in Japan since it seemed un-Japanese to talk of the country's "taking its rightful place as a world leader, shedding its outdated copycat image, and claiming a role as a revolutionary innovator in high technology" [5]. The widespread reputation was that of a country that was adept at taking existing technologies to miniaturize and perfect them. 5Gen, on the other hand, was a creative innovation. Also, Japan was lagging in computing, 5Gen was a means to catch up [11]. As an aside, it's interesting that Fuchi realized that standard production modes would need to be disrupted. As one example, to counter the entrenched hierarchy of seniority where, in advanced technology especially, senior staff would not be adept, Fuchi insisted on hiring only those who were age 35 or under! This was granted to him by MITI.

4 Project Outcomes

Though there were many successes and well-received prototypes, the early 1990's came and went with no particular outcome broadly visible –certainly not the comprehensive, coherent outcomes that had been so dreaded by competing countries. With the hype and paranoia now both receded, Feigenbaum [4] has suggested that the time is right for a new reassessment of 5Gen, a project presently in progress by the authors [16]. There it is noted that one of the origins of Western anxiety was based on a stereotype of Japanese (Asian) characteristics: that the individual is second to the group/society. It was assumed that the three-sector cooperation would be smooth and natural for an Asian country. In fact, 5Gen was plagued by secrecy and competitiveness just as was the predicted case for the U.S.'s MCC.

Any attempt at characterizing the outcomes as either success or failure is much too simplistic for a project that was expected to meet many different goals and was large-scale, pervasive, and ultimately international in scope. Indeed, MITI and ICOT sponsored many international conferences and exchanges of researchers. But perhaps the single most important reason why the parallel LP paradigm fell into relative obscurity was very simple and unexpected. Namely, standard (von Neumann) machines had become so fast that now languages and hardware not especially well suited to AI could readily be adapted to standard procedural and OOP languages that were far more familiar to practitioners. This also seems to explain the virtual disappearance of LISP as "the" AI language. This phenomenon was coupled with an enduring resurgence of attention to neural networks, previously suffering their own "winter," now eclipsing 5Gen's LP approach. Shifts in the industry were so rapid that the apt choices in 1982 turned out to be "at odds with the computer industry's direction by

1992” [12]. ⋯Nonetheless, there were considerable positive outcomes; we will go into more depth in the forthcoming reassessment paper [16].

5 Curricular Applications

5Gen is a fascinating episode in the history of our field, both for its own ambitions and outcomes and for the intensity of reactions in the U.S. and the West. Yet the outcomes were diffuse and ambiguous; and since in some senses (only some!) the Project may be deemed to have failed, 5Gen seems perhaps not worthy of becoming the topic for a full course. However, in any course or seminar involving the history of computing or AI, 5Gen is a most apt topic for one of a number of lectures, discussions, or modules.

Moreover, the more technical 5Gen aspects of logic programming, resolution, unification, knowledge representation, etc. are well suited to be included in any course that emphasizes AI, knowledge bases, logic, etc.[10]. Such course inclusion would provide an alternative or supplement to neural nets and machine learning for AI: namely, the application of logic. Vardi [13] and others believe that a robust approach to AI will incorporate both "lizard brain" (quick/ fast: machine learning) and "fore-brain" (slow/logical reasoning for decisions). Hence, it is plausible that logic may emerge within the mainstream again, giving reason enough to introduce and focus on the LP paradigm. Indeed, while there are many disparate rankings, a recent ranking by *Business Insider* for 2020, lists proficiency in Prolog as the #7 language globally for salaries [14].

6 Acknowledgements

We are indebted to many people; we especially acknowledge the support, Summers 2016-2019, of Dr. Yoshihiro Omura (小村吉弘先生), Professor & Director of the International Exchange Center, Kindai University (近畿大学). His continuing encouragement proved to be essential for our progress. And Trinity University awarded a Faculty Summer Stipend and a Department of CS HEP Summer Student Research Fellowship for the respective authors.

References

[1] W. F. Clocksin and C. S. Mellish. *Programming in Prolog.* Springer-Verlag, New York, NY, 1981.

[2] Cover. *Communications of the ACM*, 26(9), 1983.

[3] V. Dahl. On database systems development through logic. *ACM Transactions on Database Systems*, 7(1), 1982.

[4] E. Feigenbaum. Recorded phone interview with the author, May 2017.

[5] E. Feigenbaum and P. McCorduck. *The Fifth Generation: AI and Japan's Computer Challenge to the World*. Addison-Wesley, Reading, MA, 1983.

[6] E. Feigenbaum and H. Shrobe. The Japanese national fifth generation project: Introduction, survey, and evaluation. *Future Generation Computer Systems*, (9), 1993.

[7] K. Fuchi. Aiming for knowledge information processing systems. In *International Conference on Fifth Generation Computer Systems*, October 1981.

[8] M. Ginsberg. *Essentials of Artificial Intelligence*. Morgan-Kaufmann, San Mateo, CA, 1993.

[9] P. McCorduck. *Machines Who Think*. A. K. Peters Ltd., Natick, MA, 2004.

[10] J. P. Myers, Jr. The Japanese fifth generation computing project: curricular application, 2020. (submitted for publication).

[11] Y. Nakamura and M. Shibuya. *Japan's Technology Policy: A Case Study of the Research and Development of the Fifth Generation Computer Systems*. Research Institute of International Trade and Industry. MITI, Tokyo, 1995.

[12] A. Pollack. 'Fifth generation' became Japan's lost generation. *NY Times*, June 5, 1992.

[13] M. Vardi. Conversation with the authors, September 2018.

[14] M. Ward. The 15 coding languages with the highest salaries, and how to learn them online at no cost. *Business Insider*, June 6, 2020.

[15] D. H. D. Warren. A view of the fifth generation and its impact. *AI Magazine*, 3(4), 1982.

[16] K. Yamakoshi and J. P. Myers, Jr. Assessing the Japanese fifth generation project. (in preparation).

Delving into Factors Influencing New York Crime Data with the Tools of Machine Learning*

Marcus Pinto[1], *Hsinrong Wei*[2], *Kiyatou Konate*[1], *Ida Touray*[1]
[1]*Computer Systems Technology Department*
NYC College of Technology, CUNY
Brooklyn, NY 11201
{mpinto, Kiyatou.Konate, Ida.Touray}@citytech.cuny.edu
[2]*Business and Information Systems Department*
US Marine Merchant Academy (USMMA)
Kings Point, NY 11024
weih@usmma.edu

Abstract

Among the many tools used for crime detection and prevention is that of machine learning techniques which shed light in the common quest of minimization and eradication of this negative action that affect the human society. We present a comparative study of criminal data on the five boroughs of New York city: Brooklyn, Bronx, Manhattan, Queens, and Staten Island for the entire year of 2019. For this supervised classification problem, we used the following classification models: Decision Tree, Multivariate Linear Regression, and kNN. The techniques employed in this study are based on the fact that past crime data trends helps us to correlate factors which might help understanding the future scope of crimes.

1 Introduction

One of the most challenging issues of police departments is to have accurate crime forecasts to dynamically deploy patrols and other resources so as to improve deterring of crime occurrence and police response times.

Traditionally, police officers use maps of an area and place a pin on the map for every reported incident. Studying these maps, they can detect these patterns and thus, to efficiently predict hotspots [5]. A hotspot is defined as the area with the higher possibility for a crime to occur, compared to the neighboring areas. While crimes may occur anywhere, it is normal for offenders to focus on the crime opportunities they face in the most familiar places, thus the hotspots for crimes. By identifying these crime zones and evaluating the type of the location and time of the crimes committed, we hope to raise awareness of the dangerous locations in certain periods of time.

The bottom line is to automatically detect patterns of crime in order to predict crime, anticipate criminal activity and prevent it. If automated, data-driven tools for crime pattern detection are made available to assist analysts, these tools could help police to better understand patterns of crime, leading to more precise attribution of past crimes, and the apprehension of suspects [11].

There has been tremendous increase in machine learning algorithms that have made crime prediction feasible based on past data [3]. We investigate machine learning architectures for crime hotspot prediction and present design recommendations. The work does not focus on the victim and the offender, but it can, among others perspectives, be extended to include prediction of occurrence of a certain crime type per location and time using our dataset. Our proposed application, along with potentially saving lives, can help people stay away from the locations at a certain time of day. Therefore, gaining this kind of awareness will help people make better choices about their places of living. Police forces, on the other hand, can use this approach to improve crime detection and prevention. In addition, this would be useful for allocating policing resources. It can help to ensure an efficient use of police resources in the distribution of police at most likely places of crime for any given time. By providing all of this information, we hope to make our community safer for the people who live there as well as for others who will travel there.

In this project various machine learning models like decision tree, multivariate linear regression, and kNN (k Nearest Neighbors) will be used to predict the borough where the crime occurred and the type of crime.

The technical details of the implementation of the proposed intelligent system is provided, the advantages and drawbacks of the technique is outlined, examples from recent research work on crime analysis are presented and current research trends are underlined to help the reader to delve into the matter.

2 Background

Violent crime (rape, robbery, aggravated assault, and murder) in the United States is most likely to be committed in urban areas. Similarly, it is no coincidence that many of the states with the highest rates of violence are also home to some of America's most dangerous cities. Crime appears to be closely tied to economic conditions. States with limited economic opportunity and a large percentage of residents struggling financially also tend to have higher violent crime rates. Conversely, economic conditions tend to be in states with lower crime. Acoording to a 2018 study done by 24/7 Wall St. [15] using data from FBI's 2018 Uniform Crime Reporting Program for all 50 states NY was classified as number 25 on the list of most violent states in the USA.

McClendon et al. [8], showed a comparative analysis between the actual statistical crime data and a given dataset of the Mississippi state. Three different machine learning algorithms were performed and claimed that the linear regression outperforms among them. One Nearest Neighbor, Decision Tree, Support Vector Machine, Naïve Bayes, and Neural Network were utilized to predict the hotspot of crime, stated by Yu et al. [18]. They ensemble these classification models to obtain the most realistic results.

Stec et al. [16], stated that crime prediction is taking advantage of deep neural networks to get crime count of next day in a fine-grain city partition. The counts of crime are divided into ten bins and neural network forecast the most suitable bin. Training of Chicago and Portland crime data is done using increasingly complex neural network structures. With best, it was possible to forecast the right bin for the overall count of crime with 65.3% correctness for Portland and 75.6% correctness for the Chicago.

Iqbal et al. [7], applied Naïve Bayesian and Decision Tree to a dataset to forecast the crime category for several states in the USA. Between the two, Decision Tree exceeded the Naïve Bayesian and attained accuracy of 83.95%.

Nath et al. [9], used a clustering technique for a data mining strategy which assists to identify the patterns of crimes and speed up the crime-solving procedure. Some enhancements of k-means clustering were implemented to help in the process patterns identification of crime and validate obtained results.

Saeed et al. [14], applied Decision Tree and Naive Bayes classifier to the criminal activity dataset to forecast the attributes and also event outcomes. After comparing, they stated that the Naïve Bayes is much stable and more precise on analysis of crime and can extract rules for classification.

Pradhan [12], suggested a technique for the design and implementation of detection of crime and identification of criminal in San Francisco, CA. Detection of crime is examined using the k-means clustering technique, which iteratively produces two clusters of crime that are based on related attributes of crime. Identification of criminal and prophecy are examined utilizing KNN classification. Finally, 93.62% and 93.99% accuracy is measured, respectively, in the production of two clusters of crime using chosen attributes of crime.

3 Project Goals

In this paper we analyze the 2019 New York city crime data using machine learning algorithms to predict crime hotspots and types of criminal offenses for additional police assistance. Since we are interested in discrete values of variables then our supervised learning is that of classification instead of regression.

We compare the performance of three different algorithms used in supervised learning: decision tree, multivariate linear regression, and kNN. A decision tree is an algorithm that uses a tree shaped graph or model of decisions including chance event outcomes, costs, and utility. Multivariate Linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables denoted X. kNN works on the simple assumption that similar things are always in close proximity. The algorithm classifies an unknown item by looking at k of its already -classified, nearest neighbor items that have similar attributes as the unknown item.

The main goal is to develop a tool as an additional option that permits police forces to quickly make data- informed recommendations about how to allocate resources in order to maintain security in cities.

4 Methodology

The dataset [2] used includes all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) for all complete quarters for the year 2019 which consists of 461,697 cases. It consists of thirty five columns like offense type, year, month, day, hour, location (borough, address, latitude, longitude), precinct, victim and suspect's age, gender, race, and many more.

Larceny/Theft is the most common crime with a frequency of 89312 and Other Traffic Infraction (OTI) is the least common crime with a frequency of 1.

The focus is on two columns of the dataset: BORO_NM and KY_CD, borough name and offense code respectively. We will be looking for the way the other twenty five independent variables relate to each one of these variables. In the first analysis BORO_NM is the target variable while KY_CD is one of the twenty five independent variables. Then we make KY_CD as our target variable while BORO_NM is part of the independent set of variables. Figures 1 and 2 show their histogram.

Initially we need to preprocess the dataset by removing all null values, infinite values, and all columns that are unnecessary. Categorical variables are converted into numerical variables with unique IDs. This approach is inspired by the work done on Kaggle [10]. There remained 26 from a total of 35 columns.

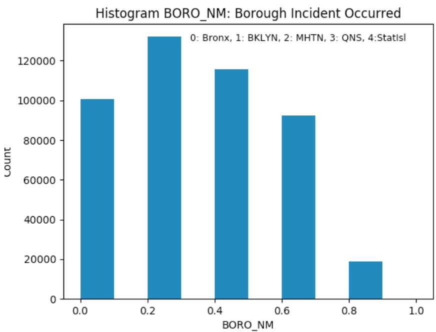


Figure 1: BORO_NM

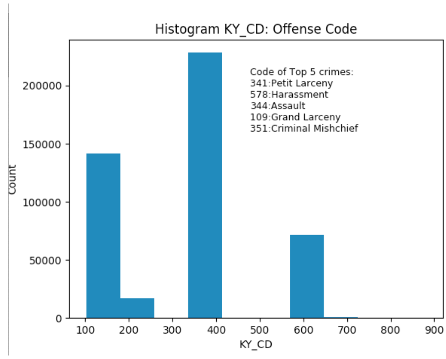


Figure 2: KY_CD

Twenty five of them are selected as features and the other is label (target) alternating between BORO_NM and KY_CD. The models were used to predict values for borough name and then for offense code.

Now it is the big question: should we normalize the dataset or not? There is those that argue that we should normalize it to get rid of noise which cause incorrect classifications, but there are those that argue that by normalizing it we remove important feature differences therefore causing accuracy to go down [6]. We took the latter choice since our dataset columns did not contain a very wide range of values which was reflected in the very small difference on the predictions using normalized or non-normalized datasets with exception of kNN when predicting offense code.

To avoid overfitting and getting more realistic accuracy, the dataset is divided into two portions: training dataset and testing dataset. Each dataset contains all features along with the target label. We set the value of the testing dataset size as 20% of the original dataset.

Next step is to train the algorithm with the training dataset (features and target). In this way the algorithm learn how to set its parameters (weight W and bias b) in order to predict correctly the values of the known targets(labels). The training process involves initializing some random values for W and b and adjusting the values for W and b such that we will have correct prediction.

Once training is complete, it's time to see if the model is any good, using Evaluation. This is where the testing dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This is meant to be representative of how the model might perform in the real world.

