A publication of ISCA*:
International Society for Computers
and Their Applications

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

*** "International Journal of Computers and Their Applications is abstracted and indexed in INSPECT and Scopus."**

Volume 27, No. 1, March 2020 **ISSN 1076-5204**

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

ISCA Headquarters···•···278 Mankato Ave, #220, Winona, MN 55987···•···Phone: (507) 458-4517
E-mail: isca@ipass.net • URL: http://www.isca-hq.org

Guest Editorial:
Special Issue from ISCA Fall--2019 SEDE Conference

This Special Issue of IJCA is a collection of five refereed papers selected from the SEDE 2019: 29th International Conference on Software Engineering on Data Engineering, held during September 30-October 2, 2019, in San Diego, CA, USA,

Each paper submitted to the conference was reviewed by at least two members of the International Program Committee, as well as by additional reviewers, judging the originality, technical contribution, significance and quality of presentation. The proceedings for this conference can be found online at https://easychair.org/publications/volume/SEDE_2019. After the conferences, five best papers were recommended by the Program Committee members to be considered for publication in this Special Issue of IJCA. The authors were invited to submit a revised version of their papers. After extensive revisions and a second round of review, these four papers were accepted for publication in this issue of the journal.

The papers in this special issue cover a broad range of research interests in the community of computers and their applications. The topics and main contributions of the papers are briefly summarized below.

MING-CHANG HUANG of the University of North Carolina at Charlotte present their paper "A Peer-to-Peer Reputation Evaluation System" where they introduce FuzRep, a reputation system for P2P networks. There are two goals of FuzRep, the first is to motivate information sharing and the second is to do this while avoiding overloading providers. Fuzzy logic is used in FuzRep to help minimize bandwidth and manage reputations using knowledge from social network researchers. By bringing together selective polling as well as service differentiation they are hoping to curtail free-riding behaviors through a reputation system that provides incentives for working together.

FENG YU, DAVID S. WILSON, TASHA M. WELLS, of Youngstown State University, and MOHAMMAD A. HAMDI and XIAOLAN HUANG of Southern Illinois University, Carbondale, and WEN-CHI HOU of Najran University, Saudia Arabia presented their paper "Scalable Correlated Sampling for Join Query Estimations on Big Data." In this work they present a scalable sampling framework for join query estimations in map-reduce. They also benchmark the framework's performance on large TPC-H datasets and present results on accuracy as well as speedup of t the join query estimations. The results show that their sampling method produces fast and accurate join query estimations on big data.

SARIKA RAJEEV of American University, Washington DC and SHARAD SHARMA of Bowie State University, Bowie MD presented their paper "Evaluation of Game-Theme Based Instructional Modules for Data Structure Concepts." In this work, they seek to use game-based learning and simulation-based learning to motivate students and keep them engaged. They build an instructional module for teaching linked lists and binary search trees and then evaluate them. The results show that this type of module is effective at promoting learning and keeping students engaged.

TERRI HEGLAR, ANDREW PENROSE, AUSTIN YOUNT, KRISTINE GALEK, YANTAO SHEN, SERGIU M. DASCALU, and FREDERICK C. HARRIS, JR of the University of Nevada, Reno presented their work "A Hardware and Software Prototype of the CTAR All-Star." The CTAR All-Star is a hardware and software system to help users perform the Chin Tuck Against Resistance (CTAR) exercise. The authors present the system consisting of a rubber ball, a pressure sensor, and a Bluetooth transmitter along with the mobile application to set up exercises and store results. The device is used as a rehabilitation tool for people with dysphagia and the software monitors and displays the pressure inside the ball on a real-time graph allowing the patient to follow exercise routines set by Speech-Language Pathologists.

COREY M. THIBEAULT, SAMUEL G. THORPE, KIAN JALALEDDINI, NICOLAS CANAC, and ROBERT B. HAMILTON of Neural Analytics, Inc., Los Angeles, CA presented their paper "Design Patterns in a Machine Learning Framework for Medical Diagnostics." This work presents the development of a generic machine-learning framework called Atlas. This framework was designed to help in the development of clinical diagnostic. As part of the design, an overview of the creational patterns and structural patterns, which were used in its development, are covered. The paper concluded with examples and uses of Atlas. This configurable pipeline framework is designed to help in the development of tools buy expert and novice users.

As guest editors, we would like to express our deepest appreciation to the authors and the program committee members of the conference these papers were selected from.

We hope you will enjoy this special issue of the IJCA and we look forward to seeing you at a future ISCA Conference. More information about ISCA Society can be found at http://www.isca-hq.org.

Guest Editors:

Frederick C. Harris, Jr, University of Nevada, Reno, USA, SEDE 2019 Conference Chair
Sergiu M. Dascalu, University of Nevada, Reno, USA, SEDE 2019 Conference Chair
Sharad Sharma, Bowie State University, USA, SEDE 2019 Program Chair
Rui Wu, East Carolina University, Greenville, NC, USA, SEDE 2019 Program Chair

March 2020

A Peer-to-Peer Reputation Evaluation System

Ming-Chang Huang*
University of North Carolina at Charlotte, Charlotte, NC USA

Abstract

Lack of incentives makes most P2P users unwilling to cooperate and lead to free-riding behavior. One way to encourage cooperation is through service differentiation based on each peer's contributions. This paper presents FuzRep, a reputation system for P2P networks. FuzRep uses fuzzy logic method which uses requester's reputation and provider's inbound bandwidth as input information to create incentives for sharing and to avoid overloading problems for primary file providers. Reputation sharing in FuzRep is implemented by interest-based selective polling, which can significantly decrease overheads for reputation communication.

Key Words: P2P network, free-riding, reputation system, incentive, fuzzy logic.

1 Introduction

File sharing system is one of the most popular P2P applications today. One main issue, however, is widely noted in current P2P systems—free-riding. Measurement study of free riding on Gnutella exhibited by Adar in 2000 [1] indicated that approximately 70% of Gnutella users did not share any files, and nearly 50% queries were responded by top 1% peers. In contrast, the percentage of free riders in Gnutella had risen to 85% in 2005 [7]. This reveals how serious the free-riding problem has become. In fact, free-riding roots in the nature of P2P networks – anonymity, autonomy, but lack of incentive. Most users have chosen to freeride since they do not get any benefit by providing resources. In addition, issues such as "hotspot" and "the tragedy of the digital commons" are coming along with serious free-riding problem.

The proposed solution in this paper, FuzRep, is a reputation system for P2P networks. FuzRep is designed for two purposes—motivating information sharing through service differentiation and avoiding overloading for providers. To these ends, FuzRep uses fuzzy logic as a tool to manage reputation and bandwidth. Fuzzy logic fills the gap between engineering mechanical design and human linguistic understanding. While reputation communication is always an important issue in a fully decentralized environment, FuzRep applies selective polling method, which is inspired from social network researches [8, 9], to discover a peer's global contribution score by selectively polling peers in the same community. Figure 1 shows the architecture and processes of FuzRep.

The rest of the paper is organized as follows. In Section 2, we present related work. In Section 3, we discuss the premises of this paper. In Section 4, we address the detail design of. FuzRep. Experiments are presented in Section 5. Section 6 is another method to implement the fuzzy reputation system. Section 7 is

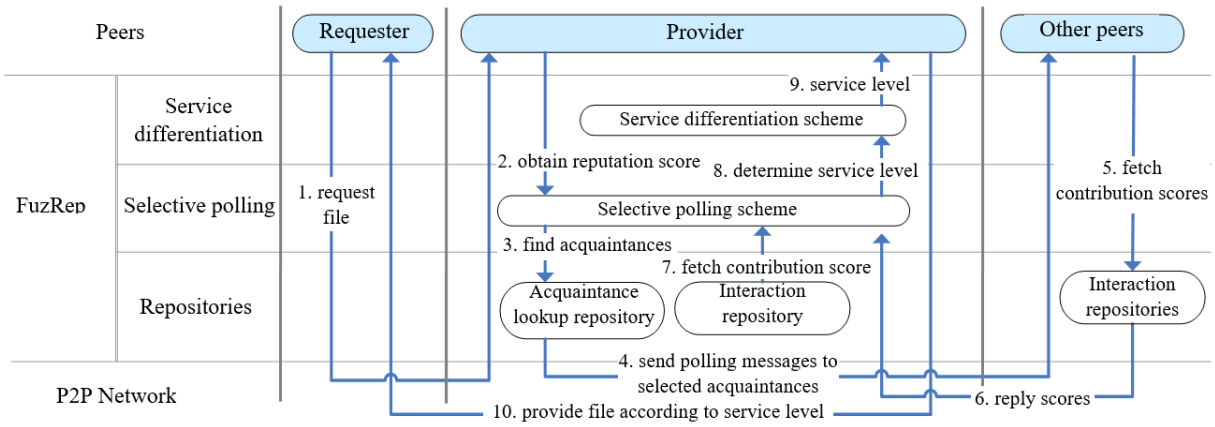


Figure 1: FuzRep architecture and operational processes

* Department of Business Information System/Operations Management.
Email: mhuang5@uncc.edu.

our conclusion and areas for future work

2 Related Work

Most research on P2P free-riding aim at providing incentives to encourage cooperation. While some recent studies are trying game theoretical approaches [2, 5], two major approaches are through monetary payment and reputation judging.

Payment schemes inherit the philosophy of economics, which treat P2P file exchanges as transaction activities and thus requesters should pay for their received files. Since tokens are required for each download, free riders are forced to share files to earn tokens. In [11, 19], incentives are driven by payment schemes. One advantage of payment scheme is its fast speed to converge; that is, peers tend to stop free-riding in a short time under payment schemes. However, one critical drawback of payment schemes is the high mental transaction cost indicated in [12, 16].

Instead, reputation systems learn from sociology concept, which treats each file exchange as a relationship building process. Sharing files with others brings higher reputation, and, on the contrary, free-riding leads to a bad social image. Reputation systems are used to map scores of reputations to adequate service levels. [3, 10] show how to manage reputation in a fully decentralized manner. However, most P2P reputation systems emphasize the designs of security related issues rather than clearly provide incentives to encourage sharing.

3 Premises

We will discuss the basic design considerations and assumptions in this section.

3.1 Design Considerations

There are several considerations while we design the reputation system for P2P networks to make it more scalable and deployable.

1. The reputation management should not rely on any auditing authority, which means there is no centralized agent to store and deliver reputation information.
2. Overhead of reputation management should be minimized, especially the bandwidth consumption on communication messages.

The design of our reputation system is mainly based on Gnutella network; however, the reputation system should be easy to be deployed to other P2P systems.

3.2 Basic Assumptions

To realize FuzRep, there are two basic assumptions. First, we assume each peer keeps two repositories locally—*interaction repository* and *acquaintance lookup repository*. Details of the two repositories are described as below.

- The *interaction repository* contains two attributes (*servent_id*, *contribution_score*) where *servent_id* is the unique identifiers of transacted peers, and *contribution_score* is calculated from old interactions based on file sizes.
- The *acquaintance lookup repository* contains two attributes (*servent_id*, *IP_address*) where *servent_id* is inherited from interaction repository, and *IP_address* is the online acquainted peers' IP addresses which can be gathered and updated from Gnutella ping and pong descriptors [17].

To manage reputation in a decentralized way, interaction records must be stored locally. However, peers have motives to tamper their interaction records for their personal benefits, either to hide downloading activities or to exaggerate the shared amount. To avoid this, a peer's contribution scores are stored in other peers' interaction repositories that the peer has interacted with. Therefore, as you can imagine, a peer's contribution scores are distributed among other interacted peers. To obtain a peer's complete contribution score, we have to collect it from those peers.

The idea of interaction repository is quite similar to XRep's servent repository [3]. We additionally consider file size as a rating criterion because the efforts to share a big-size file should be more than smaller one in terms of storage space, consumed bandwidth and required transmission time. Therefore, we separate files into 3 categories—greater than 10MB, between 1MB and 10MB, less than 1MB. Based on the file size distribution shown in [13], each category contains about one-third of total sharing files. Providers will earn 5 contribution scores for sharing a large file, and it will cost 5 contribution scores for the requesters to request that, while 3 for a medium file and 1 for a small file respectively. On the other hand, the acquaintance lookup repository is designed to accomplish the reputation communication process. Details about it will be discussed in the following section.

Secondly, to accomplish selective polling, we also assume peers with the same interests form virtual communities automatically, which have been observed and proven in [9]. Therefore, peers in the same community will frequently interact with each other. It would be much more efficient to discover a peer's global reputation by polling other members in the same communities instead of flooding polling messages to unrelated peers.

4 FUZREP

FuzRep is a design of fuzzy-based reputation system for P2P networks. It includes three techniques—reputation determination, selective polling, and service differentiation. In this section, we are going to describe how FuzRep works by revealing answers of the following questions.

- How to determine a peer's reputation level? What are the criteria? How to maintain it?
- How and when to share the contribution information?

How to encourage sharing and discourage free riding? How to differentiate the service level?

4.1 Reputation Determination

In FuzRep, one's reputation is determined by his contributions to communities. To this end, a peer should save interaction information into local interaction repository, including the unique IDs of interacted peers and subjective, accumulated contribution scores of them. The interaction repository is updated after every successful interaction. Note that the initial local contribution score is set to 0 originally for any pre-unknown peers at their first interactions.

A global aggregated contribution score is used to determine a peer's reputation. It is carried out in two phase computes - personal reputation inference and global reputation deduction. In personal reputation inference, peer simply fetches targeted peer's contribution score from local interaction repository. Then, in global reputation deduction, the peer should run a reputation aggregation process, namely selective polling, in FuzRep.

4.2 Selective Polling

Reputation information sharing in the P2P environment is a big challenge. We classify different solutions into three categories: centralized authorizing, distributed polling, and transitive computing. Examples of the three approaches are illustrated in Table 1.

Table 1: Three reputation sharing approaches

	Centralized authorizing	**Distributed polling**	**Transitive computing**
Examples	CORC, DCRC [6]	XRep [3]	EigenTrust [10]

In [8], authors present an efficient interest-based content location scheme, which gives us an idea on reputation discovering process. We propose a novel approach called selective polling to facilitate our reputation information sharing process in P2P networks. Selective polling is conceptually based on social networks. In self-organizing P2P communities, if a peer *Q* requests a file from another peer *M*, this implies that they have similar interests. It is reasonable to suppose that *Q* had downloaded files from other peers in the same community which have interacted with *M*. Therefore, from *M*'s point of view, a natural and efficient way to obtain *Q*'s global contribution score to decide a proper service level for *Q* is polling other members in his community.

Algorithm 1 demonstrates the recursive discovery algorithm of selective polling. There are two key parameters, illustrated in Figure 2, which should be decided first. There are two key parameters, illustrated in Figure 2, which need to be further decided here.

- *H* denotes how many hops should be taken to aggregate a satisfied global contribution score, which has similar function as TTL (time to live).
- *K* denotes how many peers should be selected in each iteration.

Algorithm 1: Selective polling algorithm

```
Selective_Polling (H', K) {
  Let C represent # of lacked acquaintance;
  Let TTL represent the time to live for flooding packets;
    if H' == 0 then
      if ever interacted with Q before then
        V_Q ← Q's contribution score in trans. repository;
        Return V_Q;
      else
        Ignore query;
    else if (H' == H) and (C > 0) then
      Select IP addresses of all peers from acquaintance lookup repository;
      if ever interacted with Q before then
        V_Q ← Q's contribution score in trans. repository;
        Return V_Q + Selective_Polling(H'-1, K-C)
          + Restricted_Flooding(C, TTL);
      else
        Return Selective_Polling(H'-1, K-C) +
        Restricted_Flooding(C, TTL));
    else
      K' ← (C > 0 ? K-C, K);
      Select IP addresses of K' peers from acquaintance lookup repository;
      Send query to the K' peers;
      if ever interacted with Q before then
        V_Q ← Q's contribution score in trans. repository;
        Return V_Q + Selectvie_Polling(H'-1, K') ;
      else
        Return Selective_Polling(H'-1, K');
}
```

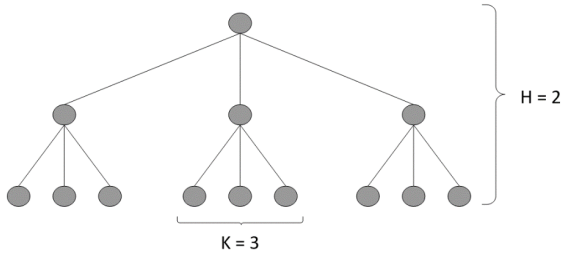


Figure 2: Selective polling with parameters H=2, K=3

One should note that if *M* is a new peer and never showed up in the system before, then its acquaintance lookup repository is empty. Similarly, even though *M* interacted with other peers before, if all of his acquaintances are not online, *M* faces the same predicament as a new peer. EigenTrust introduces pre-trust peers to fix this problem. Our solution is to turn selective polling into distributed polling which is referred as "restricted flooding" in our selective polling algorithm. Distributed polling is to spread polling messages to all direct connected peers and, then, keeping messages flooding until TTL decreased to zero. In our case, the value of TTL for flooding in our case is set to a very small number (about 2) since it is used to be an auxiliary of the first hop of *H*. After collecting the responses, peer M randomly selects K contribution scores out of the replies for the use of reputation deduction. The selective polling brings a new way to discover a peer's global reputation. After gathering global reputation scores, we are able to discriminate service level.

4.3 Service Differentiation

The service differentiation is determined by a reference matrix comprised by two metrics including the deducted global reputation of file requester and the available inbound bandwidth of file provider. The goal of the first metric, deducted global reputation, is meant to encourage file sharing; the second metric, left bandwidth, is to prevent overloading and to alleviate the hotspot issue. Figure 4 shows the reference matrix. It is a two input, one output system. In fuzzy logic, membership function is used to show the degree of membership of variables. We choose to use Gaussian membership function because its smoothness notation is capable of dealing with nonlinear variables. Figure 3 shows the membership functions of reputation, bandwidth, and service level.

The output value of service level is between 0 and 1. Then the allowing sharing speed is the output value multiplied by the maximum connection speed that the provider is willing to offer. If a requester's reputation is worst or a provider is about to overload, the output value would be close to zero which will reject file transfer.

5 Experiments

In this section, we discuss the design, implementation and results of our experiments to disclose the efficiency and effectiveness of FuzRep. They are conducted in two phases. Phase one takes a macro perspective to examine selective polling by simulating a 100 node P2P network which will be discussed in Section 5.1. Phase two, as described in Section 5.2, takes a micro perspective to evaluate service differentiation based on a designed scenario, happened on an individual peer. FuzRep is determined successful if and only if it can collect reasonable global contribution scores of requesters by selective polling and can reflect the scores to corresponding service levels.

5.1 Experiment on Selective Polling

The simulation recorded every interaction, thus a list of true global contribution scores of each peer is available for the use of evaluation. After 300 rounds of bootstrapping interaction, one randomly chosen peer *S* started to discover the other 99 peers' contribution scores by selective polling. We separate the results of selective polling into two sets, *G1* and *G2*. *G1* is the set of peers who have at least one interest file category the same with *S*. Peers in *G1* are highly possible to interact with *S*. *G2* is the set of peers who do not have any interest overlapped with *S*. They are unlikely to interact with *S*.

Simulation settings are shown in Table 2. We evaluate the selective polling scheme by comparing a peer's real global contribution score with the score gathered from the selective polling. Figure 5 shows the result of our simulation. There are 46 scores in *G1* and 53 scores in *G2*. The distances between true scores and discovered scores are errors of selective polling. Since selective polling does not poll every peer in the network, existence of errors is unavoidable. However, as we observed from Figure 5, selective polling is capable of discovering a reasonable score to represent a peer's global contribution. If we exclude outliers, whose true scores are greater than 100 or less than -100, there are 72 scores left, and the mean error between the 72 true scores and the corresponding 72 discovered scores is

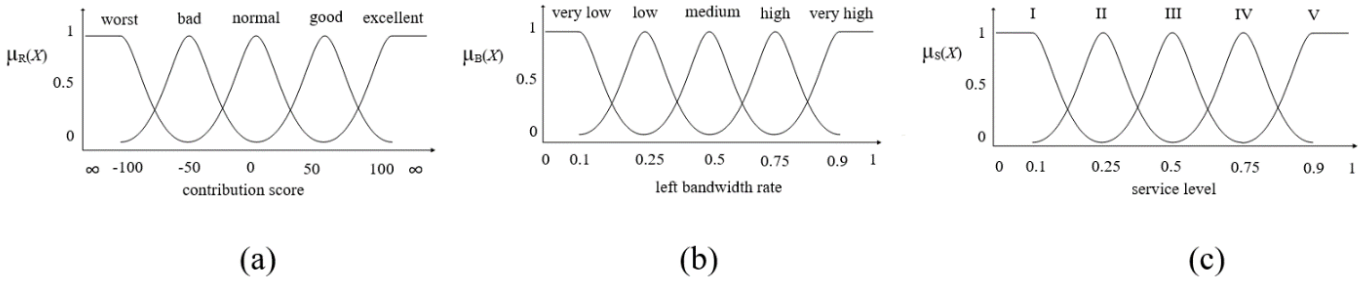


Figure 3: Membership function of (a) reputation, (b) bandwidth and (c) service level