Finally, the model can make predictions; where the value of machine learning is realized. All of these steps were implemented using Scikit-Learn [15],

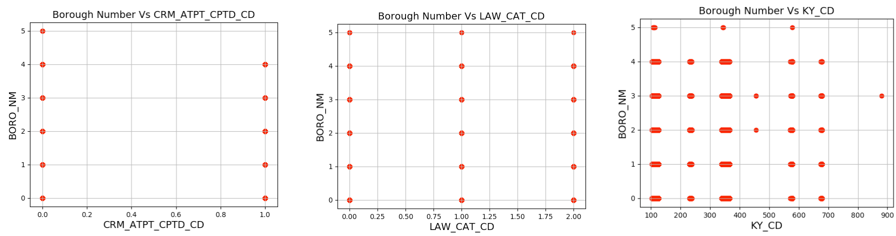


Figure 3: Linearity BORO_NM

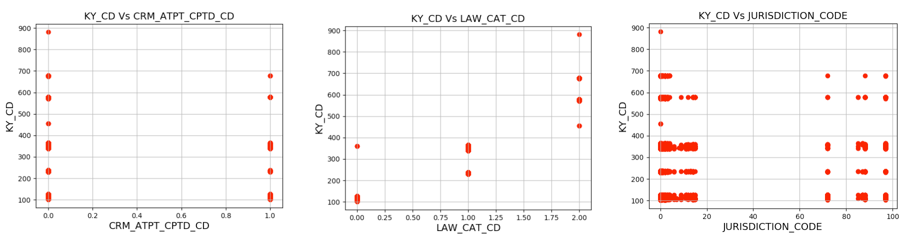


Figure 4: Linearity KY_CD

a free Python library that includes many supervised and unsupervised algorithms.

One of our models is multivariate linear regression and before implementing it we need to ensure that there is a linear relationship between the dependent variable (target) and the independent variable(s) (features). Figures 3 and 4 are scatter plots showing linearity between some of the independent variables and the targets, BORO_NM and KY_CD.

5 Results

We had two tasks: first to make prediction of the name of the borough where the crime occurred, for our dataset there are 6 values: Bronx, Brooklyn, Manhattan, Queens, Staten Island, and not specified (less than 1%); and second to predict the type of crime represented by the offense code (65 codes). The prediction hinges on the weights and biases assigned by the algorithm to each of the twenty five independent variables, aka, the dataset features. We used supervised machine learning algorithms such as decision tree, multivariate linear regression, and kNN.

Decision tree resulted in the best prediction (99.95%) for the borough where the crime occurred, followed closed behind by kNN (99.65%), while multivariate linear re-gression yielded an accuracy rate of 98.03% for the prediction of type of crime (offense) followed by a 57.07% accuracy using kNN. Table 1 below shows the accuracy rate (%) for these predictions.

	BORO_NM	KY_CD
Decision Tree	99.95	54.33
Multivariate Linear Regression	72.29	98.03
kNN	99.65	57.07

Table 1: Prediction Accuracy Rate (%)

The fact that decision tree algorithm gave us a better accuracy in predicting the correct borough name is because it can deal better with large datasets that have many layers with different nodes and a small set of targets, in our case only five boroughs[13]. Decision-tree classifiers can provide a better balance of flexibility and accuracy while limiting the number of possible decision points.

kNN is also suitable for small set of targets as well as features [4]. It yielded a second best borough prediction with only five relevant features (k=5) which might imply that most of the features are not immediately related or other factors not included in the dataset such as borough density, borough economic development index, etc, must be accounted for to give some indication of the features' relationship with each other when the focus is on the value of the borough name where the incident occurred.

Multivariate linear regression (MLR) is the top algorithm to predict the correct offense type which signifies that the dependent variable (KY_CD) are linearly dependent on the rest of the twenty five independent variables. There is also the possibility of temporal variables such as weather and holidays affect the type of crime [17].

The accuracy rate (significance level in statistics) is a function included in the scikit-learn library. To further validate our results we ran the prediction for a selected incident in the testing dataset and then comparing the predicted and actual values of the target variable (column). The comparisons shown in figures 5 and 6 follow the level of accuracy, the higher the accuracy rate the closer is the predicted value to the actual one.

6 Conclusion

Although it is the most populous city in the US, with a population of 8.6 million people, New York has one of the lowest crime rate (number of incidents per 100,000 people) in the nation. To maintain a low crime rate the city has

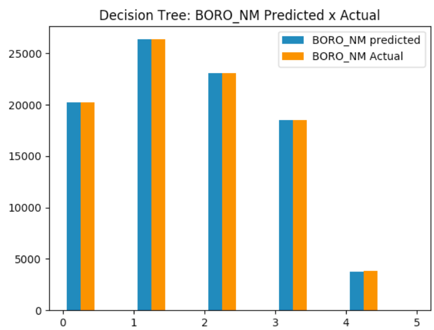


Figure 5: Decision Tree

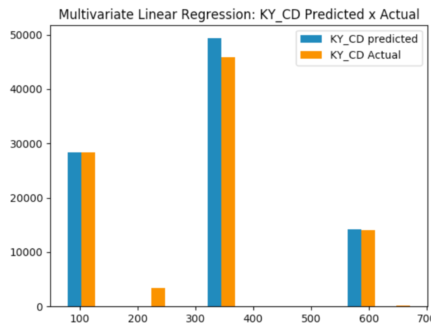


Figure 6: Multivariate Linear Regression

the largest police force in the country, with 77 precincts spread across five boroughs. To further enhance its ability to lower the crime rate, it started in 2016 to employ a machine-learning software that sifts through police data to find patterns and connect similar crimes [1].

We demonstrate a machine learning method designed to provide improved prediction of future crime hotspots and type of crimes, with results validated by actual crime data. The model tuned using supervised classification algorithms. provides the best performance. We can generate many insights on some other relevant variables of the dataset such as geographical location of the incidents that can help in understanding New York city crimes datasets thus assisting us in capturing the factors that affect society safety.

Effectively combating and preventing crime requires continuous improvements to data-driven methods. Future efforts will seek to integrate additional data sources reflecting economic and social conditions to better minimize the crime rate in the cities. Additional sources of unstructured data, such as narrative reports, could thus potentially be integrated to further improve crime prevention efforts.

References

[1] NYPD tracks crimes across city with its own pattern-seeking AI tool, 2019. https://www.aitrends.com/ai-in-government/nypd-tracks-crimes-across-city-with-its-own-pattern-seeking-ai-tool/.

[2] NYPD complaint data current year to date, 2020. https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Current-Year-To-Date-/5uac-w243.

[3] E. Amina. Using machine learning for crime prediction, 2018. https://digital.hbs.edu/platform-rctom/submission/using-machine-learning-for-crime-prediction/.

[4] Laurent Amsaleg, Gylfi Þór Guðmundsson, Cathal Gurrin, Björn Þór Jónsson, and Shin'ichi Satoh. *MultiMedia Modeling: 23rd International Conference, MMM 2017, Reykjavik, Iceland, January 4-6, 2017, Proceedings, Part I*, volume 10132. Springer, 2016.

[5] Anthony A Braga, Brandon Turchan, Andrew V Papachristos, and David M Hureau. Hot spots policing of small geographic areas effects on crime. *Campbell Systematic Reviews*, 15(3):e1046, 2019.

[6] SAS Communities. To standardize data or not to standardize data - that is the question, 2017. https://communities.sas.com/t5/SAS-Communities-Library/To-standardize-data-or-not-to-standardize-data-that-is-the/ta-p/361726.

[7] Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Payam Hassany Shariat Panahy, and Nasim Khanahmadliravi. An experimental study of classification algorithms for crime prediction. *Indian Journal of Science and Technology*, 6(3):4219–4225, 2013.

[8] Lawrence McClendon and Natarajan Meghanathan. Using machine learning algorithms to analyze crime data. *Machine Learning and Applications: An International Journal (MLAIJ)*, 2(1):1–12, 2015.

[9] Shyam Varan Nath. Crime pattern detection using data mining. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops*, pages 41–44. IEEE, 2006.

[10] Nicholas. Analysis strategy, 2018. https://www.kaggle.com/nickpaps/analysis-strategy.

[11] Walt L Perry. *Predictive policing: The role of crime forecasting in law enforcement operations.* Rand Corporation, 2013.

[12] Isha Pradhan. Exploratory data analysis and crime prediction in San Francisco. 2018.

[13] Lior Rokach and Oded Z Maimon. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008.

[14] Umair Saeed, Muhammad Sarim, Amna Usmani, Aniqa Mukhtar, Abdul Basit Shaikh, and Sheikh Kashif Raffat. Application of machine learning algorithms in crime classification and classification rule mining. *Research Journal of Recent Sciences*, 4(3):106–114, 2015.

[15] Samuel Stebbins. Most dangerous states in America, 2020. `https://247wallst.com/special-report/2020/01/09/most-dangerous-states-in-america-5/`.

[16] Alexander Stec and Diego Klabjan. Forecasting crime with deep learning. *arXiv preprint arXiv:1806.01486*, 2018.

[17] Sherry Towers, Siqiao Chen, Abish Malik, and David Ebert. Factors influencing temporal patterns in crime in a large American city: A predictive analytics perspective. *PLoS one*, 13(10):e0205151, 2018.

[18] Chung-Hsien Yu, Max W Ward, Melissa Morabito, and Wei Ding. Crime forecasting using data mining techniques. In *2011 IEEE 11th international conference on data mining workshops*, pages 779–786. IEEE, 2011.

A System for Automatic Lexical Acquisition From PubMed*

Daniel McDonald
Information Systems & Technology Department
Utah Valley University
Orem, UT 84058
daniel.mcdonald@uvu.edu

Abstract

PubMed is a repository of 30 million citations and abstracts from biomedical and life sciences publications, maintained by the U.S. National Library of Medicine. It is growing at a rate of one million new abstracts per year. With the addition of new abstracts comes the addition of new words. New genes are identified and named along with new diseases. Tools for automatically processing PubMed documents often rely on having a comprehensive lexicon. In this paper, the authors present a system, built on the Google Cloud Platform (GCP), for automatically processing PubMed abstracts to extract new words and automatically assign them to existing word categories. The system extracted 3.6 million new words from 27 million abstracts and was able to automatically assign 68 thousand of those words to existing word categories. This work is part of a larger project to lexically analyze the changes in Nursing research over the last 30 years.

1 Introduction

PubMed is a repository of more than 30 million citations and abstracts that are available for free download. PubMed is maintained by the U.S. National Library of Medicine (NLM) [12]. The citations and abstracts come from the biomedicine and health fields and include related disciplines, such as life sciences, behavioral sciences, chemical sciences, and bioengineering [12]. In 2019, there were 956,390 new MEDLINE (a component of PubMed) citations indexed [11]. The website pubmed.gov allows you to search the indexed abstracts. In 2019, the National Library of Medicine reported that 3.3 Billion searches were performed [11].

The PubMed collection has attracted government-led research for over 30 years. Led by DARPA and coordinated with the National Institute of Standards and Technology (NIST) in 1991, the U.S. government created the TIPSTER text project (including TREC) to advance the state of the art in text processing. TREC currently has tracks related to biomedical informatics including a genomics track and a clinical decision support track.

A text processing event specific to the biomedical text domain was organized in 2004 titled the Critical Assessment of Information Extraction in Biology (BioCreative) [2]. An early BioCreative task was to identify genes and proteins from text and map them to standard identifiers in the fly, mouse, or yeast databases. An additional task included identifying concepts in full-text articles that supported Gene Ontology (GO) annotations. Current tasks in BioCreative include automatically annotating biomedical text, integrating ontologies and text, detecting drugs and compounds in text, and automatically extracting experimental methods from publications.

The collection has also attracted public and private tools to help exploit the wealth of research that is contained in the citations. There have been systems to extract gene, protein, enzyme and molecular interactions and pathways from text [5, 7, 9, 10, 13, 15]. There has been algorithms to automatically assign MeSH terms or other annotations to PubMed abstracts [3, 8]. There has also been systems for clinical radiology data management and systems to index findings and diagnoses [4, 6]. There continues to be a great potential for text analysis applications in systems biology [1].

Often, a critical component of PubMed systems that retrieve relevant documents, extract named entities, and visualize relationships is the acquisition of new lexicon. As mentioned above, there are nearly one million new abstracts added each year. With new abstracts come the identification of new genes and proteins and new diseases. In this paper, we present a system to automatically extract new lexicon from PubMed and assign new words to relevant noun categories. This paper represents a step in a larger ongoing research project to analyze the changes topics in nursing research across decades.

2 Methodology

For flexibility and scalability reasons, the system developed was created to run on the Google Cloud Platform (GCP). The authors received a grant from Google to use the platform. The system can be separated into four main categories of programs: programs for downloading the PubMed XML files, programs for parsing the XML and populating the Firestore document database, programs for reading all the citations and creating lists of new words, and finally programs for predicting the noun categories of new nouns.

2.1 Downloading PubMed XML Files

We created a Virtual Machine instance using the GCP Compute Engine. The instance had 1 vCPU with 10 GB of RAM and ran CentOS 7. We connected to the instance using a Google Chrome terminal window. Once connected to the VM, the authors installed OpenJDK version 11.0.6 and the Python 3.6 virtual environment.

The authors used a program written in Java which relied on FTP libraries from the Apache Commons Project. The Apache Commons includes classes for FTP and FTPClient. Once an email address was registered with the National Library of Medicine (NLM), the Apache libraries were used to connect to the FTP servers of the NLM, authenticate and then download the compressed XML files. The Java code managed the downloads of the files and verified the checksums of all the downloads. The program downloaded 812 baseline files and 370 update files for a total of 1,182 compressed XML files. Baseline files are the complete files as of the first of the year. Update files contain any additions and changes to the baseline that have taken place since the beginning of the year. The files were split between two directories: baseline and updates.

2.2 Parsing XML Files and Populating Firestore

With all of PubMed downloaded onto the CentOS VM, the authors used the shell to create two separate text files that contained the name of all the files to be processed. The authors then wrote a program in Python to decompress the gzipped files, which were around 35 MB a piece, process the XML in the files, and insert the data into Firestore. Firestore is a NOSQL cloud database that runs on the GCP. It is a document database that is schema free and uses a JSON-like storage representation.

Each compressed XML file contained around 30,000 PubMed citations. Instead of inserting all attributes of each citation into the database, the following nine attributes were extracted from each citation and added to Firestore: ISOJournalAbbrev, abstractText, abstractTitle, authors, country, fileSource, journalTitle, language, and pubYear. The authors plan to analyze how nursing research has changed over time and these nine attributes were considered necessary to identify nursing research, its publication outlet and publication date. This process ran for about three days.

2.3 Reading all Citations and Extracting New Words

Next, we wanted to read every abstract title and abstract text and compare the words and phrases from the abstracts to an existing lexicon of about 1.7 million lexical entries. If no matching entries were found in the existing lexicon, then the word would be added to a collection of new words along with a frequency count of how often the new word appeared in the PubMed database.

The authors harnessed the scalability of GCP to speed the processing of the 27 million English abstracts that had been added to Firestore. First, the authors tried querying Firestore to return all PMIDs (PubMed ids) so that they could distribute the processing of the abstracts. However, the query was too slow and the connection to the database would time-out so that it had to be constantly restarted. In order to replace the query, the authors wrote another Python program to extract all the PMIDs from the original compressed XML files. This program ran faster and was much more successful. Another program was written to separate the 27 million PMIDs into 27 different text files with one million PMIDs per file. This file would be used to distribute the processing of the abstracts.

The authors created five more compute engine instances on GCP again running CentOS 7. The OpenJDK again was installed along with the Gradle build system. A Java program was written that would read the text file of one million PMIDs and then process one at a time querying Firestore for the abstract title and abstract text. The Java program would then tag the abstract title and text with its existing lexicon. Any words that were not in the lexicon were added to text file and then a new query of Firestore would take place. So, the current architecture had five nodes processing abstract titles and text concurrently. When a Compute Engine completed a text file, it would be restarted with a new text file of the next one million PMIDs. The five compute engines completed processing all 27 text files in two days.

Because there were 27 text files with one million PMIDs in each file, there were 27 text files that had all the new words that resulted from the PMIDs from that file. The authors downloaded the 27 text files of new words and then wrote a Java Program to combine the files into a single file of new words. All numbers were removed from the text file, which left a total of 3,619,239 new words for the lexicon. Table 1 shows some of the new words from our analysis.

2.4 Applying Lexical Patterns to Predict Word Categories of New Words

In the current system, word categories live in a hierarchy. A word can simply be tagged a noun, which means it is not an adjective, verb or adverb. A word can also be tagged a gene, which is a type of noun and is defined beneath noun

Table 1: New words from PubMed with frequency

Word	Frequency
convulsant	3,187
hemophiliacs	2,982
schistosomula	2,929
endoprostheses	2,880
prekallikrein	2,760
truncus	2,707

in the hierarchy. Patterns were used to label words into the most specific word category as possible. In addition to any lexical semantic label, a word's label also captured whether the word was capitalized and whether it was plural or singular.