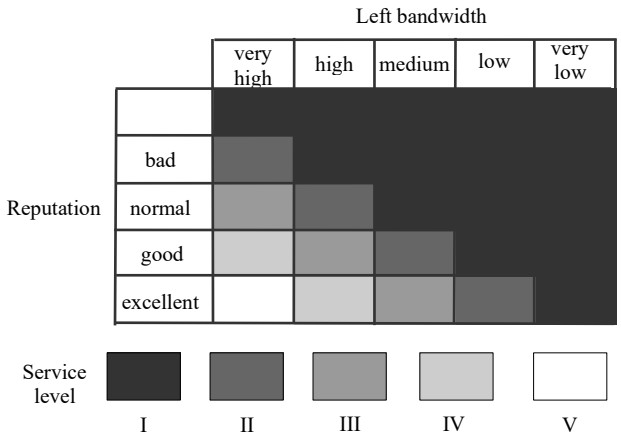


Figure 4: Reference matrix

18.25. That would not cause significant difference on reputation determination under fuzzy-based service discrimination.

5.2 Experiment on Service Differentiation

MATLAB Fuzzy Logic Toolbox [18] provides a convenient way to design and simulate fuzzy logic systems. We make use of it to develop and examine the service differentiation model

Table 2: Simulation settings

Peer	# of peers	100
	% of free rider	85%
	# of direct connected neighbor	*D-U(1,10)*
	Online possibility	*U(0,1)*
	Possibility to send a query	*U(0,0.5)*
	# of interest file category	*D-U(1,3)*
File	# of file categories	5
	# of files in each category	50
	file popularity	Zipf's distribution [7, 12]
	file size distribution	Distribution in [12]
Selective polling	Parameter H	2
	Parameter K	10
	TTL	2
Simulation	# of cycles in a experiment	300
	# of experiments	5

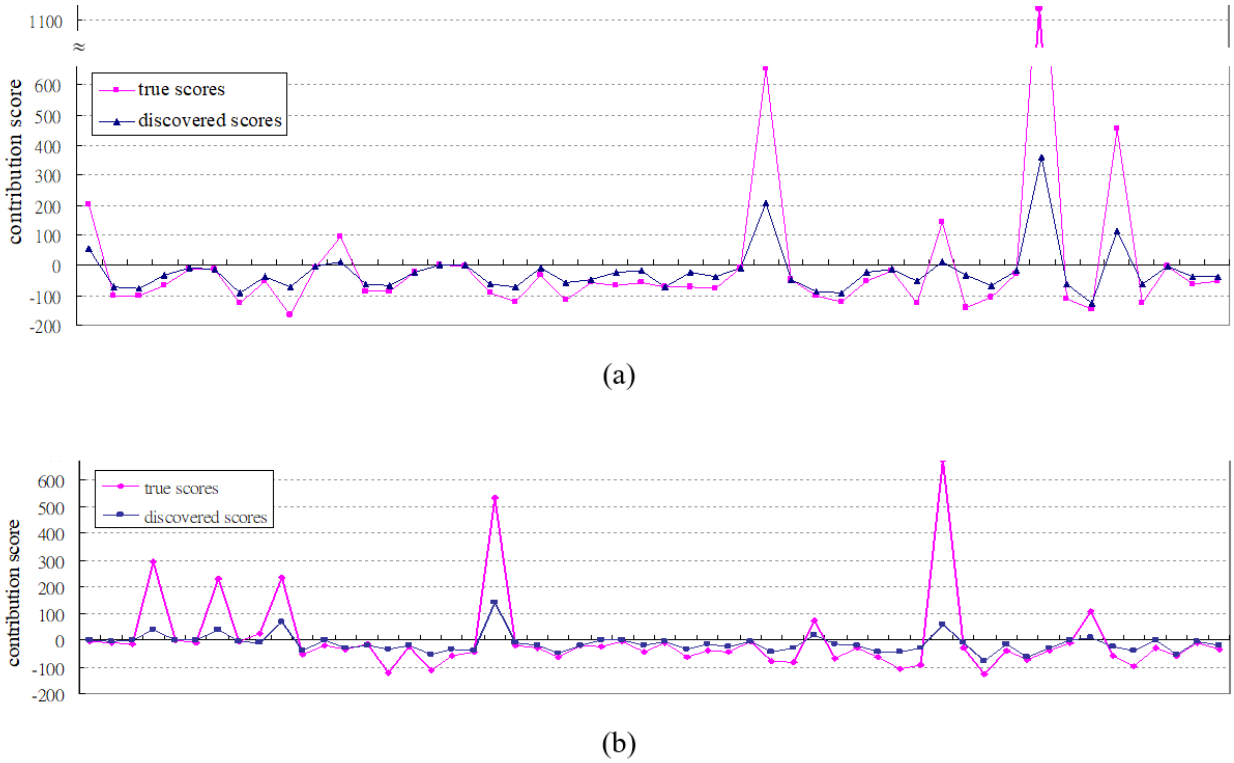
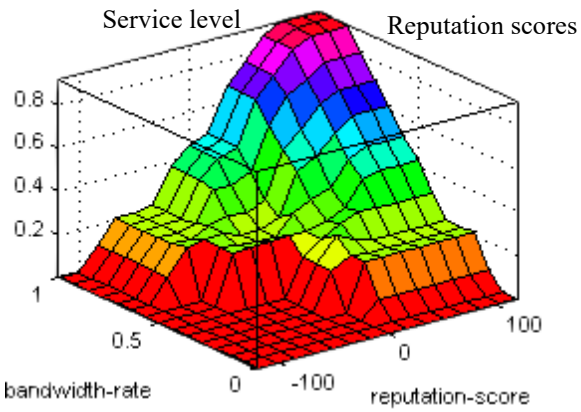
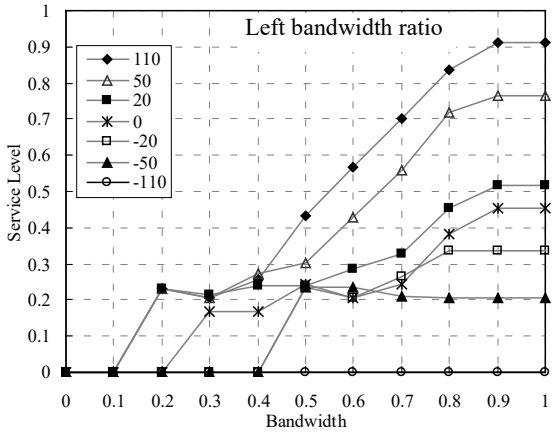


Figure 5: Selective polling evaluation: comparing the true scores with discovered scores. (a) shows the scores difference of peers who have interest overlapped with S: (b) shows those of no interest overlapped with S



(a)



(b)

Figure 6: Service differentiation; (a) surface of service level from Matlab, (b) illustration of actual values

of FuzRep. Figure 6 shows how service level, reputation, and bandwidth interact with each other.

Under service differentiation schemes like ours, it is still difficult to say whether a free riding peer will change his behavior or not because of the complexity of human decision. Heterogeneous peers have different tolerances to transmission speeds. It is difficult and obviously absurd to define a threshold, say 5 Kbps, and assumes that free riders would definitely stop free-riding if their perceived service level is lower than that threshold. However, as shown in Figure 6(b), the discriminated service level has been demonstrated as an incentive for peers to cooperate.

5.3 Discussion

As what Figure 5 shows, after excluding outliers, the mean error of scores of *G1*, the set of peers who have interest overlapped with *S*, is 18.21; meanwhile, the mean error of scores of *G2*, the set of peers who do not have any interest overlapped with *S*, is 18.39. It shows no significant difference on the effectiveness of selective polling when it is applied to different interest groups. The reasons, what we believe, may be due to the consulted peers, which have interest partially the same as *S*, were also possibly interested in categories which *S* is not interested. The diversity of interest may increase as the *H* of selective polling goes up. It could relax concerns that selective polling was only capable to discover reasonable scores of peers of the same interests. The finding is especially useful as the number of peers or interest categories increased in a real P2P network.

Also, the characteristic of zero cost identities [4] in P2P networks might be a threat to FuzRep. It is true that a free-riding peers can easily change their identity which turns their negative contribution score into zero. Although recounting contribution score may make a difference in service level, that will always keep free riders fluctuating their service level from I to III. That is, sharing files is the only way for peers to raise their service levels.

Another issue is the adaptability of FuzRep. In our experiment, *H* is set to 2, and *K* is set to 10. However, these parameters for selective polling should not be fixed. Instead, *H* and *K* should be dynamically decided based on dynamic conditions of P2P environment such as total number of peers, churn rate of peers, frequency of interactions between peers, strength of interest based P2P communities, etc. Nevertheless, applying fuzzy logic to determine peers' reputation can compensate unavoidable errors of the selective polling scheme in FuzRep.

6 Another Method to Implement the Reputation System

In this Section, 1 propose another method for implementation of the reputation system. The following issues are particularly addressed in our reputation system based on the considerations of system scalability and deployment.

1. The reputation management should not rely on any auditing authority. That is, there is no centralized agent to store and deliver reputation information.
2. The design of the reputation system should be network structure independent. In other words, it can be deployed on different P2P structures, which may be DHT-based (e.g. Chord, CAN, Kademlia), hierarchical (e.g. Gnutella, KaZaA), or completely unstructured (e.g. old Gnutella).

System efficiency and overheads become major concerns if there are no hubs of information or pre-sorting schemes. To ease this problem, we turn to a sociological construct upon interactions among peers described in the following assumptions.

6.1 Basic Assumptions

Considering the above issues of this study, an essential assumption is made: peers of the same interests form communities either implicitly or explicitly, and the clustering effect within a community is strong enough to achieve a representable consensus on the participants' reputation under the dynamic nature of P2P networks. Despite the fact that peers, in the context of our work, may not know which communities

they belong to, they can figure out who are in the same community by reviewing their interaction histories.

In fact, several studies have, at least partially, validated the above assumption. While virtual communities on the web have been studied extensively, there has been a trend towards discovering P2P communities since P2P applications are getting more and more popular. It is observed that virtual communities do exist in P2P systems [11, 13]; furthermore, some research has made use of the clustering effect of communities or interests to improve service quality or diminish overheads [8-9, 11, 14].

We intend to leave behind the low-level implementation of routing which is highly related to the structure of P2P network. Thus, we also assume that each peer has a unique identifier which serves to both identify recognition and routing direction in the reputation system.

6.2 Reputation Storage and Computation

To accomplish a reputation system, it is indispensable to store information of users' behavior as a fundamental input. Here, another method for the fuzzy reputation system is discussed which is called **FuzRep_M2**. Each peer in this method possesses an interaction repository which is treated as subjective memory of interactions with other peers. Each record in the repositories contain three attributes: an automatically increasing serial number (SN), the identifier of the interacted peer (ID), and a remark of the interaction (RM). For simplicity, remarks of interactions are either +1, requesting a resource from the interacted peer, or -1, providing a resource to the interacted peer.

To illustrate, Figure 7 is a directed graph that demonstrates interactions among peers where vertices represent peers, and edges are interactions directed from requesters to providers. Note that contribution information is managed by interacted entities instead of the actors, which can avoid actors tampering their behavior information.

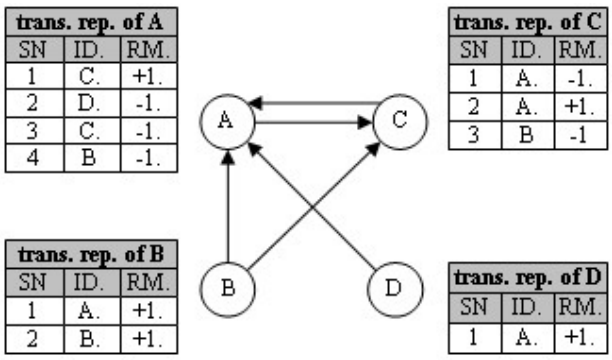


Figure 7: An example of reputation storage

Now, if interactive behavior has been properly recorded, I am able to determine a peer's reputation by this information. The reputation computation in FuzRep_M2 is two-fold. Firstly, it will calculate $R_{local}(Q)$, a local reputation value of a requester Q, based on the provider's subjective experience as follows:

$$R_{local}(Q) = \sum_{\forall sn \in SN_Q} \left(\frac{sn \times RM_{sn}}{\sum_{\forall sn \in SN_Q} sn} \right) \quad (1)$$

where SN_Q denotes the set of serial numbers of records about Q and RM_{sn} is the remark of the interaction of the record of the serial number sn in the provider's interaction repository. In fact, the $sn \times RM_{sn}$ in the $R_{local}(Q)$ is an *aging strategy* in reputation calculation: the most recent interactions represent more about the requester. The provider, then, will formulate $R_{global}(Q)$, a global aggregated reputation value of peer Q, from opinions of others in the same communities as in the following form:

$$R_{global}(Q) = \alpha \times R_{local}(Q) + (1-\alpha) \times \frac{\sum \forall r \in \vec{R}\, r}{\left|\vec{R}\right|} \quad (2)$$

where α is a weight of self-confidence and $\vec{R}$ is a set of collected response values of peer Q's contribution within the community from chosen interacted peers.

Notice that, in the above, if there is no interaction record about the requester in the interaction repository and no responses from approached peers, the requester would be treated as a newcomer where $R_{global}(Q)$=0 by default. As for the implementation of getting $\vec{R}$, it is related to the reputation aggregation process, namely selective polling.

6.3 Selective Polling

To concrete a consensus of a global reputation value towards requester without supports from hubs of information or pre-sorting schemes, a lightweight mechanism is that resource providers can get second opinions from members in the same interest group via polling interacted entities listed in their local interaction repositories. Moreover, the polling messages will be further relayed to the other members in the same community. The idea of the community-oriented scheme can be further understood by the following example. In Figure 8, the provider P presumes that the requester R, as well as the interacted peer F,

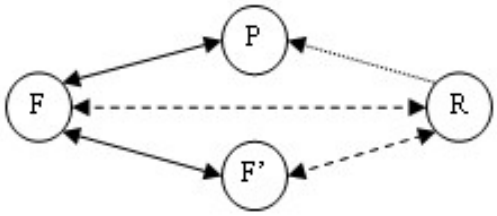


Figure 8: The triangular relationship of interactions. The dotted line represents a request; the concrete lines are conducted interactions, and the broken lines are deduced interactions

has the same interest as himself. Therefore, it is reasonable for *P* to expect that *F* and *R* have similar interests and to deduce that they may have interacted before. Thus, *P* will then approach *F* for the contribution information of *R*. Similarly, *F* will have the same expectation toward his interacted peer *F'*.

Although they are all interest-related in the same community, reaching all of the peers listed in the interaction repository is inefficient and costly; more importantly, that can evoke frauds and collusions in reputation systems. To mediate this issue, resource providers should selectively approach interacted entities that are considered similar to themselves. We pose a simple formula to measure the similarity of two peers *i* and *j*, *sim*(*i*,*j*), using local data only. In the following formula, let *p* be the number of positive remarks of records about peer *j* in peer *i*'s interaction repository and *n* the number of negative ones. The measurement of the similarity between *i* and *j* is given by

$$sim(i,j)=\begin{cases}0 & if\ n=0\ or\ p=0\\ p/n & if\ n>p\\ n/p & if\ n\le p\end{cases} \quad (3)$$

Values of *sim*(*i*, *j*) range from 0, most dissimilar, to 1, most similar. We argue that the similarity of two peers is directly related to the evenness in the number of uploading and downloading interactions among them. Although we have not found any studies to support the argument above, the reciprocity interactions clearly reflect the strength of relationship.

Then, the pseudo-code below constructs our selective polling scheme.

The recursive procedure takes three arguments: the identifier of the requester (*PeerID*), a threshold of similarity (*Threshold*), and a fading similarity (*Fading*). Clearly we need the *PeerID* to indicate who the target peer that is going to be evaluated.

In addition, the initiator of a selective polling message has to set a threshold to stop spreading polling messages when the *Fading*, which is 1 initially and decreases as polling messages spreading out as Figure 9 demonstrates, is below the *Threshold*.

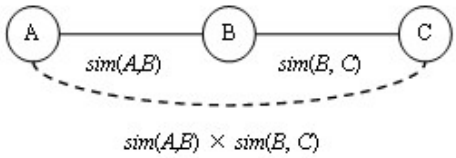


Figure 9: Calculating indirect similarity is to multiply all similarity values in between

The rationale behind the selective polling scheme is to search the referral network of a provider for a requester's contributions in a P2P community. Characteristics of community, higher interaction frequency and strong social ties among community members, make it a unique platform to acquire contribution information as well as resources such as files.

6.4 Service Differentiation

We have just presented how to store, compute, and aggregate reputation. To clearly provide incentives for cooperation, we further propose a reputation usage scheme which is related to service differentiation. Service level of each request is determined by a reference matrix, as shown in Figure 10, comprised of two metrics, the global aggregated reputation value of the resource requester as the reputation and the available inbound bandwidth of the resource provider as the capability.

Figure 10 reflects that the service level has a direct, proportional relation with the reputation and with the capability.

---

pseudo code *Selective_Polling*(*PeerID, Threshold, Fading*)

1. *Let m* be the peer currently processing selective polling;
2. *Let* A_m be the set of *ID*s in *m*'s interaction repository;
3. *Let* B_m be the subset of A_m where *sim*(*m*,*PeerID*) × *Fading* > *Threshold*;
4. Compute R_{local}(*PeerID*);
5. **if** ($B_m == \phi$)
6. **return** $\alpha \times R_{local}$(*PeerID*);
7. **else**
8. Send *Selective_Polling*(*PeerID, Threshold, Fading* × *sim*(*m*,*PeerID*)) to each peer *k* belong to B_m;
9. Compute R_{global}(*PeerID*);
10. **return** R_{global} (*PeerID*);

During the fuzzy inference stage of decision making, both metrics are degrees of linguistic variables transformed from crisp values by membership functions. Figure 11 demonstrates the three membership functions for reputation, capability, and service level respectively.

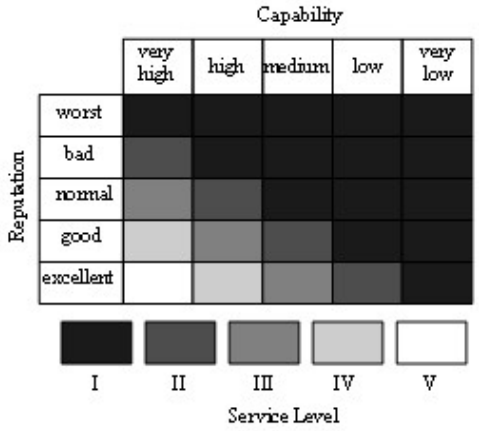


Figure 10: Reference matrix. The lightness of the squares represents service level

Observe that, in Figure 11(c), an output value of service level is within [0, 1]. A transmission speed for a request is the value of the service level by the maximum uploading speed it is willing to offer per request. If the requester's reputation is worst, or if the provider is about to overload, the output value will be 0 which will refuse the file transfer.

The experiment on the service differentiation is under construction by using a system simulation. MATLAB Fuzzy Logic Toolbox provides a convenient way to design and simulate fuzzy logic systems.

7 Conclusion

In this paper, a reputation system, FuzRep, is presented in P2P networks to create incentives for file sharing and to avoid overloading for hotspot peers. The experiments and data presented above indicate that FuzRep is able to create incentives in P2P networks. It is a simple, cost efficient and effective mechanism. By well incorporated with selective polling and service differentiation, FuzRep brings a new idea to alleviate free-riding behaviors.

Another method for the reputation system is also proposed, which creates incentives for cooperation among peers. Requiring no hubs of information or specific network structures for the implementation makes it a neutral P2P reputation system. The reputation system does not rely on physical infrastructures but abstract one.

The future work would aim at two directions. First, to provide higher adaptability, we still want to optimize settings of the reputation system under dynamic P2P network environments. Second, the reputation system should safeguard reputation values from cheating behaviors which are not addressed in the paper.

References

[1] E. Adar and B. A. Huberman, "Free Riding on Gnutella," https://doi.org/10.5210/fm.v5i10.792, *First Monday* 5(10), 2000.

[2] C. Buragohain, D. Agrawal, and S. Suri, "A Game Theoretic Framework for Incentives in P2P Systems," *Proceedings* of the *International Conference on Peer-to-Peer Computing*, pp. 48-56, 2003.

[3] E. Damiani, DCD. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks," https://doi.org/10.1145/586110.586138, *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 207-216, 2002.

[4] E. Friedman and P. Resnick, "The Social Cost of Cheap Pseudonyms," *Proceedings Journal of Economics and Management Strategy* 10(2), pp. 173–199, 1998.

[5] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust Incentive Techniques for Peer-to-Peer Networks," *Proceedings of the ACM Conference on Electronic Commerce*, https://doi.org/10.1145/988772.988788, pp. 102-111, May, 2004.