The final step in the automated process was to read through each new word and compare it to a list of patterns for matching new words. The list of patterns was manually curated by looking at naming conventions for biomedical and chemical words [14] as well as general words. The list of patterns was pruned by evaluating the patterns against the existing lexicon. Patterns that were too broad produced too many false matches. For example, words that end in -ase are often enzymes, such as kinase. However, you must look out for words like case, phase, base, disease, and increase. As a result, -ase may be too short to be useful.

Longer patterns were given preference over shorter patterns because of the reduced possibility of false matches. Patterns were processed in the file from top to bottom, so the most accurate or descriptive patterns were generally placed near the top of the list, while the less accurate/descriptive patterns were put near the bottom. For example, words that ended with clinical where labeled as a clinical adjective. Whereas words that ended in -ical were given just an adjective label. The pattern -clinical appeared high in the list of patterns.

The list of patterns included matches to prefixes as well as matches to suffixes. Table 2 shows some prefix matching patterns.

Table 3 shows some suffix matching patterns. In total there were 903 patterns that were compared against new words.

3 Results

The text file of 3,619,239 new words were compared against each of the 903 patterns. The result was that 67,807 words were matched to a pattern and given a word category that contained lexical semantic information along with

Table 2: Patterns that match word prefixes

Word Prefixes	Rule and Resulting Tag
Methyl	START:CAPITAL;TAG=ORGANICCOMPOUND_NUSXFB
phenyl	START:LOWERCASE;TAG=ORGANICCOMPOUND_NLSXFB
Pheno	START:CAPITAL;TAG=ORGANICCOMPOUND_NUSXFB
benzo	START:LOWERCASE;TAG=ORGANICCOMPOUND_NLSXFB

Table 3: Patterns that match word suffixes

Word Suffixes	Rule and Resulting Tag
ectomy	START:LOWERCASE;TAG=SURGERY_NLSXFB
ectomies	START:CAPITAL;TAG=SURGERY_NUPXFB
splasias	START:LOWERCASE;TAG=MEDICALCONDITION_NLPXFB
xides	START:CAPITAL;TAG=BIOCOMPOUND_NUPXFB
oxyl	START:LOWERCASE;TAG=RADICAL_NLSXFB
graphy	START:LOWERCASE;TAG=MEDIMAGING_NLSXFB
scope	START:CAPITAL;TAG=IMAGINGTOOL_NUSXFB

information about the word's part of speech, whether it was uppercase or not. Some of the words that were automatically tagged are shown in Table 4.

Table 4: Automatically tagged words

Word	Tag
sygmoidoscopy	DIAGNOSTIC_NLSXFB
selenotherapy	TREATMENT_NLSXFB
stereoaortography	MEDIMAGING_NLSXFB
Livertectomy	SURGERY_NUSXFB

A tag in the system is separated into two parts by the underscore. The first part of the tag references the tag's lexical semantic category. The second part after the tag encodes additional information. For example, in Table 4, the first tag DIAGNOSTIC_NLSXFB, the 'N' stands for NOUN, the 'L' stands for LOWERCASE and the 'S' stands for SINGULAR. The other letters are used later in language processing.

4 Conclusions

The authors presented a system running on the Google Cloud Platform for downloading 27 million PubMed abstracts, extracting and loading those abstracts into Firestore, a document database, and then processing the abstracts using five Compute Engines running CentOS. The abstracts were processed to recognize new words not a part of the existing lexicon, which already contained 1.7 million words. A total of 3.6 million new words were extracted. Of the new words, almost 68 thousand were automatically matched by prefix and suffix patterns and assigned to lexical semantic categories.

With so many new words (over 3 million), automatically matching 68 thousand may seem like a low number. However, of the new words, almost two million of them appeared only once in all of PubMed. Perhaps the words were misspelled or there was a missing space between two common words. Many of the new words include combinations of numbers and letters. The purpose of adding words to our lexicon is to increase the sensitivity of our comparison of research topics over the years. We are interested in words that appear a lot, in patterns, across multiple research abstracts and not words that appear only once in the PubMed database. Given this perspective, adding 68 thousand words to the lexicon is a meaningful addition because it will improve the sensitivity of the comparison between abstracts from different years.

5 Acknowledgements

The authors would like to acknowledge the generous grant from the Google Cloud Platform (GCP) that made this project possible.

References

[1] S. Ananiadou, D.B. Kell, and J. i. Tsujii. Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12):571–579, 2006.

[2] Blaschke and A. Valencia. Overview of biocreative: Critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(1):1–10, 2005.

[3] D. Chen, H.-M. Müller, and P.W. Sternberg. Automatic document classification of biological literature. *BMC Bioinformatics*, 7(1):1–11, 2006.

[4] C. Friedman, P.O. Alderson, J.H. Austin, J.J. Cimino, and S.B. Johnson. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174, 1994.

[5] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. Genies: A natural-language processing system for the extraction of molecular pathways from journal articles. *Journal of Bioinformatics*, 17(1):S74–S82, 2001.

[6] W. Hersh, M. Mailhot, C. Arnott-Smith, and H. Lowe. Selective automated indexing of findings and diagnoses in radiology reports. *Journal of Biomedical Informatics*, 34(4):262–273, 2001.

[7] K. Humphreys, G. Demetriou, and R. Gaizauskas. Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Pacific Symposium on Biocomputing*, PSB 2000, pages 502–513, 2000.

[8] K. Ibushi, N. Collier, and J. Tsujii. Classification of medline abstracts. *Genome Informatics*, 10:290–291, 1999.

[9] D. McDonald, H. Chen, H. Su, and B. Marshall. Extracting gene pathway relations using a hybrid grammar: The arizona relation parser. *Bioinformatics*, 20 18:3370–8, 2004.

[10] S. Novichkova, S. Egorov, and N. Daraselia. Medscan, a natural language processing engine for medline abstracts. *Journal of Bioinformatics*, 19(13):1699–1706, 2003.

[11] National Library of Medicine. MEDLINE PubMed production statistics. `https://www.nlm.nih.gov/bsd/medline_pubmed_production_stats.html`. Accessed 2020-07-10.

[12] National Library of Medicine. PubMed overview. `https://pubmed.ncbi.nlm.nih.gov/about/`. Accessed 2020-07-10.

[13] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. Automated extraction of information on protein–protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161, 2001.

[14] Wikipedia. List of medical roots, suffixes and prefixes. `https://en.wikipedia.org/wiki/List_of_medical_roots,_suffixes_and_prefixes`. Accessed 2020-07-10.

[15] D. Zhou and Y. He. Extracting interactions between proteins from the literature. *Journal of Biomedical Informatics*, 41(2):393–407, 2008.

Teaching Students about Usability Testing in Distance Learning Environments*

Osvaldo Jiménez, Daniel C. Cliburn
Computer Science Department
University of the Pacific
Stockton, CA 95211
{ojimenez,dcliburn}@pacific.edu

Abstract

This paper documents two different approaches taken to teach students how to conduct remote usability testing during the COVID-19 pandemic. The pandemic has necessitated new requirements that limit or entirely exclude students from sharing the same physical space at the same time. The paper will first detail how two different sections of a human-computer interface design course implemented remote usability testing into their offerings. The paper will then use the notes and details from those two courses to analyze and discuss key points learned. By discussing the implementations and lessons learned, the paper aims to be informative for instructors interested in teaching or supporting students to conduct remote usability testing in the future.

1 Introduction

Usability testing involves testing whether a product is usable in a controlled environment [9]. While one of the cornerstones of human-computer interface design has been that of having a usability test [9], since the proliferation of the web, there have been calls to consider conducting usability testing remotely, as they could cut down on costs [10, 7, 11, 4] and provide ways of increasing sample size [8, 6] and diversity [4] when compared to having subjects come in person to a lab. Since there are many ways to perform remote usability testing, including allowing individuals to file self-reports [2], some in the design community have questioned its effectiveness in particular forms [7]. Meanwhile, others have provided best practices [10] for how to implement remote usability studies. Two practices highlighted in remote usability testing have been interacting with users synchronously via video conferencing software and remotely monitoring them at the same time. Research has shown those two practices are comparable to in-lab testing [1, 11] when it comes to helping researchers find usability problems. Additional modalities have also been explored for remote usability testing including using virtual reality [4]. Nonetheless, that and other studies still highlight that doing remote video conferencing is comparable to in-person lab testing on a majority of factors, with some differences reported in how users feel in each environment [3, 4]. Based on this information, the authors would not claim that remote usability testing is completely equivalent to in person testing, as other researchers have even found that the errors that users find in each modality differ slightly [10]. Instead, the authors would argue that remote usability testing, while not equivalent, is an acceptable alternative to in-person testing done in a lab.

Due to the pandemic stemming from COVID-19, the authors found themselves in a position of wanting to teach students how to conduct usability tests, but unable to have students come to campus to perform these tests in traditional settings. This paper therefore details what the authors did to teach students how to oversee usability testing remotely, how such practices were implemented in the classroom, and what the authors learned along the way. The paper begins by providing a discussion of the human-computer interface design course through which students have been taught about usability testing in the past, and then details how two sections of the course supported remote usability testing. It will end with an analysis of the lessons learned from each implementation. There has been little reported in the literature about how to teach students to conduct usability testing in distance learning environments. This paper is an attempt to begin filling that gap.

2 Human-Computer Interface Design Course Structure

Human-Computer Interface Design is an interdisciplinary course in the Computer Science department at University of the Pacific [5]. The course is taught to upper division undergraduate students, both majors and non-majors, and covers the field of human-computer interaction (HCI). The two authors have been the only instructors of the course at the university for the past fourteen years. During the course, student teams develop a working prototype for some application, then conduct usability testing on the prototype and develop a final report that describes the team's findings regarding the usability of the prototype. In semesters before COVID-19, students performed these usability tests in face-to-face environments, with students participating as subjects in each other's tests. However, as described previously, in the spring and summer 2020 sections of the course, the instructors were forced to develop procedures to allow students to run these usability tests remotely. The following section describes how the authors managed the course in different terms to support usability testing.

2.1 Term 1 - Spring Semester

In the spring 2020 semester, the primary author taught the course for the first time. Since the course is open to non-majors, the instructor wanted to make sure to democratize the creation process. Therefore, the course initially focused on need-finding and in having students develop prototypes using a WYSIWYG rapid prototyping program called Axure, which has been lobbied to be used in HCI classes [12]. At the university where the authors teach, the decision to move to online classes due to the COVID-19 pandemic was not made until the spring break period. Once the decision came, the instructor made the decision to have all student groups continue using Axure for their final projects. Continuing the use of Axure meant that all groups created interactive sites that were hosted on Axure's servers and could be accessed by anyone via a URL. With 36 students in the class, the class was divided into nine groups of four for the final project. Students had roughly six weeks to build their prototype, run their user studies and then analyze and write their final report.

Nine prototypes were constructed for the class based off of ideas from other students on particular problems that they wanted to address, such as raising awareness of campus events, splitting bills with others, and building an aid for novice players of Dungeons & Dragons. When it came time to conduct usability studies, previous sections of the course followed one of two options. The first option reserved a class period (or two) to run the studies, and students would float around and try out different applications while in the same room. The second option tasked students with finding other participants typically around campus and scheduling times to meet outside of class. In this spring section, students were only asked to run studies with their peers from the course.

To provide each group with an equal number of participants each of the nine project groups was assigned to one of three "mega-groups", which consisted of twelve students each. From there, the groups were provided with a chat channel and with an online spreadsheet where they outlined their availability. Students then scheduled times where they and a partner from their group could meet synchronously with two members from one of the other teams in their mega group via the video conferencing platform Zoom. These Zoom meetings were further broken down into four sessions, each session was devoted to one of the four members serving as the "user" for the other team's prototype.

During each session, the user for the study would share their screen and a URL would be provided for them to access the other prototype. The two representatives from the other team split responsibilities of observing in real time the user, as well as running the user through their study. Each of these sessions was recorded and uploaded for the entire team (and potentially teaching staff) to analyze afterwards. In order to enhance validity, the groups in Zoom used two breakout rooms during their agreed study time, one as a video-conference waiting room (for the one person not involved with the user study at the moment) and the other to conduct the video-conference study. Students were provided with graphics in a set of slides to help them understand the process that they needed to take, (using two famous teams of four, Ninja Turtles and Teletubbies), to help illustrate how the sessions would run and what each member would be doing during each session.

At the end of each session, users filled out a general class survey online that asked them to rate the prototype based on 11 questions using a 5 point Likert scale. Groups were tasked with running all eight other members of their mega group through their remote usability study, and so they wrote a report which analyzed those eight interactions. As part of the report, each group provided URLs for the eight videos that they had recorded. To encourage each group to follow through with running the studies remotely, each group was incentivized with extra credit should they be able to interview all eight of the other participants.

2.2 Term 2 - Summer Semester

Human-Computer Interface Design was also taught as an intensive five-week course during the summer of 2020. Unlike the spring 2020 offering, the instructor (the second author) knew at the beginning of the term that the course would be taught through distance learning. The course met four times per week through WebEx (except for holidays and exam days). Due to the compressed nature of the course student teams had much less time to prepare and evaluate their prototypes, than did the students who took the course in the spring. Thus, the instructor made the decision to allow student teams to develop their prototypes using whatever platform and development environment was most familiar to them, rather than require all teams to learn and use Axure. Twelve students were enrolled in the course and divided into four teams, with each team developing a functional prototype. Prior to usability testing, each team was asked to create an executable version of their prototype and upload it to the course management system.

During the last week of the five-week term, one two-hour class period was devoted to conducting the usability tests. The period was divided into six 20-minute sessions. For each session four students played the role of the user for another team's project, while the remaining students acted as the evaluation team for their own project. The instructor created four "breakout sessions" in WebEx (one for each project team) and manually added the user and project team members to each session. As was the case in the spring 2020 course with Zoom, the breakout sessions in WebEx provided a private virtual usability lab in which the user could use the prototype with only members of the evaluation team being able to observe [10, 3]. Each project team had six total participants act as users of their prototype, and each student was able to play the role of the user for two of the other team's projects. The instructor prepared the schedule for the usability tests in advance of the class period and posted it in the course management system. Thus, each student knew for which projects they were to be the user, and they were asked to download the appropriate prototypes in advance of their usability testing sessions.

During each session, users were asked to share their screens, unmute their microphones, and talk aloud as they attempted to perform tasks with the prototypes. In this way members of the evaluation teams could see what users were doing and hear what they were thinking as they interacted with the prototypes. Evaluation team members took notes and recorded data about the user's experience using the prototype. At the conclusion of each session, users completed a user satisfaction questionnaire prepared by the evaluation team that asked about their opinions and suggestions for improvement of the prototype.

3 Lessons Learned

Overall the experience the instructor gained from the spring term is that the remote usability study worked surprisingly well. Most of the students did not find any particular complaints about the study or the process that was undertaken. Initially the instructor was worried about sampling and availability, that the students would run into problems recording or that they would not be able to interview all students due to scheduling constraints and a limited time frame. However, with a small bit of nudging, all teams were able to run all eight of their participants. Having a synchronous classroom or a time when all students could meet would certainly make it easier for all involved, though one may also run into problems related to participants not showing up, even if the class is mandatory, so the instructor was relieved that the pressure and the ability for students to meet on their own did not seem to cause many issues.

Table 1: Number of Responses mentioning specific words

Word	Responses
work	16
online	7
axure	5
reading	5
quiz	4
zoom	1
remote	0

To gain insight for lessons learned, for the spring term the instructor reflected on the class and reviewed course evaluations. The majority of the students (24 out of 36) filled out course evaluations, with most respondents providing comments or suggestions for improvement with respect to the course and the instructor. The evaluation contained 15 responses in each of the two areas for a total of 30 responses. The responses were read multiple times and categories were devised and subsequently analyzed for the course by looking for feedback mentioned by students on improving the usability study portion and the course in general. Table 1 provides some of the words that were searched and the number of times that such responses were found in the analysis. In the analysis and reflection, the biggest comment about the course was the workload (work being mentioned in 16 out of the 30 responses, and workload being categorized in the responses 14 out of 30 times). While most of the responses were about the abrupt transition to online and workload, more importantly none of the reviews mentioned aspects of running the remote usability study.