[6] M. Gupta, P. Judge, and M. Ammar, "A Reputation System for Peer-to-Peer Networks," *Proceedings of 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, https://doi.org/10.1145/776322.776346, pp. 144-152, 2003.

[7] D. Hughes, G. Coulson, and J. Walkerdine, "Free Riding on Gnutella Revisited: The Bell Tolls?" *Proceedings IEEE Distributed Systems Online* 6(6),

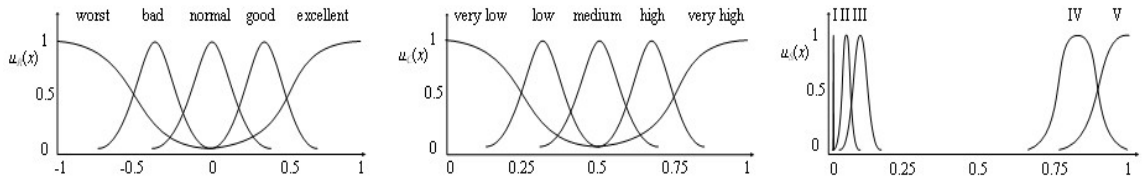


Figure 11: Membership function of (a) reputation, (b) capability, and (c) service level. Each of them has five different labels to represent the linguistic variable

https://doi.org/10.1109/MDSO.2005.31, pp. 1-1, 2005.
[8] A. Iamnitchi and I. Foster, "Interest-Aware Information Dissemination in Small-World Communities," *Proceedings of the 14th IEEE International Symposium on High Performance Distributed Computing*, pp. 167-175, 2005.
[9] A. Iamnitchi, M. Ripeanu, and I. Foster, "Small-World File-Sharing Communities," DOI: 10.1109/INFCOM.2004.1356982 , *Proceedings of INFOCOM'04*, pp. 952-963, 2004.
[10] S. D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," https://doi.org/10.1145/775152.775242, *Proceedings of the International World Wide Web Conference*, pp. 640–651, 2003.
[11] M. Khambatti, K. Ryu, and P. Dasgupta, "Structuring Peer-to-Peer Networks Using Interest-based Communities," *Proceedings of International Workshop on Databases, Information Systems and Peer-to-Peer Computing*, pp. 48-63, 2003.
[12] A. Odlyzko, "The Case Against Micropayments," *Financial Cryptography 2003, J. Camp and R. Wright, eds, Lecture Notes in Computer Science, Springer*, pp. 77-83, 2003.
[13] K. Ooi, S. Kamei, and T. Mori, "Survey of the State of P2P File Sharing Applications," *NTT Technical Review*, pp. 24-28, 2004
[14] G. Sakaryan, H. Unger, and U. Lechner, "About the Value of Virtual Communities in P2P Networks," *Proceedings of the 4th IEEE International Symposium and School on Advanced Distributed Systems (ISSADS)*, pp. 170-185, 2004.
[15] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient Content Location Using Interest-Based Locality in Peer to Peer Systems," DOI: 10.1109/INFCOM.2003.1209237, *Proceedings of the INFOCOM'03*, pp. 2166-2176, 2003.
[16] N. Szabo, "Micropayments and Mental Transaction Costs," *Proceedings of the 2nd Berlin Internet Economics Workshop*, pp.
[17] *The Gnutella Protocol Specification*, v0.4, 2001. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.
[18] The Mathworks, Inc., *MATLAB 6.5*, 2001.
[19] V. Vishnumurthy, S. Chandrakumar, and EG. Sirer, "KARMA: A Secure Economic Framework for P2P Resource Sharing," *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, 2003.

Ming-Chang Huang obtain his bachelor degree from Tsing Hua University in Taiwan in 1987. He holds his MS degrees both in Computer Science and in Industrial Engineering from the University of Wisconsin at Madison in 1992 and 1994 separately. Dr. Huang received his Ph.D. in Computer Science and Electrical Engineering from the University of Wisconsin – Milwaukee in 2003. He joined the University of North Carolina at Charlotte in August 2011. Currently he teaches in the Belk College as an associate professor. His research interests focus on Web and Network application programming design, Big Data Analysis, load balancing in network and distributed systems, Network and Information Management, Network system performance evaluation.

Scalable Correlated Sampling for Join Query Estimations on Big Data

Feng Yu*, David S. Wilson, Tasha M. Wells
Youngstown State University, Youngstown, OH 44555, USA

Mohammad A. Hamdi
Najran University, Najran, Saudi Arabia

Xiaolan Huang, Wen-Chi Hou
Southern Illinois University, Carbondale, IL 62901, USA

Abstract

Estimate query results within limited time constraints is a challenging problem in big data management. Query estimation based on simple random samples performs well for simple queries, such as selections; however, it often produces high relative errors for complex join queries. Existing methods only work well with limited types of joins, such as foreign key joins, and the sample size can grow dramatically as the dataset gets larger. This research proposes a scalable sampling scheme for join query estimations, namely correlated sampling in map-reduce, that can speed up search query length results, provision precise join query estimations, and minimize storage costs when presented with big data. Extensive experiments with large TPC-H datasets in map-reduce show that our sampling method produces fast and accurate join query estimations on big data.
Key Words: Approximate query processing; sampling; join; map-reduce.

1 Introduction

Big data is everywhere. Approximately 2.5 quintillion bytes (or 2.5 billion Gigabytes) of data is produced each day. Ninety percent of the data created in the world has been created in the past two years [10]. To put that into perspective, IBM created the IBM Model 350 Disk File in 1956. It was the size of a compact-size car and had a storage capacity of five megabytes. If one were to place these machines side by side, based on the amount of data we use in one day, they would circle the earth nine thousand one hundred and ninety times. With the sizes of company databases reaching terabytes and even petabytes, and at the speed of which this data is being accumulated, the need for fast query processing has never been so high.

Generating accurate and fast answers for complex queries on big data is always a critical challenge. Recently, the research field called *Approximate Query Processing* (or *AQP*) [3, 5] has drawn much attention. Instead of computing for the exact answers from the original database, AQP aims to quickly provide approximated answers to queries using statistical methods, such as sampling. AQP is extremely beneficial for many applications, such as data mining and machine learning, especially when the volume and velocity of data increased exponentially.

Query optimization [9] is the process of using statistics about the database, as well as assumptions about the attribute distributions, to acquire the lowest-cost plans for submitted queries. Some databases and data streams are so large and fast that queries can take minutes, hours, even days to process. The AQP research aims to develop statistical structures, called synopses, which can summarize the original large datasets into smaller data for query estimations. CS2 [20] is a sample-based synopsis for centralized databases that aims to provide a fast and precise result size estimation for queries with joins and selections. However, CS2 is designed for centralized databases that can only handle a limited volume of data. The aim of this work is to extend the methods of CS2, apply them to join query estimations on big data, and present the findings through empirical studies.

The contributions of this work are as follows:

1. We introduce a scalable correlated sampling framework in map-reduce with consideration of the memory constraint for join query estimations.
2. We discuss the join query estimation based on the developed synopsis on big data.
3. We benchmark the performance of the proposed correlated sampling scheme on large TPC-H datasets and present the findings on both the accuracy and speedup for join query estimations.

Compared with the conference version work [19], additional contributions are made including:

- We compare the performance of the synopsis database when deployed in the centralized and distributed settings and provide findings of further optimizing the query estimation speed.
- We include the implementation of correlated sampling and the test queries for experiments in the appendices.

The rest of this work is organized as follows. Section 2 states the background and Section 3 includes the related work.

Section 4 introduces the correlated sampling scheme on big data. Section 5 presents the experiment results of benchmarking CS2 on large datasets. Section 6 concludes the work. The code of correlated sampling and test queries are included in the appendices.

2 Background

2.1 Big Data Management

Big data may be one of the most misunderstood terms in the technology field. It is commonly mistaken as merely a large volume of data. While not entirely incorrect, there is much more to big data then just size. John Mashey of Silicon Graphics, Inc. first coined the term "big data" in 1998, although this is debated [16]. Others had written about big data before this date, but Mr. Mashey was the first that used the term in the context of computing. Even though the term of big data was created in the '90s, it was not until the early 2000's that it took the form of what is considered today. In February 2001, Doug Laney created the three V's of big data, which are Volume, Variety, and Velocity [7].

The Apache Hadoop framework [18] is considered as a prominent big data management platform that consists of multiple modules, each having its own distinctive responsibilities. Hadoop Common is the storehouse for other Hadoop modules. Hadoop Distributed File System [2], or HDFS, deals with the storage of data of a Hadoop cluster. A major issue with storing large sets of data in a distributed setting is hardware failures. HDFS is built to combat this, by using a process called data replication. A typical HDFS cluster consists of a name node, which stores the metadata of all files stored, and multiple data nodes, which hold all of the actual data. Each data file stored in HDFS is distributed into multitude blocks, with each block of data being duplicated into multiple different data node locations in the cluster. If at any time there is a data node that fails, another block copy shall be available on a different data node.

2.2 Database Systems on Big Data

With the rise of big data, many database systems have been developed on big data for scalable data management and processing such as Hive [14], HBase [17], Dynamo [6], etc. Among them, Hive was created to make it easier for users to be able to use the map-reduce and HDFS in Hadoop without advanced knowledge of distributed computing. As mentioned earlier, Hive uses a similar language to SQL, called Hive Query Language (HQL). With the use of this language, users are able to perform data queries, as well as summarizing and analyzing data. To work with Hive, users can use traditional command-line interfaces of Hive, such as `hive` or `beeline`[1].To simplify the use of hive, Hive Web Interface (HWI) is also available which is a web-based graphical user interface (GUI).

[1] `https://cwiki.apache.org/confluence/display/Hive/HiveServer2+Clients`

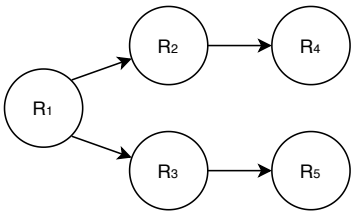


Figure 1: A basic join graph

2.3 Join Graph of a Database

Definition 1. (Join Graph) A join graph [13] is a visual representation of a database in which the flow of joins is explained. It can be created to take into consideration the types of join relationships, such as many-to-many, many-to-one, one-to-one. It is a general representation in which the join relations of a database are mapped out.

Definition 2. (Joinable Relations) Two relations considered joinable, R_i and R_k, $i \neq k$, when there is a path with length ≥ 1 between the relations R_i and R_k.