Upon further reflection, the instructor believes that most of the grievances for the course from the students were based on readings, workload and the prototyping software that was used. Five out of the 30 responses mentioned Axure, most of the responses were negatively aimed at Axure as they mentioned "Axure is not designed to make a fully functioning application, yet we were expected to make fully functional apps in them". At the same time another student mentioned that Axure made certain things easier to do, "I think for the final project it took a lot of the pressure off to allow us to use only Axure." Even one of the students who had a very detailed critique of the course, Axure and the instructor wrote: "Of course, he did give us a choice to just spend the first two weeks working in Axure and then spend the last two weeks doing it in our intended language, but that wasn't really much of a choice. Two weeks to do a project intended for five, in addition to doing it first in Axure? Give me a break.” This was true, the instructor mentioned to students that they could switch for the final two weeks, but all groups decided to stay with using Axure. In this last comment, the student is highlighting an advantage of using Axure for a course with short deadlines and covering a broad area of HCI. In such an introductory, survey course, using Axure will allow students to more quickly create prototypes where one can run a user study remotely without many up front issues. Needing only a URL improves security for other users as they do not need to install or run unknown applications to test prototypes [4]. While it is to be decided if Axure would be required to be used in future courses, for the spring term course, the instructor acknowledges that to combat some of the additional challenges presented in setting individuals up for remote usability studies, the instructor should do a better job of framing the course objectives and in providing students more resources or labs that would allow them to lessen the amount of work needed to create a prototype.

While the instructor was already familiar from interactions with students about the workload and about Axure, the instructor encountered a single issue with students conducting remote usability tests during the course. This includes issues with running Zoom, and of any technical difficulties that could arise with having a completely new product. As an aside, this one difficulty was not mentioned in the course evaluations; the course evaluation remark about Zoom recorded in Table 1 refers to a comment where a student was pleased with the instructor’s availability via Zoom. Instead, the one Zoom issue was found from searching the chat logs for the course (the class used slack). The chat logs contained the one instance where a student contacted the instructor thinking that their recording of their user study was overwritten, but it turned out that the recording was simply in a different folder. The instructor often checked in with the students and does not remember any other problems, nor does the instructor have any mention of additional problems mentioned through their chat logs from the class about technical problems related to their studies. Throughout the course there is no recollection of hearing of any issues surrounding test participants being able to access the website, or that the software did not work on another person’s machine, but rather the complaints were about not being able to produce something that felt like fully functioning software. In a remote environment, teaching developers how to identify relevant tasks for users to engage in, evaluate said interfaces, and gain insight from those interactions to identify problems provides a worthwhile tradeoff to using Axure’s limited environment. Future instructors therefore, should frame the class as more interdisciplinary from the very beginning and limit the scope and workload of what students are creating. This keeps the course’s interdisciplinary focus [5] at the forefront throughout, and nudges students to focus on generating questions and potential solutions that would allow them to gain insight into the process and promise of usability testing, even when done remotely.

The instructor of the summer 2020 section was also quite pleased with how well remote usability testing worked in the course. The primary challenge was that, even though project teams were asked to provide prototypes that could run on both a PC and a Mac, not all users were able to successfully run the software provided by the project team on their own computer. Clearly, instructors must weigh the advantages and disadvantages of using a prototyping program such as Axure when deciding whether to allow teams to choose their own platforms for developing projects. In hindsight, given the compressed time frame of summer sessions at the authors' university, perhaps requiring use of Axure would have been a good choice since teams rarely have time to create fully functioning applications during the summer sections of the course.

4 Conclusion

In this paper the authors described two sections of a human-computer interface design course that taught students about usability testing in distance learning environments. Both Zoom and WebEx were explored as platforms for conducting remote usability testing, and both were found to be effective. Students participated as subjects (users) in each other's usability tests. In both sections breakout sessions were used in Zoom and WebEx to create private virtual usability testing spaces, through which users could share their screens with the development teams while trying to use their applications. Both authors found this approach to be effective for teaching students about usability testing. In the future the authors would like to conduct a more detailed comparison of face-to-face and remote usability testing in educational settings, perhaps allowing students to experience both environments and collecting their opinions on each.

References

[1] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. What happened to remote usability testing?: an empirical study of three methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07*, pages 1405–1414, San Jose, California, USA, 2007. ACM Press.

[2] José C. Castillo, H. Rex Hartson, and Deborah Hix. Remote usability evaluation: can users report their own critical incidents? In *CHI 98 conference summary on Human factors in computing systems - CHI '98*, pages 253–254, Los Angeles, California, United States, 1998. ACM Press.

[3] Kapil Chalil Madathil and Joel S. Greenstein. Synchronous remote usability testing: a new approach facilitated by virtual worlds. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, page 2225, Vancouver, BC, Canada, 2011. ACM Press.

[4] Kapil Chalil Madathil and Joel S. Greenstein. An investigation of the efficacy of collaborative virtual reality systems for moderated remote usability testing. *Applied Ergonomics*, 65:501–514, November 2017.

[5] Daniel C. Cliburn. An Interdisciplinary Course on Human-Computer Interaction. *J. Comput. Sci. Coll.*, 33(2):38–45, December 2017. Place: Evansville, IN, USA Publisher: Consortium for Computing Sciences in Colleges.

[6] Susan Dray and David Siegel. Remote possibilities?: international usability testing at a distance. *interactions*, 11(2):10, March 2004.

[7] Erik T. Nelson and Angelos Stavrou. Advantages and Disadvantages of Remote Asynchronous Usability Testing Using Amazon Mechanical Turk. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 55(1):1080–1084, September 2011.

[8] Jean Scholtz. Adaptation of traditional usability testing methods for remote testing. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, page 8, Maui, HI, USA, 2001. IEEE Comput. Soc.

[9] Helen Sharp, Jennifer Preece, and Yvonne Rogers. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley and Sons, Indianapolis, IN, 5th edition, May 2019.

[10] Katherine E. Thompson, Evelyn P. Rozanski, and Anne R. Haake. Here, there, anywhere: remote usability testing that works. In *Proceedings of the 5th conference on Information technology education - CITC5 '04*, page 132, Salt Lake City, UT, USA, 2004. ACM Press.

[11] Tom Tullis, Stan Fleischman, Michelle McNulty, Carrie Cianchette, and Margaret Bergel. An empirical comparison of lab and remote usability testing of web sites. In *Usability Professionals Association Conference*, July 2002.

[12] Hongyan Xu, Zhiming Wu, Tao Lin, Ningjiu Tang, and Yu Chen. Research on teaching model of project-driven hci course. In *International Conference on Education, Management and Computing Technology (ICEMCT-16)*. Atlantis Press, 2016.

Effectiveness of Online Teaching Paradigm on Learning Experiences in Database Courses*

Mohammad Amin, Pradip Peter Dey and Bhaskar Raj Sinha
National University
San Diego, CA 92123
{mamin, pdey, bsinha}@nu.edu

Abstract

In the post-COVID 'new normal', people are likely to live and learn in new ways. It is expected that there will be demand for more virtual activities in all aspects of life. Currently, most academic institutions are changing the way they offer their degree programs. New challenges and opportunities for robust online educational environments need to be critically examined. Many educators are now striving to learn how to teach online. National University started offering online programs more than twelve years ago. The authors of this paper have engaged in online teaching activities since then. In the early stages, many educators were reluctant to adopt this new method of education and they raised questions about the quality of online education. This article addresses this issue with evidence and compares it with the assessment data obtained by the university at the end of database classes taught between 2011-2016. The outcomes of this research indicate that the students at NU who took the database classes during 2011 to 2016 preferred online classes and learned either equally or better in online database classes than the onsite database classes. Suggestions are made about future research that may contribute towards generalizing these trends in other areas of computer science education with opportunities to serve the rapidly increasing online student population.

1 Introduction

In a rapidly changing educational landscape due to coronavirus (COVID-19), alternatives to traditional onsite education are being frantically examined worldwide. One of the alternatives is a robust online system that performs as good or better than the onsite system. Online educational systems are not new; they existed mostly as a secondary option with the primary option being onsite education at prestigious universities. Today, every university is forced to explore alternatives to onsite education with a renewed interest in examining the scope and limitations of the alternatives. This paper makes contributions towards building confidence towards offering online courses in a highly technical area which may allow periodic enhancements with adequate reflections. It analyzes the performances and responses of students who took onsite and online offerings of a Database Design and Implementation course with the same instructor and content materials, during 2011-2016 at NU [www.nu.edu]. NU is a nonprofit academic institution with the goal of providing accredited quality education, service, and support to a diverse mature student population in a convenient learning environment. At a very early stage, this university recognized the importance and need of online education and started offering online degrees. In 2019, approximately 70% of its students participated online and all programs were made ready for online delivery. Students at NU were given the choice to join their classes either on-site, online, or in a hybrid format. The NU faculty in the Department of Engineering and Computing (DOEC) are continually engaged in exploring effective teaching methods to deliver quality instructions and materials to its students in the real or virtual classrooms for efficient and successful long-term and short-term learning. Today, at NU, online education has become equally or more popular because of its quality, accessibility and affordability. Recently, due to the COVID-19 pandemic in the year 2020, most universities and colleges are offering their programs online. Proactively, NU has made significant innovative improvements on both technology platforms and pedagogy for its online offerings.

Based on a research conducted from 1996 to 2008 on medical training in higher education the US Department of Education has published a report [8] concluding *"···online learning has been modestly more effective, on average, than the traditional face-to-face instruction with which it has been compared"*. Babson Research Survey Group has published a report based on a survey of 2800 universities and colleges. It was concluded that the demand for online education is on the rise and quality is improving with the advancement of technology, more faculty engagement, and administrative support [2]. Literature mentions that the use of WhatsApp social networking among students has a great positive impact on their learning [4]. In the past, community college students preferred to take "easy" academic subjects online and "difficult" or "important" subjects from onsite classrooms [7]. Another study observed that computer science students learned better in a blended learning environment than in the online and onsite learning environment [6]. Illinois Online, offering many online degrees, claims five major benefits of online educations: 1) Career Advancement, 2) Flexible Schedule, 3) Cost Effective, 4) Self-Discipline and Responsibility, and 5) More Choice of Courses [9]. As the World Health Organization (WHO) officially declared the COVID-19 pandemic, teaching and learning has brought many challenges and has opened up many opportunities to academicians. This paradigm shift to online education is now quite obvious and needs all educational institutions worldwide to analyze and adopt this change. NU is currently offering all its courses online and for most of the courses and may continue this in the future. The Online Learning Consortium has recently developed a Faculty Playbook to provide resources and guidelines for development of online course materials and delivery methods [5]. The authors have also published a number of reports on online teaching and learning analyzing the effectiveness of this academic paradigm [1, 3, 10, 12, 11, 13, 14, 17, 16, 15].

2 Case Study

Over the past 20 years, the authors have been teaching a variety of engineering, computer science, and technology graduate and undergraduate classes, including database courses in both online and onsite modalities. These instructors continuously strive to provide quality instructions to students by using a variety of different instructional modes, and reviewing students' assessment feedback in order to facilitate better learning outcomes. They have collaborated with other educators and published articles/papers to present their teaching experiences and teaching related research findings [1, 3, 10, 12, 11, 13, 14, 17, 15]. For online classes, NU had earlier adopted the eCollege delivery platform with ClassLivePro. Currently, the university is using Blackboard with Collaborate Ultra. Recently, the university started providing the same version of the online course shell for every onsite class. This change has improved, consistency of courseware content for instructors for both onsite and online delivery, and improved student learning as instructors have more latitude to focus on multiple avenues of delivery through multimode teaching methods. Instructors strive to present a well-organized and concise discussion of the course material to help students 1) understand new concepts, 2) apply their knowledge effectively, and 3) "think out of the box" , to identify and solve new problems. In this article, a case study is considered that deals with a database course offered both online and onsite. Fourteen database classes (seven onsite and seven online) are considered for this analysis.

In online classes, the instructor used a Microsoft Surface Book (Tablet) for free-hand writing and drawing required for all database classes. Complex problems were solved in multiple ways step-by-step, showing students the tablet desktop using Blackboard Collaborate Ultra. This enabled students get the benefits of an onsite class during these online sessions. The real-time collaborative nature of an onsite classroom were replicated, enhancing the student learning experience. These dynamic two-way online sessions, as opposed to using static PowerPoint presentations, for these conceptually intense database classes, were a significant value-add. The average GPAs for all the database classes analyzed for this research were within the university target. The instructor gave two two-hour lectures per week using a microphone through Collaborate in the Blackboard and wrote important/critical notes on the HP tablet screen with a smart pen and shared the tablet desktop with the students. Students could see all the classroom activities on their computers, listen to lectures using earphones and ask questions through the microphone or chat texts. All these instructional activities were recorded and saved in eCollege and Blackboard. These recorded sessions were made available 24/7 to the students for download and replay at the students' convenience. Instructor responses to student inquiries/questions were prompt and he proactive. Students could also contact the instructor through email, phone, and in-person during office hours –just like the onsite students.

At NU, an end of course survey is offered and all students are expected and encouraged to participate. The survey form used by NU has a total of 22 questions in three major areas: 1) Student Self-Assessment of Learning- 7 questions, 2) Assessment of Teaching- 12 questions and 3) Assessment of Course Content- 3 questions. This survey form also allows students to write their comments. Survey data is analyzed by the NU Office of the Institutional Research and Assessment (OIRA). A summary of the survey data with student comments are made available to the respective instructors, department chairs, and school deans for review and comments. The authors of this paper always review their course survey reports from OIRA carefully and adjust their future teaching plans by improving their pedagogical approach to teaching and adopt different or modified multimode techniques as appropriate.

3 Results and Discussion

Summary of the data for the recent seven onsite and seven online DAT604 Database classes are shown in figure 1. These results indicate that the instructor is consistently receiving high scores on teaching. Table 1 depicts the average values of learning and teaching for DAT604 onsite and online classes during 2011-2016 academic years. These indicate that the teaching performances of the instructor are excellent for both online and onsite classes and students are also learning from the online classes are same or better.

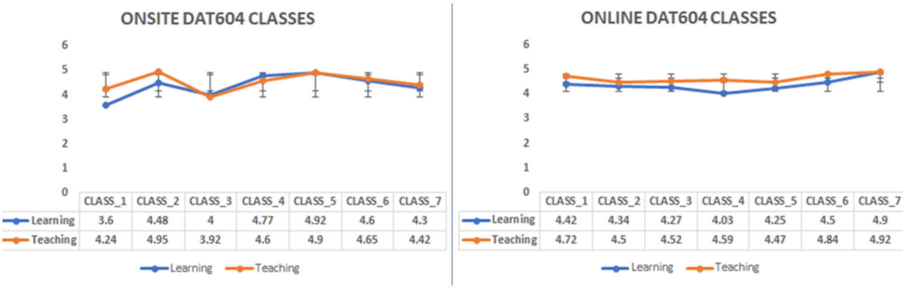


Figure 1: Graphical Representations of Learning and Teaching for DAT604 Onsite and Online Classes during 2011-2016 Academic Years

DAT604 CLASSES	Average of Learning	STDEV of Learning	Average of Teaching	STDEV of Teaching
Onsite	4.38	0.46	4.53	0.37
Online	4.39	0.27	4.65	0.18

Table 1: Average Values of Learning and Teaching for DAT604 Onsite and Online Classes during 2011-2016 Academic Years

Students consistently expressed appreciation for the approach, teaching skills, and care of the instructor. Many students explicitly stated on their comments that instructor did an excellent job, explained the topics very well, and presented difficult materials in a simple and understandable format. Instructor used different methods and technologies during the classroom presentations including audio, video, graphic, animated simulations, handouts, and appropriate references. Classes were very organized, structured and well managed. Class time was utilized wisely and interactively with hands on activities. Instructor asked higher level challenging questions to students and expected well thought out answers. Instructor also encouraged students to ask questions and answered them carefully. Instructor provided a very conducive learning environment where students could get deeply involved and engaged in the class with both instructor and other fellow students.

4 Conclusion

From the graphs in figure 1, it is concluded that learners experience basically the same perceived benefits of database classes from both online and onsite offerings. The Grade Point Average (GPA) for all classes in this study was approximately 3.25 (grade: B+) indicating appropriate rigor. The number of students enrolled in each of the classes were about 25. From the graphs it is observed that the online teaching and learning curves are steadier than the onsite. Standard deviation for the onsite mode of delivery was higher than the online classes. Finally, it can be concluded that the students get the same or better perceived database learning experience through online classes.

Our review and reflection on the study of these courses suggest that opportunities for online education can be made attractive to students given adequate attention to student needs. The outcomes of this research contribute towards NU's current and future to generally include online offerings of all courses to deal with the challenges emerging from the "new normal" environment due to coronavirus pandemic. Future research may be explored for contributing towards developing robust online learning systems that may be available to a wide spectrum of computer science learners in both synchronous and asynchronous modes. Currently, these authors are focusing on and evaluating the Asynchronous Online Teaching and Learning Techniques. Due to the challenges of the COVID-19 pandemic, onsite face to face classes are becoming more difficult. As a result, the demand for asynchronous academic education, Precision Learning, Differentiated Learning, and Competency Based Education is increasing. It has now become essential for every educator to experiment with what to deliver and how to deliver in an online modality. Future research opportunities may include potential contributions of artificial intelligence (AI), virtual reality (VR), augmented reality (AR), intelligent tutoring systems, and other emerging technologies towards the new developments of e-learning. Even if the impact of COVID-19 pandemic ends rapidly research on potential engagement of students in e-learning with new technologies should continue in order to provide various options to students with a focus on student success.

5 Acknowledgement

The authors are thankful to the NU administrators, staffs, and students for their sincere support and cooperation during this research.

References

[1] A Abu-Arafah, M Amin, and M El-Eidarous. Distance learning and technology in the Kingdom. In *Proceedings of A Millennium Dawn in Training and Continuing Education Conference (MDTCE)*, 4 2001.

[2] I Elaine Allen and Jeff Seaman. *Changing course: Ten years of tracking online education in the United States.* ERIC, 2013.

[3] M Amin, A Abu-Arafah, and M Al-Eidarous. The use of multimedia in a technical college classroom in the Kingdom. In *The First Saudi Technical Conference and Exhibition*, STCEX2000, 11 2000.

[4] Aicha Blehch Amry. The impact of WhatsApp mobile social learning on the achievement and attitudes of female students compared with face to face learning in the classroom. 2014.

[5] Online Learning Consortium. Delivering high-quality instruction online in response to COVID-19. `https://onlinelearningconsortium.org/tools/delivering-high-quality-instruction-in-response-to-covid-19-faculty-playbook/`. Accessed 2020-06-10.

[6] Meltem Eryilmaz. The effectiveness of blended learning environments. *Contemporary Issues in Education Research (CIER)*, 8(4):251–256, 2015.

[7] Shanna Smith Jaggars. Choosing between online and face-to-face courses: Community college student voices. *American Journal of Distance Education*, 28(1):27–38, 2014.

[8] Barbara Means, Yukie Toyama, Robert Murphy, Marianne Bakia, and Karla Jones. Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies. *US Department of Education*, 2009.

[9] University of Illinois. Illinois online. `https://online.illinois.edu/`. Accessed 2020-01-05.

[10] J Reeves, M Amin, and B Sinha. Enhancing student learning in online and on-site engineering courses. Academic Forum, National Instruments Week Worldwide Graphical System Design Conference, 2012.

[11] J Reeves, B Sinha, A Davis, and M Amin. Innovations in remote laboratories and simulation software for online and on-site engineering students. In *Proceedings of the 120th ASEE Annual Conference and Exposition*, pages 23–26, 2013.

[12] Jodi Reeves, Mohammad Amin, Marcos Turqueti, and Pradip Peter Dey. Improving laboratory effectiveness in online and onsite engineering courses at national university. *Journal of Research in Innovative Teaching*, 5(1), 2012.

[13] B Sinha, P Dey, H Badkoobehi, J Jaurez, M Amin, and Romney. G. Best practices for the online education environment. In *the World Scientific and Engineering Academy and Society (WSEAS) 9th International Conference on Computer Science and Information Systems*, pages 57–62. Athens Institute for Education and Research, 2013.

[14] B Sinha, J Jaurez, P Dey, , and M Amin. Online course design to promote reflective and critical thinking in learners. In *Proceedings of the International Conference on Learning and Administration in Higher Education*, ICLAHE 2013, pages 575–583, 5 2013.

[15] R Uhlig, B Sinha, S Jawad, P Dey, and M Amin. Enhancing student collaboration for improved learning. *Journal of Modern Education Review (JMER)*, 7(8), 2017.

[16] Ronald P. Uhlig, Shatha Jawad, Pradip Peter Dey, Mohammad Amin, and Bhaskar Sinha. Enhancing learning experience in online courses through enriched responsiveness. *International Journal of Advanced Engineering and Management Research (IJAEMR)*, 4(3), 2019.

[17] Ronoald Uhlig, S Jawad, P Dey, M Amin, and B Sinha. Enriching responsiveness to enhance student learning in online courses. In *Proceedings of the 2018 Hawaii Universities International Conferences on STEM/STEAM, Honolulu*, 2018.

Experiences Offering an Online Version of Computer Science Support (Peer Tutoring) to Undergraduate Computer Science Majors in the Era of COVID-19*

Karina Assiter
Computer Science Program
Landmark College
Putney, VT 05346
`KarinaAssiter@landmark.edu`

Abstract

This paper presents the experiences that we had offering an online version of Computer Science (CS) Support (peer tutoring) to undergraduate computer science majors in the era of COVID-19. It describes the online CS Support format, student sessions, strategies for promotion, challenges and finally proposals for improvement.

1 Introduction

At our institution, an undergraduate college for students with learning challenges such as ADHD, dyslexia and ASD, support for computer science majors is in the form of peer-tutoring: hired students generally available in a lab room from 6 –10 pm, 6 nights a week. This resource is a critical component of the success of our programs as students generally have full daytime schedules (classes are 75 minutes long versus the standard 50 minutes) and have time primarily during the evening hours (when instructors are generally not available) to actually work on their programming projects. Many of our students with learning challenges need a lot of guidance and clarification of assignment instructions, so they rely heavily on CS Support.

When learning transitioned to an online format in March 2020 (after COVID-19 started to have an impact in the US) the author, generally involved in recruiting students for CS Support and creating the staff schedules, had to determine whether (and then how) to continue to provide this essential service.

In the rest of the paper we present related work, our methodology, challenges faced, results and then proposed future work.

2 Related Work

Necessity required creating online CS Support quickly in an ad hoc fashion using the resources available, all while also learning the tools of, and restructuring our courses for, online teaching. Thus, we did not have time to review, let alone implement, best-practices [4]. After the fact, in doing searches for related materials, the keywords that produced the most relevant results were "online tutoring" or "online academic support".

By now, there are many successful online programs [9] [2] and all those that we have reviewed so far seem to provide some form of online tutoring. There are many models for offering online tutoring, including in house [1], provided by professionals, external services [6] or undergraduate/graduate TA's [3] [7], per single course or across all courses in a program/institution, by appointment or drop-in (on demand) [6], synchronous or asynchronous. The model selected depends on the institutions needs and resources. As we will show, the model that we selected for providing online tutoring closely fits both our face-to-face model of CS Support as well as our institutions' available resources.

3 Methodology

Our CS Support model of tutoring is in house, undergraduate led, drop-in, one staff person assisting (generally) one student at a time, synchronous and across all courses in the computer science program. Thus, that is the format that we emulated in the online environment.

As part of our methodology, we present the format of online CS Support, how it was promoted to students and the details of CS Support sessions.

3.1 Online CS Support Format

As with most online formats CS Support had to have a virtual meeting space. The tool that our institution sanctions for online meetings is Microsoft Teams [8]. Thus, we first created a Computer Science Support (CSS) Team in Microsoft Teams (see Figure 1). We then added as members all CS Support staff and computer science faculty. Students from cs courses (would-be customers

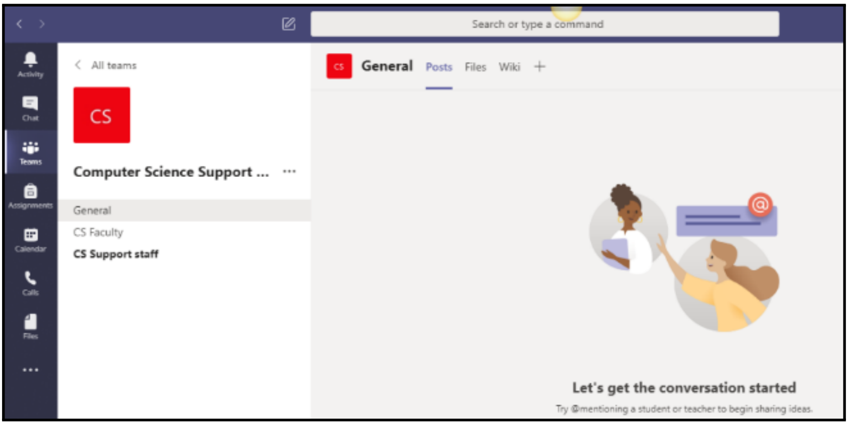


Figure 1: Computer Science Team in Microsoft Team.

to CS Support) were added later when it was determined that would be the best way for them to find the CSS team (unfortunately, they had to be added individually from course class lists).

In addition to having the default *General* channel that Microsoft Teams provides[1], we created two additional channels: one for the computer science faculty and another for CS Support staff. The latter channel was where staff would show up for their shifts and where students (customers) would then arrive seeking assistance (see Figure 2).

The CS Support schedule was in the format of a simple spreadsheet saved as a file in the CSS Team. Figure 3 shows the contents of the open file and then Figure 4 shows the file listed under the Files tab in the CCS Team.

3.2 Promotion of CS support

Once we had established the basic CS Support framework, we then had to determine the best way to advertise both the staff schedule as well as a link to

[1]In Microsoft Teams, channels are where an organizational subset of a team can gather and communicate.

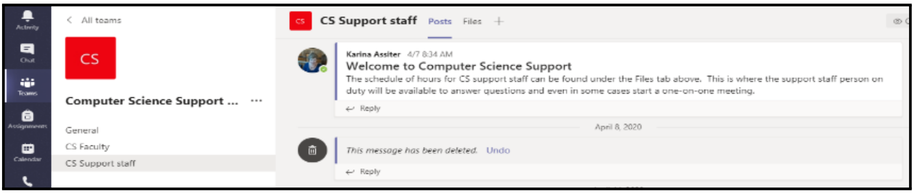


Figure 2: CS Support staff channel.

Computer Science Support Schedule

Spring 2020

Location: Computer Science Support Team (Microsoft Team)

Link to channel: CS Support Channel

Day	Hours	Staff Person
Monday	6:00 - 8:00 PM EST	Trey Stevens
Tuesday	6:00 - 10:00 PM EST	Reece Rountree-Hanscom
Thursday	6:00 - 10:00 PM EST	Sam Mortinger
Friday	6:00 - 8:00 PM EST	Trey Stevens

Figure 3: Computer Science Support Schedule.

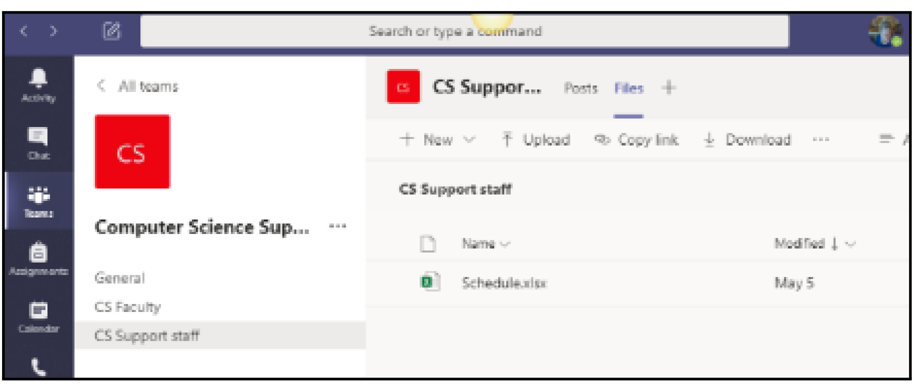


Figure 4: Schedule in files tab.

the CSS Team. The first step was to inform the CS Support staff about the planned process (Figure 5).

Then, we sent emails to all cs faculty with a link to both the schedule, as well as the channel where students should arrive for assistance. A similar email was sent to students in the authors' CS2 and Data Structures courses (Figure 6). In addition, the contents of the emails were posted as an announcement in both courses on Canvas [5] (our institutions Learning Management System).

All students in the authors CS2 and Data Structures course were added manually to the CS Support team; ideally, Microsoft Teams could have provided a feature for inputting lists of names in batch. We then added other students to the team that we knew to be in the computer science program (feasible, as we are a small college with a small computer science program). In addition, all advisors were informed about online CS Support; they were asked to encourage their advisee's to attend.

Subject: Finalized schedule and Process

Hey all,

After hearing from the cs support staff members (or not) I have put together the cs support schedule and placed it under the 'files' tab in the **Computer Science Support Team** in Microsoft Teams: schedule. Note that I've also included a link to the channel where students can find you (the team should be public and hopefully there are no issues).

I plan to publish this schedule tomorrow morning by sending to CS faculty and my classes. Hopefully all goes according to plan (we're all learning as we go).

Thanks everyone for being willing to do this; I know things are hectic right now!

Figure 5: Email to CS support staff about planned process.

3.3 CS support sessions

As mentioned previously, CS Support staff would show up in the *CS Support* channel and indicate, in some fashion, that they were on duty. Students who "showed up" could then request help and be invited into a video session with the staff person (Figure 7). The video sessions are similar to Zoom [10], with tools such as screen-sharing, chat, adding rooms, etc. At the end of their shift, the staff person would post a message indicating that they were going off duty.

Students in my Computer Science II and Data Structures courses:

Every time you attend a Computer Science Support Team session for help you can earn an EC **point** that will be applied toward past CBA's.

Here is a link to: schedule of hours which can also be found under Files in the team. **Put the dates and times on your calendar if you need reminders.** Forward this email to your advisors if you need even more help remembering.

I've added all of you to the Computer Science Support team already so you just need to show up in the CS Support staff channel during staff hours and let the person on duty know that you're there (just say, "hello, I need help").

Please let me know if you have trouble accessing the team, channel or a staff person.

Thanks!

Figure 6: Initial promotional email to all courses.

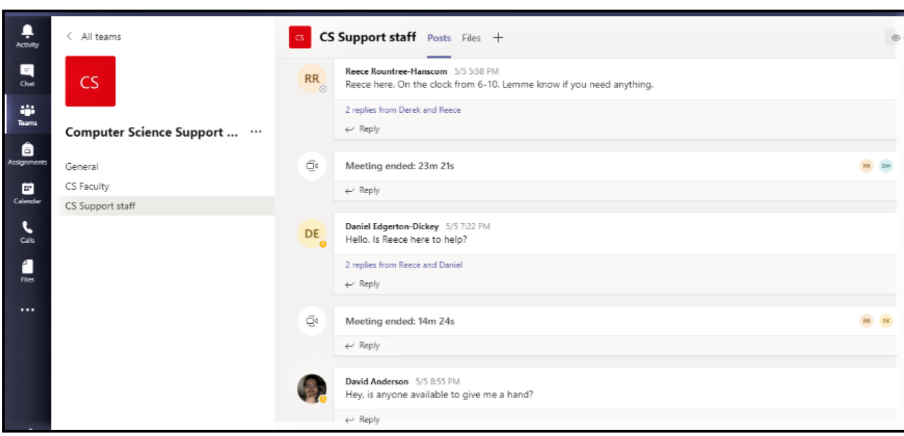
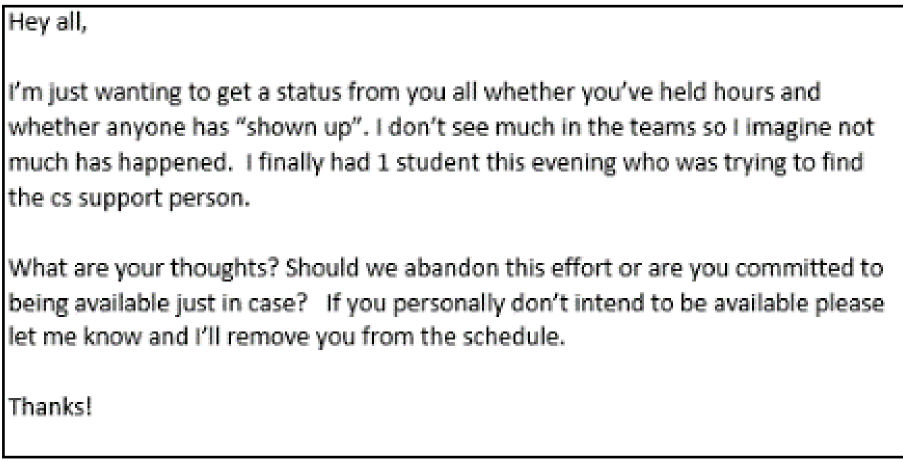


Figure 7: CS support session.