Definition 3. (Joinable Tuples) Under the assumption that R_i and R_k is a joinable relation, a tuple in R_i, denoted by t_i, and a tuple in R_k, denoted t_k, is considered joinable if t_i can find a match t_{i+1} in R_{i+1}, t_{i+1} can find a match t_{i+2} in R_{i+2}, ..., and t_{k-1} can find a match t_k in R_k.

Figure 1 is a basic join graph of a database. It shows that a relation R_1, has joinable attributes with a relation R_2, as well as a relation R_3. R_2 has joinable attributes with R_4, but does not have any joinable attributes with R_3 or R_5. R_3 has joinable attributes with R_5, but no joinable attributes with R_2 or R_4. In addition, $R_i \rightarrow R_j$ denotes a many-to-one or one-to-one join relationship from R_i to R_j.

3 Related Work

Random sampling has been widely adopted for query size estimation. Simple Random Sample Without Replacement (SRSWOR) [8, 11] has previously been tested for query selectivity estimation. An SRSWOR of each relation is taken separately, and then the resulting independent samples are joined for estimations. These estimations often end in massive errors for join size estimations simply because random samples of separate relations do not retain join relationships [4]. SRSWOR is beneficial if one is only seeking to get a cardinality estimation on an individual relation.

While the Join Synopses (JS) [1] uses SRSWOR in its mechanics, the process does add join correlations between individual relations, causing a much lower relative error. JS uses foreign key joins and procures samples of *all possible joins* in a schema. These samples are then stored, and joined with individual SRSWOR relations to form a finalized set of correlated random tuples that can be used for unbiased query estimations. A major drawback of JS is compared with CS2 is

the sampling complexity. It requires SRSWOR on each relation in the database followed by correlated sampling on each joinable relation along the path in the join graph. For a path of the join graph consists of n relations starting from R_1 to R_n, $O(n^2)$ sampling operations will be performed to generate a JS. i.e. the accumulation of total sampled relations grows quadratically as the total relations of a sampling path increase for JS, while linearly for CS2. Also, JS can only work with foreign key joins due to its sampling restrictions.

Figure 2a shows an example of the Join Synopses process and how preparations for a join estimation is made. For any of the three relations, an individual relation can be estimated by using just SRSWOR. These are denoted in the figure as S_1^*, S_2^*, and S_3^*, in which the '$*$' signifies SRSWOR. The relations needed to calculate a join query size estimation of R_1 and R_2, would be a simple random sample of R_1 and a correlated sample of R_2, denoted by S_1^* and S_2^1, needed. To get the join query size estimation of a join between R_2 and R_3, a simple random sample of R_2 and a correlated sample of R_3, denoted as S_2^* and S_3^2, are needed. Finally, the relations needed to estimate a join query size estimation for R_1, R_2, and R_3, shall include S_1^*, S_2^1, and a correlated sample of R_3 with R_1 as the source relation, denoted by S_3^1.

Tuple Graph [12] (or TuG) is a graph-based synopsis viewing the relational data as the semi-structured data, namely XML. TuG abstracts the tuples in a relation as graph nodes and summarizes the join relationships between tuples using graph edges. After that, the graph edges are further compressed in TuG for efficient storage. For selection estimations, TuG summarizes the distributions of selection attributes by using histograms. One limitation of TuG is it takes much more time to be generated on databases with complex join relationships. Another is the accuracy of TuG can be poor given a tiny synopsis space budget since the join graph and histograms are extremely compressed.

4 Correlated Sampling on Big Data

4.1 Correlated Sampling

Correlated Sample Synopsis (or CS2) [20] is a statistical summary for a database and can be used for both query estimation and approximate query processing (AQP). The purpose of CS2 is to create an unbiased, fast, and precise estimation for queries with all types of joins and selections. CS2 preserves join relationships between tuples and their relations by the unique sampling scheme called correlate sampling which collects correlated tuples that are joinable with previously sampled tuples in the synopsis. Unlike JS, CS2 doesn't require SRSWOR on every relation in the join graph but employs a special value called Joinable Tuple Sample Ratio (or JR) with a Reverse Estimator (or RV Estimator) to provide unbiased join query estimations.

Figure 2b illustrates a sample example of the correlated sampling that transpires once the source relation and sampling path are decided. An SRSWOR is first performed on the source relation R_1, denoted as S_1^*. The next relation, denoted by R_2, is now ready to be sampled. To create the correlation between relations and preserve the join relationships, S_1^* is joined with R_2, with the results being placed into a second sample relation, denoted as S_2. Relation R_3 is then joined with S_2 with the results being placed in the third sample relation, denoted as S_3. The combination of all of the sample relations and JR values is considered the CS2 synopsis, denoted by $\mathscr{S}$.

4.2 Sampling in Map-Reduce

In big data file systems, such as HDFS, data access is required to be translated into operations of the map and reduce procedures. Sampling operations on traditionally centralized database systems are not exceptional when converted to the environment of map-reduce. JS and CS2 preserve the joinable relations of tuples between sampled relations by performing join operations which are categorized into two different operations in big data, namely map-join and reduce-join.

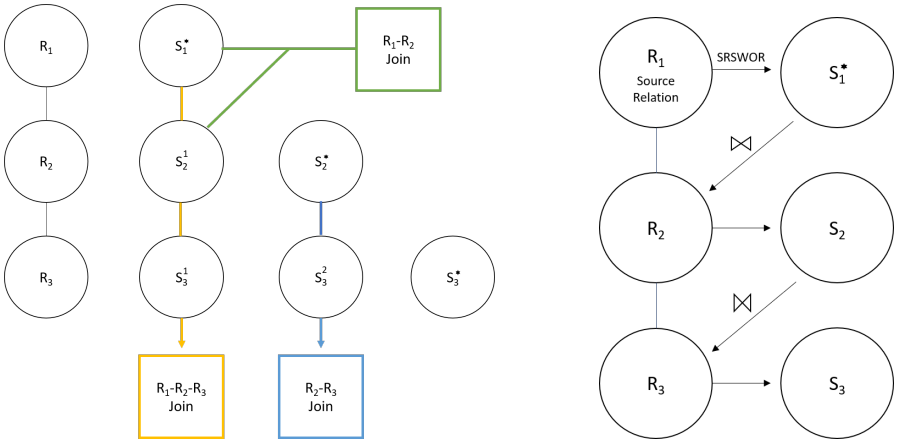
Assume the memory heap of each data node in the big data cluster is limited to a threshold value of T_m. Given two relations R and S, when joining R and S, denoted by $R \bowtie S$ in traditional databases, if R is smaller than S and can be fit into the memory heaps of data nodes, i.e. $|R| \leq T_m$, then R is mapped to all data nodes where S is distributed and a *map-join* is performed, denoted by $R \overset{m}{\bowtie} S$. On the other hand, if both relation tables are too large to be fit into memory heaps, i.e. $|R| > T_m$ and $|S| > T_m$, then a common map-reduce procedure will be initiated to compute the join result, called *reduce-join*, denoted by $R \overset{r}{\bowtie} S$. Given the same data, a reduce-join is usually more resource and time consuming compared to a map-join [18]. To mitigate the sampling cost, correlated sampling on big data aims to control the sample size small enough and use map-join as much as possible during the process.

4.3 Source Relation and Join Graph Path Selection

Correlated Sampling begins with the SRSWOR on a source relation of a join graph of the database. It is important to note, CS2 does work with most join relationships (one-to-many, many-to-one, many-to-many). However, when selecting the sampling paths and source relations, it is suggested to follow a many-to-one or one-to-one relationship as much as possible since following a one-to-many or many-to-many relationship may cause the synopsis to grow considerably when high data skew presents. In a complex join graph, multiple source relations can be chosen, which can simplify the sampling procedure and increase the randomness of sampling. For a complicated join graph, multiple source relations are allowed to follow many-to-one-relationships.

4.4 Correlated Sampling in Map-Reduce

Algorithm 1 shows the process of correlated sampling in map-reduce. For a complicated join graph, multiple source relations are allowed and the join graph can be partitioned into



(a) Sampling of join synopses (b) Correlated sampling of CS2

Figure 2: Join synopses and correlated sampling

Algorithm 1: Correlated sampling in map-reduce

Input: G — Join Graph of the Database; n_a — Sample Size
for $R_a \in Source\_Relations(G)$ **do**
  $S_a = \emptyset$, $S_i = \emptyset$ $(\forall i \neq a)$
  S_a=Map-Reduce(SRSWOR(R_a,n_a))//simple random sampling on R_a
  $W = \{R_a\}$//mark relation as visited
  while $\exists$ *unvisited edge* $\langle R_i, R_j \rangle$ *with* $R_i \in W$ **do**
    S_j=$\Pi_{R_j}(S_i \overset{m}{\bowtie} R_j)$//sample the next relation
    $W = W \cup \{R_j\}$ //mark R_j as visited
  end
  $\mathscr{S}$=Reduce($\{S_a\} \cup \{\cup_{j \neq a} S_j\}$)
end
return $\mathscr{S}$ — *Generated CS2 in Map-Reduce*

multiple join graph paths. For each source relation R_a in a join graph path, an SRSWOR is first performed by a map procedure with n_a tuples sampled from R_a. In big data, the SRSWOR is implemented by first mapping the randomness to all worker nodes, and then reduce tuples randomly selected by each worker together to generate the final sample output. Since the most time-consuming randomness process is distributed into many map jobs, the entire workload is dramatically optimized.

To retain the joinable relations, the correlated tuples are collected in R_j when map-joined with S_i, which is the parent joinable relation along the join path. Note that, the sampling path in CS2 is recommended to follow many-to-one relationships; therefore, the size of S_j shall be no larger than S_i and the map-join can be continued along the join path while the sampled relation size shall generally decrease. Finally, a reduce function is initiated to collect all sampled relations into $\mathscr{S}$ as the generated CS2 synopsis in map-reduce. Section 5 includes the details of implementing correlated sampling in Hive.

4.5 Query Estimation

The process of query estimation is taking the query result running on the synopsis database, and using said results to estimate query result on the original database. In this work, we focus on estimating join queries with selection filters and categorize them into the source and no-source queries. If the set of operand relations of a query includes the source relation of the synopsis, then it's called a *source query*; otherwise, it's called a *no-source query*.

4.5.1 Source Query Estimation. Referring back to Figure 2b, a source query would be considered a join of relations R_1 and R_2, or a join between relations R_1, R_2, and R_3, where R_1 is the source relation.

Given S_1^*, an SRSWOR of the source relation R_1, the estimation of the query result is estimated by

$$\widehat{Y}_{\text{source}} = \frac{N_1}{n_1} \sum_{i=1}^{n_1} y_i \tag{1}$$

where $N_1 = |R_1|$ and $n_1 = |S_1^*|$, and y_i is the number of result tuples generated by the ith tuple of S_1^*.