4 Challenges

Many of our students have executive functioning challenges so it took a while for them to get into the rhythm of online learning. Thus, because they were not, at first, working on their assigned programming projects, they were also not showing up for CS Support.

After the first week of online support we polled the CS Support staff to gage attendance and whether we should continue our efforts (Figure 8). The consensus among the CS Support staff was to give it one more week. The author



Hey all,

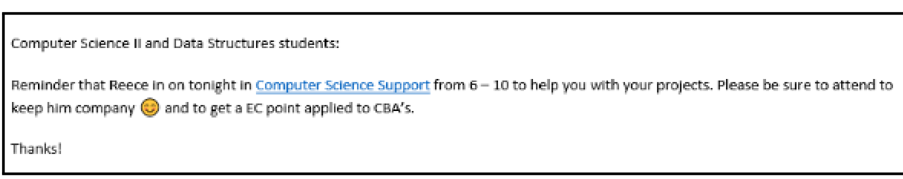
I'm just wanting to get a status from you all whether you've held hours and whether anyone has "shown up". I don't see much in the teams so I imagine not much has happened. I finally had 1 student this evening who was trying to find the cs support person.

What are your thoughts? Should we abandon this effort or are you committed to being available just in case? If you personally don't intend to be available please let me know and I'll remove you from the schedule.

Thanks!

Figure 8: Email to gage success of cs support.

then started to strongly encourage attendance at CS Support by offering students an extra credit point that would be applied to their class-based activities (CBA's). In addition, she started sending out reminders to her class lists on the days when there would be CS Support in the evening (Figure 9). Another



Computer Science II and Data Structures students:

Reminder that Reece in on tonight in Computer Science Support from 6 – 10 to help you with your projects. Please be sure to attend to keep him company 😊 and to get a EC point applied to CBA's.

Thanks!

Figure 9: Reminder about CS support that evening.

issue that may have initially made it difficult for students to find CS Support is that unless the person was already added to the CSS Team, they would not have seen it listed among their teams; we incorrectly assumed that if it was created as a public team (which it was) then it would automatically show up for everyone in the organization. This might have discouraged students who were told to go to the CSS Team in Microsoft Teams, but then could not see it listed among their Teams (and they either never received an email with a link, or they could not find the email). That is the point at which we tried to add as many cs students to the CSS Team as possible.

A final challenge was the CS Support staff, themselves, who are also students with learning challenges. Adapting to the online environment and then managing to show up for online office hours proved to be too overwhelming for a few of them. Before we transitioned to an online format, we had 5 staff members covering 6 evening shifts per week. One staff member was already missing shifts before the transition, so he was not included in the new online schedule. A second staff member informed us within the first week that online CS Support was not working (essentially, for him). Then a third staff member did not show up for an online shift and, when confronted, he indicated that his schedule was just too busy in the online environment. At that point we were down to two staff members, just as attendance was starting to pick up. Thus, we recruited an additional staff member from among those formerly employed in the colleges' Hackspace (which did not have an online alternative). With 1 staff member willing to work 2 shifts we were able to cover 4 nights a week (versus the usual 6).

5 Results

There are a few statistics about attendance that we can draw upon. Whenever a student receives assistance, staff are supposed to enter the details in a Google sheet (Figure 10).

The CS Support Log is an imperfect method of recording attendance. Sometimes when there are multiple attendees or students arrive back-to-back, staff do not get a chance to make a submission to the Google sheet; later they may forget to go back and enter the details for one or more students (they have admitted that this is sometimes the case).

After we went online the number of Google sheet entries decreased; however, some of the missing numbers could be gathered by evaluating communications in the CS Support channel (when a video chat ends the attendee's initials are listed). Figure 7 shows an example.

The result of our data gathering is shown in Figure 11. The graph shows CS Support attendance from the beginning to the end of the semester, broken down by week. Table 1 provides a key to the graph.

A few simple ways to compare attendance between face-to-face and online

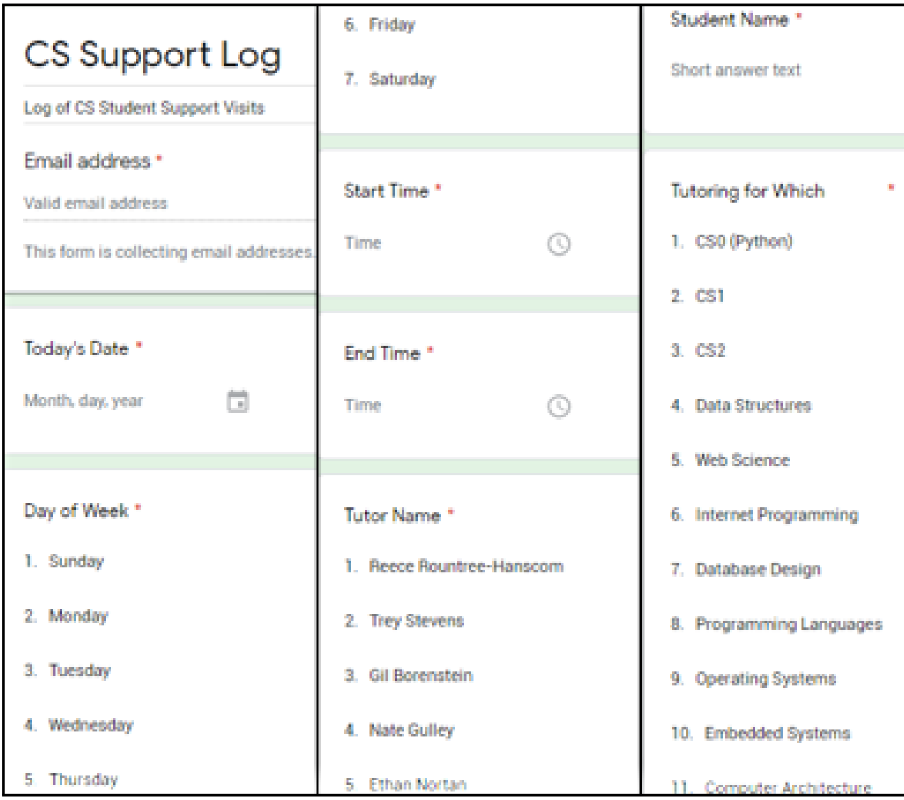


Figure 10: CS Support Log.

is to compare either the largest numbers (6/11 so 55%) or the totals (14/35 so 42% capacity).

Considering that the online format does limit both the number of students that can be assisted simultaneously and the amount of switching that can occur between students, the numbers are not as low as they might have been, considering our challenges. With strategies discussed in Future Work, we expect that these numbers could be improved.

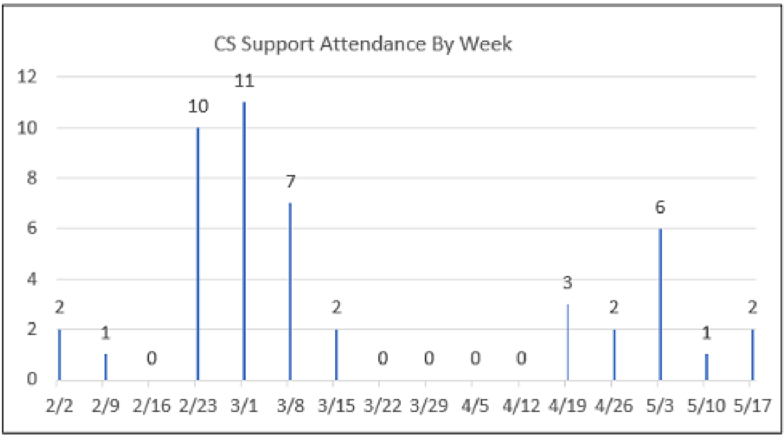


Figure 11: CS Support attendance by week.

Table 1: Explanations based on weeks

Week	Notes
2/2	Start of the semester (we have a January term)
3/15	Spring break
3/22	Additional week of no classes (due to COVID19) with faculty preparing for online teaching
4/5	Start to offer CS Support
4/12	Evaluated process, made alterations, publicized more. Addressed staff absenteeism (attendees may have lost confidence in process)
4/19	Started to have success
5/3	Last week of classes. Spike due to imminent deadlines
5/10	Finals week
5/17	Extension of semester; used by most faculty

What the graph does not show is that only a handful of students attended

CS Support (repeatedly). Also, all attendees of the online version of CS Support were in one of the authors' courses, most likely because she offered extra credit for attendance.

We would have preferred to also have student evaluation results (both qualitative and quantitative) for CS Support. In fact, there is a question on the student evaluations that is meant to evaluate student use of academic support for the class being evaluated, but it asks about attendance at DCAS, the Drake Center for Academic Support; since CS Support is not in the Drake Center students do not associate it with DCAS so there were no results for this question. Thus, we have not been able to use this as a tool to evaluate students' perceptions of CS Support.

6 Future Work

If we were to provide CS Support online in the future[2] here are the strategies that we would employ to improve both attendance and student and staff satisfaction with the process:

- Initially add students from all cs classes to the CSS team.
- Develop instructions for CS support sessions for:
  - Staff members : where and how to start and end their shifts.
  - Students : how to indicate that they have arrived for a session.
- Provide training materials (videos) for staff and students.
- Provide an initial online training session for staff.
- Integrate the use of Google sheet submissions into the instructions and training for CS Support staff.
- Add a question to the electronic student evaluation forms directly related to students' perceptions of CS support.

[2]Since students and CS support staff still will not be able to sit next to each other during a session, this may continue to be the model even when we return to campus in the fall.

References

[1] Joseph Cottam, Suzanne Menzel, and Janet Greenblatt. Tutoring for retention. In *Proceedings of ACM SIGCSE 2011*, SIGCSE '11, pages 213–218, New York, NY, USA, 2011. ACM.

[2] Nadine de Metz and Adele Bezuidenhout. An importance–competence analysis of the roles and competencies of e-tutors at an open distance learning institution. *Australasian Journal of Educational Technology*, 34(5), 2017.

[3] Paul Dickson, Tony Dragon, and Adam Lee. Using undergraduate teaching assistants in small classes. In *Proceedings of ACM SIGCSE 2017*, SIGCSE '17, pages 165–170, New York, NY, USA, 2017. ACM.

[4] The Academic Senate for California Community Colleges. Effective practices for online tutoring. `https://files.eric.ed.gov/fulltext/ED601995.pdf`.

[5] Instructure. Canvas. `https://www.instructure.com/canvas/`.

[6] Link-Systems International. Nettutor : Professional online tutoring 24 hours a day, every day of the year. `https://www.nettutor.com/`.

[7] Ramona Santa Maria and Banarjee Sarbani. Using undergraduate teaching assistants in computer courses. *JCSC*, 27(6), 2012.

[8] Microsoft. Microsoft teams. `https://www.microsoft.com/en-us/microsoft-365/microsoft-teams/group-chat-softwarehttp://www-cs-faculty.stanford.edu/~uno/abcde.html`.

[9] Computer Science Zone. The 30 best online bachelors in computer science degrees 2018 - 2019. `https://www.computersciencezone.org/the-best-online-bachelors-in-computer-science-degrees/`.

[10] Zoom. Zoom. `https://zoom.us/`.

Supporting Asynchronous Learners with Multiple Representations*

Pradip Peter Dey, Bhaskar Raj Sinha and Mohammad Amin
Department of Engineering and Computing
National University
San Diego, CA 92123
{pdey, bsinha, mamin}@nu.edu

Abstract

Mathematical models of computation such as Turing Machines are learned by most computer science students. This paper reports on experiments conducted for supporting asynchronous learning environments with multiple representations of Turing Machines for enhanced learning experience. Most learners appreciated multiple representations used for the asynchronous environment, and preferred multiple representations of Turing Machines even when the standard tabular representation was good enough for explaining fundamental aspects of information processing. 43 out of 48 (89.58%) learners preferred multiple representations of the Turing Machine over one representation in an asynchronous mode. The representations of the Turing Machine are presented on the following website: `http://www.asethome.org/mathfoundations/asynchronous/`.

1 Introduction

Recent studies suggest that the challenges of building resilient asynchronous learning systems can be overcome with innovative approaches and advanced technologies that may enable learners to thrive providing a varied sense of engagement in asynchronous mode [1, 17, 23, 25, 26, 27]. This paper reports on experiments conducted for supporting asynchronous learning environments with multiple representations of Turing Machines for modeling dynamic aspects of problems. Dynamic aspects of computation are often modeled with

Turing Machines (TMs) as suggested in popular books [2, 8, 24]. Alternatively, dynamic aspects of software can be modeled with state-charts which are simplified TMs proposed by David Harel [5, 6]. Mathematical models including TMs, two-stack Pushdown Automata (2PDA), Linear Bounded Automata (LBA), Pushdown Automata (PDA), Finite Automata (FA) and Non-deterministic FA (NFA) define the most elegant machines of computation. These models of computation define computability in clear terms and describe the scope and limitations of computation in revealing ways. There are some excellent textbooks on automata theory or theory of computation [2, 4, 8, 7, 9, 10, 11, 14, 19, 22, 24].

Alan Turing developed TMs for mechanizing aspects of mathematics in the form of algorithms [24]. TMs define the most powerful automata class for processing the most complex sets of languages, namely, recursively enumerable sets. The class of recursively enumerable sets properly includes all other computable sets [2, 8, 24]. A TM can be presented in a table, or drawn in a graphical form (transition graph), or coded in a string [2, 8]. These three presentations are equivalent in characterizing TMs. Any of these representations can be translated to any other without loss of essential information. An important question is are there representations that are better in helping learners understand models of computation in an asynchronous mode? This study suggests that understanding of essential features and critical thinking are promoted when multiple representations of computational models are provided with appropriate student engagement opportunities in asynchronous mode. Following the pioneering work of Rodger in the area of visualization of automata [21, 20], we consider the role of presentation forms for asynchronous learning. For supporting student engagement, we present visualization of a TM for {aba*} along with a tabular representation and an encoded string representation at the following website: `http://www.asethome.org/mathfoundations/asynchronous/`.

2 Background

One should not reject asynchronous learning systems without examining the potential contributions of artificial intelligence (AI), augmented reality (AR), virtual reality (VR), intelligent tutoring systems, and other emerging technologies towards new developments of asynchronous learning [1, 3, 12, 13, 18, 21, 20, 23]. Recent developments in intelligent tutoring systems suggest important progress towards providing various effective engagement opportunities in the learning environment [1]. Learning is a complex process which is gradually being explored utilizing new research in a number of disciplines including neuroscience, psychology, media, artificial intelligence, precision education, instructional design, cognitive science, sociology and other related areas [1, 3].

Our main goal is to examine learning experience of students in an asynchronous environment. Currently there is an emerging consensus that sensory data are more effectively perceived though multiple modalities than just one modality [1, 3, 12, 13, 18, 27]. Our research design is based on some recent studies on asynchronous learning [1, 23, 27] and our experience with computer science education.