4.5.2 No-Source Query Estimation. In Figure 2b, a join of R_2 and R_3 would be considered a no-source query, since R_1 is not included as an operand relation. Due to the conditions of a no-source query not containing an SRSWOR based off the source relation, additional steps must be taken for accurate estimation. Joinable Tuple Sampled Ratio (or JR) [20] is a value obtained in the procedure of backtracking joinable relationship of each correlated sample tuple to the source relation (reverse sampling) and supplying it with the ability to estimate the size of a no-source query.

Given R_h the top relation in a no-source query and S_h the

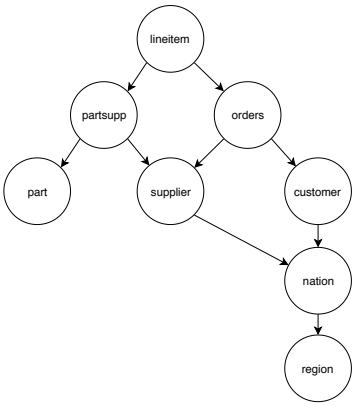


Figure 3: Sampling graph of the TPC-H dataset

correlated sample of R_h, the query result is estimated by

$$\widehat{Y}_{\text{no\_source}} = \frac{N_1}{n_1} \sum_{j=1}^{n_h} r_j y_j \tag{2}$$

where $n_h = |S_h|$, y_j is the number of result tuples generated by the jth tuple, u_j, in S_h, and r_j is the JR value associated with u_j, which equals its total number of joinable tuples in S_1^* divided by its total number of joinable tuples in R_1.

5 Experiments

5.1 Experiment Setup

A cluster of five nodes is constructed in the Sarah Cloud running in YSU Data Lab[2]. This cluster consists of two master nodes and three worker nodes. The first master node has four Intel Xeon CPU's (E5-2630 v4 @ 2.20 GHz) and 16GB of RAM. The second master node has two Intel Xeon CPU's (E5-2630 v4 @ 2.20 GHz) and 10GB of RAM. All worker nodes consist of the same setup, an Intel Xeon CPU (E5-2630 v4 @ 2.20 GHz) processor and 8GB of RAM. The cluster is running Apache Hadoop and Hive with Tez.

Two datasets are used, both datasets are generated using the TPC-H benchmark [15]. The first dataset created has a total size of 1GB. The second dataset created has a total size of 10GB. Each dataset holds eight relations, including `Lineitem`, `Customer`, `Orders`, `Partsupp`, `Part`, `Supplier`, `Nation`, and `Region`.

The following steps are taken to prepare for experiments.

Step 1. The source relation as well as the sampling paths must be decided. The `Lineitem` table holds the most many-to-one relationships, and is selected as the source relation. The sampling paths chosen are as follows:

`Lineitem` → `Orders`, `Lineitem` → `Partsupp`, `Orders` → `Customer`, `Partsupp` → `Part`, `Partsupp` → `Supplier`, `Customer` → `Nation`, `Nation` → `Region`

[2]http://datalab.ysu.edu

Step 2. A new table must be created to store the samples of the source dataset. The 1GB and 10GB datasets are denoted as `tpch1g` and `tpch10g` respectively. The sample datasets are denoted as `s_tpch1g` and `s_tpch10g` respectively.

Step 3 (Source Relation Sampling). Before creating the SRSWOR, a sample dataset size must be selected. The decision was made that the sample dataset size would be one percent of the source dataset. The HQL to create the SRSWOR is as follows:

```
create table s_tpch10g.lineitem as
select * from tpch10g.lineitem
where rand() <= 0.01
distribute by rand() sort by rand();
```

The HQL lines `distribute by rand()`, and `sort by rand()` were added to create a higher rate of randomness. `distribute by rand()` takes the entire set of tuples from a table and distributes them randomly to different reducers. `sort by rand()` then takes these sets of random tuples and sorts them randomly on each reducer.

Step 4 (Correlated Sampling). Using the created SRSWOR, and following the join graph path, the rest of the sample relations are constructed. The HQL code that was used to create the sample relations are included in Appendix A.

5.2 Query Estimation Tests

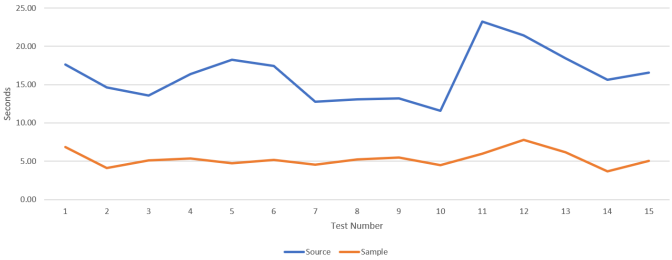
Overall, a total of 15 queries were generated based on the TPC-H benchmark. They were tested five times each, over both the 1GB and 10GB source dataset, as well as the 1GB and 10GB sample dataset. The test queries used in this process are included in Appendix B.

The original dataset is denoted as the "source" on all graphs, and the correlated sample dataset is denoted as "sample". Discussion about the results for the 1GB dataset will be presented first, followed by the 10GB dataset results, and finishing with discoveries made through the testing phase. The datasets were tested on the speed of queries, as well as the accuracy of the estimations.

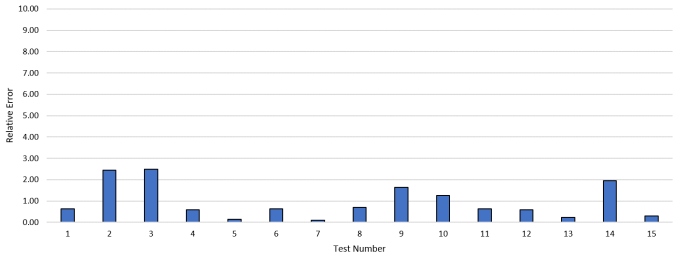
For accuracy tests, we compared the estimated query results by CS2, denoted by $Q_{\text{estimated}}$, with the ground truth query results from the source database, denoted by $Q_{\text{ground\_truth}}$, and calculate the absolute relative error. The formula of the absolute relative error is given by:

$$\text{Absolute Relative Error} = \left| \frac{Q_{\text{ground\_truth}} - Q_{\text{estimated}}}{Q_{\text{ground\_truth}}} \right| \times 100\%$$

5.2.1 Query Estimation Tests on 1GB Dataset. Depicted in Figure 4, the 1GB dataset source dataset tests averaged a total query time of 16.26 seconds, with a high average of 23.24 seconds in Query 11, and a low average of 11.57 seconds in Query 10. The 1GB sample dataset tests averaged 5.33 seconds, with a high average of 7.81 on Query 12, and a low average of 3.71 on Query 14. The average speed up from the sample,

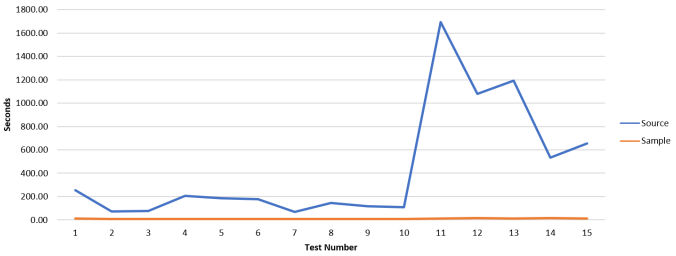


(a) 1GB Dataset Query Search Length Results (Seconds)

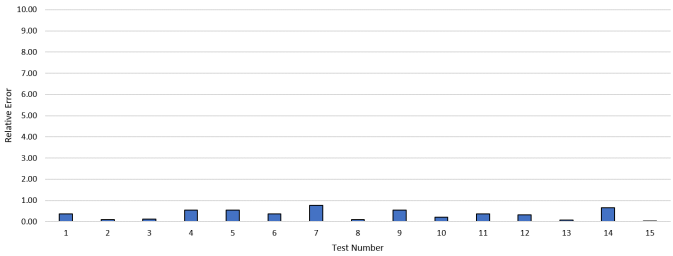


(b) 1GB Dataset Relative Error Results (%)

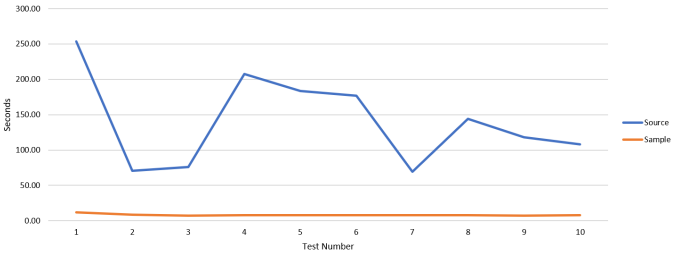
Figure 4: Experiments of 1GB dataset



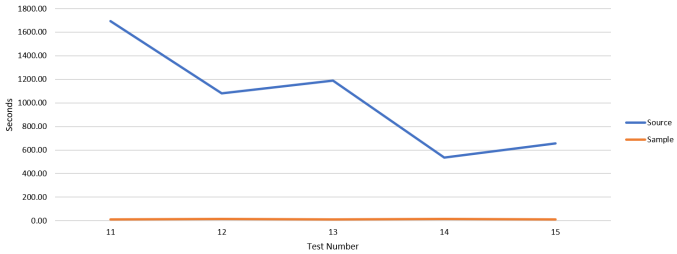
(a) 10GB dataset query search length (seconds)



(b) 10GB dataset relative error results (%)



(c) 10GB dataset two relation join query results (seconds)



(d) 10GB dataset three relation join query results (seconds)

Figure 5: Experiments on 10GB dataset

over the source, would be a 305%. The largest speed up was Query 14 at 321.71%, and the lowest speed up was Query 9 at 141.65%. The results were impressive on the 1GB dataset with the sample dataset processing much faster than the source.

The average count of tuples for the source dataset was 2,520,952. The average count of tuples for the sample dataset was 25,385. The average join estimation results based on the source query estimator was 2,538,533. The average relative error for the 1GB dataset was 0.96%. The highest relative error was 2.50% on Query 3 with the source dataset holding 592,794 tuples, the sample dataset holding 6,076 tuples and source join estimation showing 607,600 tuples. The lowest relative error was 0.09% on query 7, with the source dataset showing 24,877, the sample dataset showing 249, and the source join estimation showing 24,900.

5.2.2 Query Estimation Tests on 10GB Dataset. Depicted in Figure 5, the 10GB dataset source dataset tests averaged a total query time of 437.49 seconds, with a high average of 1692.87 seconds in Query 11, and a low average of 69.57 seconds in Query 7. The average speed up from the sample results over the source results would be a 4,489.44%. The largest speed up was Query 11 at 16,071.90%, and the lowest speed up was Query 2 at 758.39%. This shows that the larger the dataset the better CS2 performs in speed compared with JS.