3 Multiple Representations Of Turing Machines

Alan Turing developed a class of computational models in 1936, which have come to be known as TMs [25]. In the same year, Emil Post independently introduced algorithm machines that have come to be known as Post Machines (PMs) [16]. TMs and PMs are proven to be equivalent and their theory developed in 1930s and 1940s has provided the foundation of the theory of computation. TMs are the most popular models for recursively enumerable sets. Following Cohen [2], a TM, composed of six components, is defined as follows:

1. An alphabet, Σ , which is a finite non-empty set of symbols from which input is built.
2. A READ/WRITE TAPE, divided into a sequence of numbered cells; each of the cells contains one character or a blank, Δ . The input is presented to the machine one letter per cell beginning in the leftmost cell. The rest of the tape is initially filled with blanks.
3. A TAPE HEAD points to the current letter being read from the READ/WRITE TAPE. It can in one step read the contents of the READ/WRITE TAPE, write a symbol on the tape and move left or right one cell.
4. An alphabet, Γ , of symbols for the READ/WRITE TAPE. This can include symbols of Σ.
5. A finite set of states including one start state from which execution of the machine begins and some (may be none) HALT states that cause execution to terminate when entered.
6. A set of transitions from state to state where each transition has three elements: (Read-Letter, Write-Letter, Move).

The first element of the triplet is a letter read by the TAPE HEAD of the machine from the current cell. The second element is a letter written on the tape on the same cell where the first element was read from. The third element, Move, tells the TAPE HEAD whether to move one cell right, R, or one cell left, L. The HALT state cannot have any outgoing transition. To terminate execution on certain input successfully, the machine must be led to a HALT state. The input is then said to be accepted by the machine.

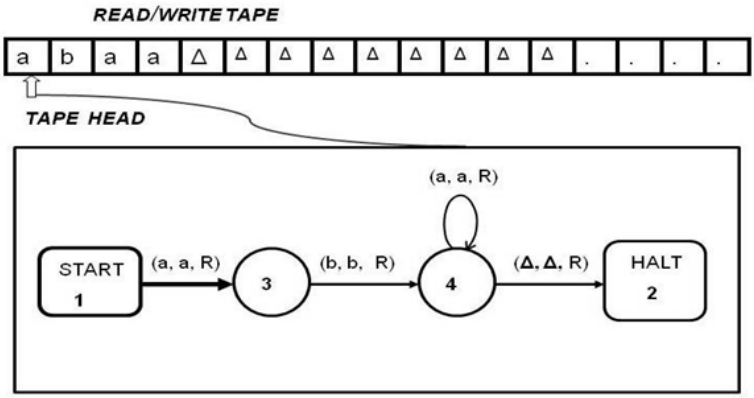


Figure 1: A Turing Machine for aba* with input abaa

In order to clarify aspects of the definition, consider the example TM presented in Figure 1 with the input abaa. The TM given in Figure 1 is designed to accept every string of the following set: {ab, aba, abaa, abaaa, abaaaa, abaaaaa, ...}. Such a set is called a language. A language is a set of strings. This language is usually abbreviated as aba* which is a regular expression. The star in this string, after a is known as the Kleene star which means zero or more of a's. The regular expression aba* means one a followed by one b followed by zero or more a's. This language denoted by the regular expression aba* can be written as: L1 = aba* = {ab, aba, abaa, abaaa, abaaaa, abaaaaa, ...}.

The TM of Figure 1 is shown to have just begun to process the input by starting at the start state. The TM, then takes the first transition from the start state to state 3, reads the first symbol a from the leftmost cell of the tape, writes back a on the same cell and moves right. This is shown in Figure 2.

Next, the machine, from state 3, taking the transition marked by (b, b, R) reads the symbol b from the second cell of the tape and writes back b on the same cell and moves right on the tape. Taking this transition the machine reaches state 4. In the next step, the machine, from state 4, taking the transition marked by (a, a, R) reads the symbol a, from the third cell of the tape, and writes back a on the same cell and moves right on the tape. Taking this

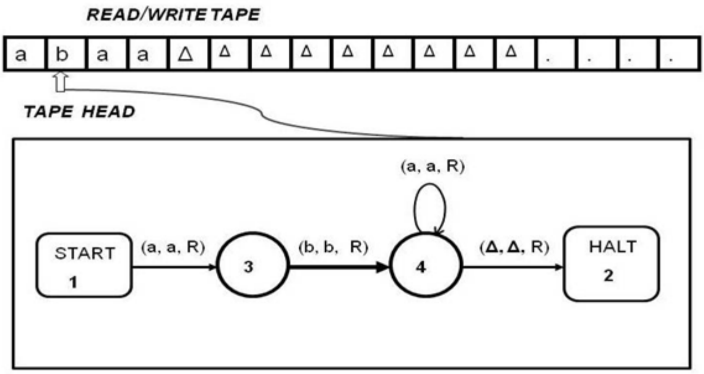


Figure 2: A Turing Machine for aba* which has just scanned the first symbol

transition the machine comes back to state 4. Next, the machine, from state 4, taking the same loop transition marked by (a, a, R) comes back to state 4. In the next step, taking the transition marked by (Δ, Δ, R) from state 4, the machine reads Δ from the fourth cell of the tape, writes back Δ on the same cell and moves right on the tape. Taking this transition the machine reaches the HALT state. The machine accepts the input, abaa, since it reaches the HALT state after traversal of transitions from the start state. From the TM instances of this section, it is obvious that the machine is designed to accept every string of the set, {ab, aba, abaa, abaaa, abaaaa, abaaaaa, abaaaaaa, . . . } = aba* which is a regular language. The above TM for aba* is represented in the table below [2].

FROM	TO	READ	WRITE	MOVE
1	3	a	a	R
3	4	b	b	R
4	4	a	a	R
4	2	Δ	Δ	R

Table 1: The TM for aba* is represented in tabular form

One type of visualization of the TM for aba* is on the following website: `http://www.asethome.org/mathfoundations/asynchronous/`. The website is used as a supplement for an introductory course on automata theory. Learners are not required to take the survey at the bottom of the website; they are encouraged to take the survey when they have time. 43 out of 48 (89.58%) learners answered the following question affirmatively: Do you prefer multiple representations over one representation for asynchronous learning? The learners' response and comments indicate that asynchronous learning can be improved by providing support with multiple representations and engaging learners in the environment through user controls. Learners perceive a sense of engagement or interaction with the content through innovative user control on animated and other representations which may otherwise appear to be just passive lecture notes.

4 Conclusion

In addition to passive lecture notes, students need to be engaged in various ways in an asynchronous learning environment. This study presents evidence on the perceived satisfaction of learners in an asynchronous environment supported with multiple representations and engagement opportunities. We realize that potential contributions of AI, AR, VR, intelligent tutoring systems, and other emerging technologies need to be considered in future research [1, 3, 12, 13, 21, 20, 23]. There are also additional opportunities in related application areas. For example, software development relies on modeling the software in various levels, including the design level. Design tools based on statecharts [5, 6] have been very useful for modeling dynamic aspects of software. Statecharts can be an intuitive way of presenting TMs in a notation that is appropriate for representing software features.

We are planning to build an inclusive and comprehensive asynchronous learning environment for math foundations on the following site: `http://www.asethome.org/mathfoundations/`. Our approach is to build the system in an iterative evolutionary development process [15, 17] so that the design and design review can be thoroughly performed in each iteration. In each iterative review we can ask: "Is this the best we can do? Or is there a better design for serving our students?" It is reasonable to assume that an autonomous (or semi-autonomous) intelligent asynchronous system can be designed and implemented in order to satisfy modern learners. The global nature of the current COVID-19 pandemic emphasizes the need for fully exploring the potential of asynchronous learning and addressing educational challenges and presents wider opportunity for further research into this space. Additional future work may include more general and comprehensive studies about asynchronous learning including innovative interactions and architectural support focused around student success. Support through innovative computer aided interactions may give a sense of engagement highly desired by asynchronous learners in situations where in person face to face engagement may not be achievable.

5 Acknowledgement

The authors are thankful to the National University (nu.edu) administrators, staffs, faculty, and students for their cooperation and support during this research.

References

[1] Ali Alkhatlan and Jugal Kalita. Intelligent tutoring systems: A comprehensive historical survey with recent developments. *arXiv preprint arXiv:1812.09628*, 2018.

[2] D Cohen. *Introduction to Computer Theory.* John Wiley, New York, 2 edition, 1996.

[3] Clayton R Cook, Stephen P Kilgus, and Matthew K Burns. Advancing the science and practice of precision education to enhance student outcomes. *Journal of School Psychology*, 66:4–10, 2018.

[4] Wayne Goddard. *Introducing the theory of computation.* Jones and Bartlett Publishers, Inc., 2008.

[5] David Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.

[6] David Harel and Michal Politi. *Modeling reactive systems with statecharts: the STATEMATE approach.* McGraw-Hill, Inc., 1998.

[7] J Hopcroft, R Motwani, and J Ullman. *Introduction to Automata Theory, Languages, and Computation.* Pearson Education, 2 edition, 2007.

[8] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65, 2001.

[9] Dexter C Kozen. *Theory of computation.* Springer Science & Business Media, 2006.

[10] Vivek Kulkarni. *Theory of Computation.* Oxford University Press, 2013.

[11] Harry R Lewis and Christos H Papadimitriou. *Elements of the Theory of Computation.* Prentice Hall, 2 edition, 1997.

[12] Richard Lowe and Wolfgang Schnotz. *Learning with animation: Research implications for design.* Cambridge University Press, 2008.

[13] João Maroco, Ana Lúcia Maroco, Juliana Alvares Duarte Bonini Campos, and Jennifer A Fredricks. University student's engagement: development of the university student engagement inventory (usei). *Psicologia: Reflexão e Crítica*, 29(1):21, 2016.

[14] John Martin. *Introduction to Languages and the Theory of Computation.* McGraw-Hill New York, 4 edition, 2010.

[15] John Ousterhout. *A Philosophy of Software Design.* Yaknyam Press, 2018.

[16] Emil L Post. Finite combinatory processes-formulation 1. *The journal of symbolic logic*, 1(3):103–105, 1936.

[17] Roger S Pressman and Bruce Maxim. *Software engineering: a practitioner's approach.* McGraw-Hill Education, 8 edition, 2014.

[18] Elizabeth F Purinton and Megan M Burke. Engaging online students: Using a multisensory exercise for deeper, active learning. *Marketing Education Review*, 30(1):29–42, 2020.

[19] Elaine Rich. *Automata, computability and complexity: theory and applications.* Pearson Prentice Hall Upper Saddle River, 2008.

[20] Susan H Rodger, Bart Bressler, Thomas Finley, and Stephen Reading. Turning automata theory into a hands-on course. *ACM SIGCSE Bulletin*, 38(1):379–383, 2006.

[21] Susan H Rodger and Thomas W Finley. *JFLAP: an interactive formal languages and automata package.* Jones & Bartlett Learning, 2006.

[22] Jacques Sakarovitch. *Elements of automata theory.* Cambridge University Press, 2009.

[23] Mehdi Mehri Shahabadi and Megha Uplane. Synchronous and asynchronous e-learning styles and academic performance of e-learners. *Procedia-Social and Behavioral Sciences*, 176(20):129–138, 2015.

[24] Michael Sipser. *Introduction to the Theory of Computation.* Cengage learning, 3 edition, 2012.

[25] A. M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 01 1937.

[26] Anna Wong, Nadine Marcus, Paul Ayres, Lee Smith, Graham A Cooper, Fred Paas, and John Sweller. Instructional animations can be superior to statics when learning human motor skills. *Computers in Human Behavior*, 25(2):339–347, 2009.

[27] Dezhi Wu, Michael Bieber, and Starr R Hiltz. Engaging students with constructivist participatory examinations in asynchronous learning networks. *Journal of Information Systems Education*, 19(3):9, 2019.

Assessing Higher-Order Thinking Skills for Program-Level Student Outcomes: A Longitudinal Case Study*

Kimberly W. Bartholomew
Information Systems and Technology
Utah Valley University
Orem, UT 84058
barthoki@uvu.edu

Abstract

Some university programs such as Information Systems, Information Technology, and Computer Science are under accreditation from organizations like the Accreditation Board for Engineering and Technology (ABET). ABET accredited programs are required to assess student learning outcomes related to the specific discipline. These student outcomes may be assessed at a course-level or globally at a program-level. This case study presents a structure for design, administration and evaluation of summative, ABET program-level assessment, cycling over five years, embedded in a capstone course, that employed complex questions requiring higher-order thinking skills (HOTS) from students in various Information System program emphases. Justification for the choices related to embedded, direct, summative, assessment that measured HOTS is provided.

1 Introduction

University programs in the area of technology often seek accreditation to help differentiate the curriculum and quality of the program. The Accreditation Board for Engineering and Technology (ABET) accredits programs at both the bachelor and master degree levels [1]. Specifically, the ABET Computing Accrediting Commission (CAC) provides program general criteria, student outcomes, curriculum guidelines, and a continuous improvement process for Computer Science, Cybersecurity, Information Systems, and Information Technology programs [2].

ABET like many modern accrediting organizations have shifted their focus toward a framework that supports Outcomes Based Education (OBE). The OBE movement began in K-12 education and expanded to post-secondary education. The focus of OBE is to measure student achievement based on what they have learned, or their "outcomes" [5]. ABET asks programs to identify a set of program objectives that are long-range goals that graduates should exhibit several years after graduation [10]. These program objectives must be aligned with the parent organization mission and be posted for review by program stakeholders. For programs accredited under the ABET CAC, a group of student outcomes provide the set of skills, knowledge, and behaviors student should acquire as they advance through the program toward graduation [2]. While some of these defined student outcomes can be viewed as course-level outcomes, one or more of these outcomes may represent more comprehensive abilities that are achieved across multiple courses. This report will refer to these more comprehensive student outcomes as program-level outcomes, not to be confused with program objectives. This case study will focus on the evolution of a program-level assessment of a student outcome for an Information Systems (IS) program accredited by the ABET CAC.

The Information Systems & Technology (IS&T) Department, IS Curriculum Committee developed program objectives for the Information Systems bachelor's program that are aligned to the mission of Utah Valley University (UVU). Given the CAC IS set of student outcomes, the IS Curriculum Committee examined courses in the program where these student outcomes are addressed and where courses have significant application where the outcomes will be assessed. A mapping of student outcomes to courses for the different program emphases was created as a guide to see the coverage across courses.

While the goal of ABET student outcomes is to measure competencies of program graduates, the assessments used to evaluate these outcomes are often conducted at the course-level. More general or comprehensive student outcomes that do not map to a specific course may need to be assessed instead at a program or global level.

One main goal of accreditation is to encourage continuous improvement through a process of assessing and then weighing the degree of student outcome success. When aspects of the assessment show progress is not at a desired level, a plan of improvement should be created and executed. An example of a continuous improvement assessment cycle as implemented at UVU, for the IS program, is shown in Figure 1.

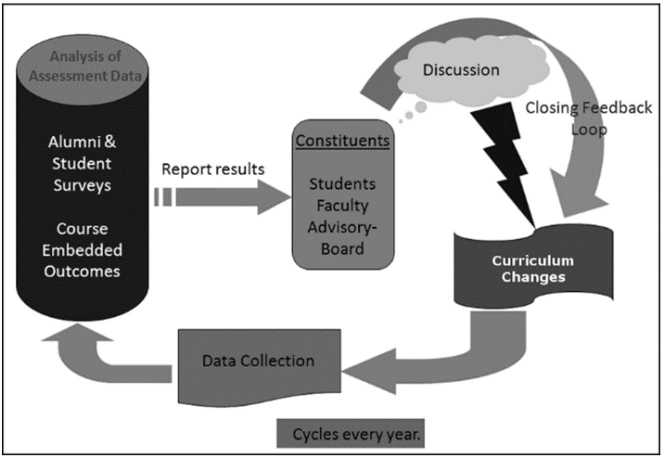


Figure 1: Continuous Improvement Assessment Cycle

Assessment is not the only means of achieving continuous improvement of an accredited ABET program, but it can be a valuable tool. ABET requires that a program under accreditation follow a process for documenting the ongoing process of assessing and evaluating student outcomes [2][12].

In an educational setting, assessment can be formative or summative. Problem-Based Learning (PBL) or Self-Regulated Learning (SRL) are pedagogies that encourage student to use formative assessment or as part of their learning process [9]. The focus of assessment in this case study is solely on summative assessment for the purposes of reporting and accreditation.

2 Assessment Definitions

Faculty and scholars who design and execute assessments have a set of definitions that are widely referred to by practitioners. Some commonly used terms will help to differentiate between assessment options.

- *Course-level assessment* - Assessment that takes place within a course, also known as embedded assessment [14].
- *Program-level assessment* - Assesses student learning across courses.to measure how students are learning as they progress thought a program [6].

- *Direct assessment* - Direct methods of assessment used to evaluate student work like projects, and portfolios, This assessment is also known as authentic assessment [14]. Direct methods can provide effective and efficient creation and administration of assessments for faculty responsible for an assessment.
- *Indirect assessment* –Assesses student learning by gathering information through methods that do not use actual student work, indirect methods, like surveys, interviews, or focus groups [14].
- *Self-Assessment* - The act of evaluation that a student carries out to understand and learn from assessing their own abilities, processes and learning practices [4].
- *Triangulation* - Using multiple assessment methods for the same student outcome. Triangulation can provide a multifaceted look at student learning outcomes [14].
- *Rubrics* - Scoring guide used by a teacher to evaluate student learning [14]. Use of rubrics help provide consistency in scoring across a group of students. When designing a rubric for accreditation assessment, the dimensions built in into the rubric should be directly related to the program outcomes. Using a rubric for scoring is not a guaranteed method for consistency as problems may still arise when more than one faculty member applies the rubric differently.
- *Higher-Order Thinking Skills (HOTS)* - Critical thinking that consists of using reasoned judgement, being reflective, and knowing when to apply standards or best practices [3]. Asking a student to demonstrate their thought processes with real-world, authentic problems allows an evaluator to assess the level of a student's thinking. Educators seek evidence of HOTS in students when a student can demonstrate that they have obtained new information when they can correlate it with what they already know, and apply the knowledge to achieve a specific purpose.

3 Longitudinal Case Study

Starting in the Spring of 2016, the Information Systems Curriculum Committee (IS Curriculum Committee) at UVU was tasked to design, administer, and assess the ABET CAC IS student outcomes for the bachelor's program with the following emphases: Business Intelligence, Geographical Information Systems, Healthcare Information Systems and Information Security Management. Based on an issue with the previous round of accreditation review, the Outcome j assessment structure and rubric was targeted for revision. ABET CAC Outcome j has recently been renumbered and will be referred to as Outcome 6 throughout the remainder of this report.

ABET Outcome 6: Support the delivery, use, and management of information systems within an information systems environment.[2].

ABET CAC defines an information systems environment as: "an organized domain of activity within which information systems are used to support and enable the goals of the activity. Examples of information systems environments include (but are not limited to) business, health care, government, not-for-profit organizations, and scientific disciplines." [2] The Information Systems emphases in this program were considered "information systems environments" . These IS program emphases did not have a separate capstone course that mapped to this student outcome. Considerations related to the design, and implementation of the assessment for this student outcome are addressed next.

3.1 Concerns with Assessment of Outcome 6

Best practice for developing an assessment strategy recommends triangulation by using several assessment techniques at different levels throughout the program [14]. Other student outcomes assessment across the IS curriculum utilize a wide range of techniques like: surveys, exams, components of student work, peer evaluations, and self-assessments. The IS Curriculum Committee wanted to ensure assessment of this student outcome was a non-trivial, authentic assessment that measured student HOTS directly related to their specific program emphasis. As this assessment is seen as a summative assessment, the goal of asking students to demonstrate a higher level thinking and reasoning is to discover where there are holes or problems in the curricula and not to target remediation for specific students.

In a brainstorming session, the IS committee identified the following questions that needed to be answered before designing and implementing the Outcome 6 assessment:

1. Would the assessment be an embedded course-level assessment or a program-level assessment?
2. Would the assessment be a direct or indirect assessment?
3. How would the assessment questions be structured to require HOTS by challenging students to interpret, analyze, evaluate, synthesize, and apply their skills and knowledge to solving relevant real-world problems?
4. How would the scoring rubric define a set of common performance criteria used to evaluate student answers in each program emphasis?
5. How would the assessment be adopted to find appropriate questions for students who are in non-IS programs?
6. How to make sure student answers are readable and understood by the evaluator?

7. How to prevent students from using the Internet to answer the assessment questions? How to make sure students do not collaborate on their answers? How to require each student in the course to take the assessment.
8. How to incentivize students to give the assessment their best efforts.

Note: The assessment concerns were numbered for easy reference, not to designate any given order or priority.

3.2 Strategies to Address Assessment Concerns

Faculty and other program stakeholders who are responsible for carrying out assessments to support accreditation have many different choices and combinations at their disposal. The key to choosing the most appropriate assessment tool, level and placement for an assessment is to focus on how to best evaluate what students have learned [11]. The strategies used by the author's program are tightly tied to that curriculum, the students in the program, and the learning outcomes to be assessed. These strategies serve as a step in the continuous improvement cycle and are presented here to show an example of an assessment design process, not as a generalized case.

The IS Curriculum Committee chose the following solutions to address concerns regarding the creation and implementation for the Outcome 6 assessment:

- Strategy #1: Because the committee wanted to ensure student investment and effort in the assessment, the assessment was embedded within a course. The problem with this decision was that each of the four Information Systems program emphases did not contain a course that mapped well to this student outcome. All IS majors are required to complete a two-course capstone sequence. The second course in the sequence requires implementation and delivery of a management information system, exactly what was needed for this student outcome. As suggested by Skylock and Reed, the assessment would be placed in the Systems Design and Implementation, a capstone required course for all Information Systems majors [13][7].
- Strategy #2: The committee wanted this assessment to measure individual student knowledge and output. While there is a place for self-assessment, possibly as part of Self-Regulated Learning (SRL), some researchers have shown that students often inflate their measurements if it is tied to the course grade [8]. The assessment type chosen was a direct assessment because this would be a summative assessment without a follow-up learning opportunity.

- Strategy #3: A small group of faculty Subject Matter Experts (SMEs) for each of the program emphases was organized. The faculty SMEs met and created five essay questions, one for each of the following performance criteria: Problem Solving, Environment/Tools, Policy Enforcement, Technical Solution, and Current Events. The quiz questions would ask students to provide expected answers in the upper Blooms level quality answers: evaluation, synthesis, analysis, and application. Other IS programs may choose to map to different performance criteria for this student outcome.
- Strategy #4: A rubric is often used to provide a common scoring tool for an assessment [13]. The committee proposed performance criteria to evaluate this student outcome, trying to ensure that the criteria had relevance for each of the program emphases. The final list of performance criteria used to measure this outcome were: Problem Solving. Environment/-Tools, Policy Enforcement, Technical Solution, and Current Events.
- Strategy #5: The capstone class also enrolls students from other programs such as: Information Technology, and Information Management. Because the assessment is embedded and counted as required coursework, all students in the course needed to participate equally in the assessment. In order to have students take the assessment seriously, relevant, similar questions for each of the non-IS programs needed to be created. SMEs from Information Technology and Information Management helped to suggest and review quiz questions for students in these majors. The other majors could make use of these assessment results, but currently these assessments are not being used outside of the IS program.
- Strategy #6: The assessment quiz answers needed to be easily readable for scoring purposes. Because of the difficulty in reading hand-written answers, the committee decided students would need to enter their answers on the computer. The assessment would be administered with a printed question sheet for each student based upon their IS emphasis or non-IS major. A quiz would be created in the CMS with five empty slots where students would type their answers.
- Strategy #7: To prevent students from collaborating on answers and searching the Internet, the assessment quiz should be proctored. The faculty administering the assessment should be allowed how to deal with students who have missed the assessment. The integrity of the assessment should be kept by all faculty administrators to prevent questions being distributed by students, as these are non-trivial questions to create.
- Strategy #8: The IS committee together with faculty teaching the capstone course examined how the assessment would fit into the required coursework. Course quizzes were scored out of 15 points and were given a low percentage of the final grade, only 10%. The faculty recommended

scoring the assessment quiz at 25 points to add incentive for students to give their best effort in their work.

The next section will describe assessment cycle and lessons learned as the department and faculty conducted the Outcome 6 assessment and reporting.

3.3 Conducting the Assessment

3.3.1 First Assessment: Fall 2016

The program-level assessment for Outcome 6 was proctored in the CMS with a time limit of 45 minutes to answer the five essay questions. Students were informed ahead of time that this was a required quiz. Figure 2 Shows the assessment questions that were administered for Geographic Information Systems majors [1].

The first Outcome 6 assessment was conducted in two sections of the capstone course during Fall semester. One instructor taught both sections, so no coordination was needed to evaluate the quiz to provide students a coursework grade. The assessment quiz was placed in the course Quizzes section, and participated as part of the 10% of the course grade. The capstone faculty graded each quiz question for completeness (all parts of the question were addressed), good grammar, professional tone, and use of complete sentences.

The evaluation of the assessments was then turned over to small teams of SME faculty who graded answers grouped by emphasis. Each quiz and scored the questions based on the IS&T Department's Outcome 6 rubric which were Exemplary=3, Satisfactory=2, Unacceptable=1. A table was then created showing results by program emphasis. For example, Table 1 lists the summary statistics for the Business Intelligence Emphasis for Fall 2016, including frequency and percentages for Exemplary, Satisfactory, and Unacceptable for each category. Category averages are calculated, and the mode is listed for each category.

The following semester, the faculty mentor for the capstone course prepared an assessment report and reported on the assessment findings and recommendations in a department meeting. Recommendations for targeting problem areas in both the Information Security and Healthcare Information Systems courses were given.

Lessons Learned from the First Assessment:

- The number of majors taking the assessment in each of the emphases was relatively small: 8 Business Intelligence, 1 Healthcare Information, 2 Information Security and 3 Geographic Information Systems. It was

[1] Note: The Geographic Information Systems Emphasis is no longer active at the author's institution. There is no longer a problem with sharing these quiz questions as an example.

Geographic Information Systems Questions

Problem Solving:
How should the "human factor" enter into an organization's GIS development and implementation?

Environment (understanding of different computing environments e.g. Tools):
Layer (or levels) are a fundamental means of organizing geographic data in almost all GIS. Explain with some examples why layers are needed.

Policy Enforcement:
Many modern digital cameras and smartphones include geolocation information embedded in a photograph. As the number of geolocation users grows and the proliferation of mobile devices continues, the prospect of individual or enterprise information becoming available to hackers or other unauthorized users is a significant concern. A proposed amendment from the U.S. Federal Trade Commission to the Children's Online Privacy Protection Act (COPPA) specifically addresses the collection of geolocation data from children under age 13. What best practices would you suggest for individuals posting images on social media and photo sharing sites to protect themselves from malicious use of location information?

Technical Solution and Best Practices:
The city of Orem is trying to be more responsive to citizen requests for street repair by offering a Report a Pothole mobile application. How could information from smartphones be used provide exact locations of potholes that could be scheduled for repair? What location information would a citizen need to provide if they did not have a smartphone?

Current Events:
List three international environmental issues that could be addressed using GIS? Explain why GIS could be specifically leveraged in visualizing these issues.

Figure 2: Example Assessment ABET Outcome 6 Quiz Questions: Majors in the Geographic Information Systems Emphasis

difficult to generalize suggested curriculum improvements when numbers of students in emphases were so small. In order to increase the number of majors assessed, it was decided by the IS Curriculum Committee to combine all sections over Fall and Spring semesters for the next assessment.

- Students asked for more time to complete the answers. The instructor was able to increase the quiz duration to 60 minutes. In the future, this would be the time limit given to the assessment. The disadvantage to allowing student more time is that the assessment would consume most of the allotted 75 minute class period.
- Most students took the assessment seriously and wrote lengthy answers. Some international students struggled to write complete or thoughtful answers.

Category	Exemplary		Satisfactory		Unacceptable		Average	Mode
	Freq	*%*	*Freq*	*%*	*Freq*	*%*		
Problem Solving	4	50%	4	50%	0	0%	2.42	Exemplary/ Satisfactory
Environment/ Tools	2	25%	6	75%	0	0%	2.25	Satisfactory
Policy Enforcement	4	50%	3	37.5%	1	12.5%	2.26	Exemplary
Technical Solution	2	25%	6	75%	0	0%	2.21	Satisfactory
Current Events	6	75%	2	25%	0	0%	2.59	Exemplary

Table 1: Outcome 6 Assessment Scoring Rubric

3.3.2 Second Assessment: Fall 2017 & Spring 2018

Before the assessment could be administered, a group of SME faculty in Application Development met and created quiz questions for this new IS program emphasis. Students in the capstone course received a notice as to the purpose of the ABET assessment and reminded that their attendance in class was required to proctor the quiz. The second assessment was administered in two sections for both semesters. There were no GIS majors who took the assessment as this program emphases was being discontinued. By combining these semesters, the number of majors in each emphasis increased. The number of majors was: 22 Business Intelligence, 7 Information Security, 5 Application Development, and only 1 Healthcare Information Systems.

The faculty mentor gathered the scoring from the SME faculty groups and prepared an assessment report which was presented in a department meeting. Most students were able to explain, justify and support their answers to these five categories of questions.

Lessons Learned from the second assessment:

- Students were better able to complete their answers given the longer time limit of 60 minutes.
- Because student received prior notice about the purpose and importance of the assessment, the instructor did not need to justify to the students why they had to take the quiz and give it their best efforts.

- Again, several international students struggled to complete meaningful answers, even leaving some answers blank. The IS Curriculum Committee should attempt to address this issue in the future.

3.3.3 Third Assessment: Fall 2019 & Spring 2020

The assessments have been conducted. Assessment answers are in the process of being delivered to SME faculty groups for scoring.

4 Conclusion

The IS Curriculum Committee at UVU has chosen to implement the ABET Outcome 6 assessment as an embedded, program-level assessment and has experienced that authentic, performance-based assessments usually take more time to create and administer [3]. These types of assessments may also be more subjective when scored. The IS Curriculum Committee and SME evaluators have been successful in using shared rubric and good communication to ensure consistency in scoring.

The process of continuous improvement as required by ABET accreditation has successfully been applied to this assessment process as quiz questions are reviewed, new questions created as program changes necessitate, and recommendations for curriculum improvement are acted upon.

References

[1] ABET. About ABET. `https://www.abet.org/about-abet/`. Accessed 2020-02-15.

[2] ABET. Criteria for accrediting computing programs, 2020 –2021. `https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2020-2021/`.

[3] Yousef Abosalem. Assessment techniques and students' higher-order thinking skills. *International Journal of Secondary Education*, 4(1):1, 2016.

[4] Heidi L Andrade. A critical review of research on student self-assessment. In *Frontiers in Education*, volume 4, page 87. Frontiers, 2019.

[5] Jamal Ahmad Dargham, Ali Chekima, Renee Chin Ka Yin, and Farrah Wong. A direct assessment method of the achievement of the program outcomes from the courses outcomes. In *2013 IEEE 5th Conference on Engineering Education (ICEED)*, pages 131–135. IEEE, 2013.

[6] Hansen Hunt. What is program-level assessment and how to do it, 2017. `https://www.instructure.com/portfolium/blog/program-level-assessment`.

[7] Y Jin, K Ahn, and B Choi. An use of course-embedded assessment for assessing program outcomes. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 1. Citeseer, 2010.

[8] Brenda Logan. Reviewing the value of self-assessments: Do they matter in the classroom? *Research in Higher Education Journal*, 29, 2015.

[9] Ernesto Panadero, Heidi Andrade, and Susan Brookhart. Fusing self-regulated learning and formative assessment: A roadmap of where we are, how we got here, and where we are going. *The Australian Educational Researcher*, 45(1):13–31, 2018.

[10] Muhammad H Rashid. The process of outcome-based education-implementation, assessment and evaluations. *age*, 21:1, 2012.

[11] Nadia Rifaat, Waleed M Al Sabhan, and Mohammed Nour. Effective learning outcomes assessment: The case of the mis department at the uos. *Journal of Education and Vocational Research*, 3(2):58–70, 2012.

[12] Bruce Saulnier. A paradigm for student learning outcome assessment in information systems education: Continuous improvement or chasing rainbows? *Information Systems Education Journal*, 12(1):4, 2014.

[13] Kristi Shryock and Helen Reed. Abet accreditation: Best practices for assessment. In *Proceedings of the ASEE Annual Conference and Exposition*, 2009.

[14] Joni E Spurlin, Sarah A Rajala, and Jerome P Lavelle. *Designing better engineering education through assessment: a practical resource for faculty and department chairs on using assessment and ABET criteria to improve student learning*. Stylus Publishing, LLC., 2008.