The average count of tuples for the source dataset was 25,208,072. The average count of tuples for the sample dataset was 251,168. The average join estimation results based on source join estimation was 25,116,820. The average relative error for the 10GB dataset was 0.34%. The highest relative error was 0.76% on Query 7 with the source dataset holding 248.493 tuples, the sample dataset holding 2,466 tuples and source join estimation showing 246,600 tuples. The lowest relative error was 0.01% on query 15, with the source dataset showing 6,047,718, the sample dataset showing 60,474, and the source join estimation showing 6,047,400. The results show that not only does CS2 speed up the larger the data size gets, but its accuracy also improves.

The advantage of CS2 in map-reduce was recognized. Not only does CS2 speed up the join queries, but when moving from a two relation join to a three relation join, the time increase is very minimal for CS2. In Figure 5c, the two-relation join query

Table 1: Query search time

Dataset	Type	Average (sec)	Average Speedup (1.0x)	High (sec)	Low (sec)
1GB	source	16.26	3.05	23.24	11.57
	sample	5.33		7.81	3.71
10GB	source	437.49	45.91	1692.87	69.57
	sample	9.53		14.84	7

Table 2: Synopsis database speed overhead

Query	Hive (ms)	PostgreSQL (ms)	Overhead (%)
1	1048.2	87.6	1197
2	867	50.2	1727
3	1032.8	68.6	1506
4	882.8	59.2	1491
5	963.8	93.6	1030
6	1859.6	146.4	1270
7	1363.2	134.2	1016
8	1204.4	84	1434
9	1394.8	90.6	1540
10	1326.2	84.2	1575
11	4090	90.4	4524
12	1117	93.8	1191
13	1307.2	26.6	4914
14	979.6	59.4	1649
15	1176.4	65.2	1804
overall	1374	82	1858

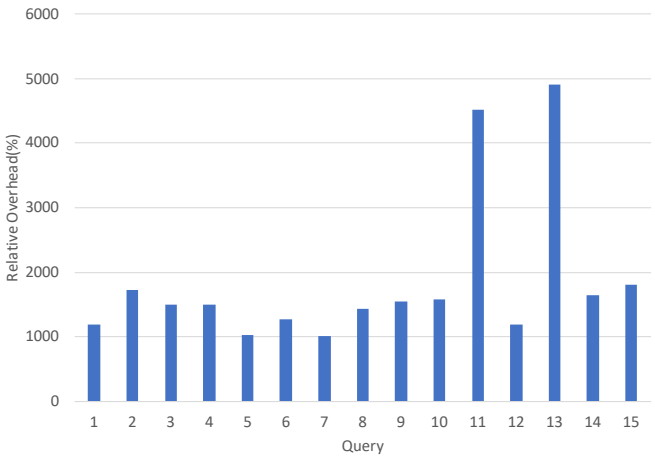


Figure 6: Synopsis database relative speed overhead

results, CS2 holds at an average about seven seconds while the original averages around 180 seconds. When the queries switched to a three relation join in Figure 5d, the average for CS2 bumps up to about 10 seconds, while the original relation explodes and averages about 1,100 seconds per query.

5.2.3 Synopsis Database Performance Tests. In query estimation experiments, the synopsis databases, i.e. the generated CS2 synopses, are in-place by the Hive database in a distributed setting. During the query estimation process, the CS2 synopses are accessed by a map-reduce manner even though their data sizes might be tiny, which may decrease the data accessing speed.

We tested the synopsis database performance improvement by deploying the generated CS2 synopses in a centralized database, namely PostgreSQL[3], which is a prominent relational database. The same generated CS2 synopsis for the 1GB dataset query estimation experiment is exported from Hive into a PostgreSQL database running on a different data server.

We ran the same 15 test queries in previous experiments on both the synopsis databases in Hive and PostgreSQL, generated estimations for the true query results, and compared their running time and relative speed overheads. The detailed running time for the queries are listed in Table 2, with the calculated relative overheads (%). The relative overheads are illustrated in Figure 6. The relative speed overhead is defined as

$$\text{Relative Speed Overhead} = \frac{\text{Time(Hive)}}{\text{Time(PostgreSQL)}} \times 100\%$$

Surprisingly, PostgreSQL enables much faster data accessing speeds than Hive for the synopsis database. In general, the synopsis database in PostgreSQL is more than 18x faster than in Hive. This may be due to the centralized file system is more efficient to access small sized synopsis in this experiment meanwhile the Hive still needs to generate distributed map-reduce jobs on HDFS.

In practice, one may export the synopsis databases from Hive to a centralized database or in-memory database for faster query estimations. Many open-source tools are available for this purpose, such as the Apache Sqoop[4]. A potential drawback is when there are updates on the synopsis database in map-reduce, the centralized synopsis database must also be synchronized which may take additional time.

6 Conclusion and Future Works

In this research, the scalable correlated sampling of CS2 on big data was introduced. It was discovered that not only does CS2 in map-reduce maintain the accuracy of tuples from

[3] https://www.postgresql.org/
[4] https://sqoop.apache.org/

its samples in join query estimations, but also increases in precision as the dataset grows larger. CS2 in map-reduce maintained a constant speed and did not increase much as the datasets expanded in size. When the source relation query search length ballooned in size with the three relation joins, CS2 continued to produce low search query lengths. Based on the results, CS2 in map-reduce proved to be successful in query optimization and more efficient in regards to scalability and accuracy requirements. We also compared the query estimation speeds when the synopsis database deployed in the centralized and the map-reduce environment. The results showed that centralized synopsis database enabled faster query estimation speeds when the synopsis size is small. Future research will seek to efficiently calculate JR in map-reduce proving that CS2 excels with approximate query processing on big data.

Acknowledgement

This research is partially supported by Cushwa Shearing Graduate Fellowship, University Research Council Grant, and Research Professorship Award at Youngstown State University, and Amazon Research and Education Grant.

References

[1] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy, "Join Synopses for Approximate Query Answering," *SIGMOD Rec.*, 28(2):275–286, Jun. 1999.

[2] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design," *Hadoop Project Website*, 11(2007):21, 2007.

[3] S. Chaudhuri, B. Ding, and S. Kandula, "Approximate Query Processing: No Silver Bullet," *Proc. SIGMOD'17*, 2017, pp. 511–519.

[4] S. Chaudhuri, R. Motwani, and V. Narasayya, "On Random Sampling Over Joins," *ACM SIGMOD Record*, 28(2):263–274, 1999.

[5] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine, "Synopses for Massive Data: Samples, Histograms, Wavelets, Sketches," *Foundations and Trends in Databases*, 4(1–3):1–294, 2011.

[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store," *ACM SIGOPS Operating Systems Review*, 41(6):205–220, 2007.

[7] A. Gandomi and M. Haider, "Beyond the Hype: Big Data Concepts, Methods, and Analytics," *International Journal of Information Management*, 35(2):137–144, 2015.

[8] W.-C. Hou, G. Ozsoyoglu, and B. K. Taneja, "Statistical Estimators for Relational Algebra Expressions," *Proc. SIGMOD'88*. ACM, 1988, pp. 276–287.

[9] Y. E. Ioannidis, "Query Optimization," *ACM Computing Surveys (CSUR)*, 28(1):121–123, 1996.

[10] B. Marr, "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read," *Forbes*, Jul 2018.

[11] F. Olken, "Random Sampling from Databases," Ph.D. dissertation, University of California, Berkeley, 1993.

[12] J. Spiegel and N. Polyzotis, "TuG Synopses for Approximate Query Answering," *ACM Transactions on Database Systems (TODS)*, 34(1):1–56, 2009.

[13] A. Swami, "Optimization of Large Join Queries: Combining Heuristics and Combinatorial Techniques," *ACM SIGMOD Record*, 18(2):367–376, 1989.

[14] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive: A Warehousing Solution Over a Map-Reduce Framework," *Proceedings of the VLDB Endowment*, 2(2):1626–1629, 2009.

[15] Transaction Processing Performance Council, "TPC-H Benchmark Specification," Available at http://www.tpc.org/tpch/.

[16] M. Van Rijmenam, "A Short History Of Big Data," `https://datafloq.com/read/big-data-history/239`, 2015.

[17] M. N. Vora, "Hadoop-HBase for Large-Scale Data," *Proceedings of 2011 International Conference on Computer Science and Network Technology*, IEEE, 1:601–605, 2011

[18] T. White, *Hadoop: The Definitive Guide*. O'Reilly Media, Inc, 2012.

[19] D. Wilson, W.-C. Hou, and F. Yu, "Scalable Correlated Sampling for Join Query Estimations on Big Data," *Proc. 28th International Conference on Software Engineering and Data Engineering*, 64:41–50, 2019.

[20] F. Yu, W.-C. Hou, C. Luo, D. Che, and M. Zhu, "CS2: A New Database Synopsis for Query Estimation," *Proc. SIGMOD'13*, New York, NY, USA, 2013, pp. 469–480.

Appendix

A Correlated Sampling HQL Code

Creating Sample Partsupp Table

```
create table s_tpch10g.partsupp as
select distinct partsupp.ps_partkey,
partsupp.ps_suppkey, partsupp.ps_availqty,
partsupp.ps_supplycost, partsupp.ps_comment
from tpch10g.partsupp join
s_tpch10g.lineitem on partsupp.ps_partkey
= lineitem.l_partkey and partsupp.ps_suppkey
= lineitem.l_suppkey;
```

Creating Sample Orders Table

- `create table s_tpch10g.orders as select distinct o_orderkey, o_custkey, o_orderstatus,o_totalprice,o_orderdate, o_orderpriority, o_clerk, o_shippriority, o_comment from tpch10g.orders join s_tpch10g.lineitem on orders.o_orderkey = lineitem.l_orderkey;`

Creating Sample Part Table

- `create table s_tpch10g.part as select distinct p_partkey, p_name, p_mfgr, p_brand, p_type, p_size, p_container, p_retailprice, p_comment from tpch10g.part join s_tpch10g.partsupp on part.p_partkey = partsupp.ps_partkey;`

Creating Sample Supplier Table

- `create table s_tpch10g.supplier as select distinct s_suppkey, s_name, s_address,s_nationkey,s_phone, s_acctbal,s_comment from tpch10g.supplier join s_tpch10g.partsupp on supplier.s_suppkey = partsupp.ps_suppkey;`

Creating Sample Customer Table

- `create table s_tpch10g.customer as select distinct c_custkey,c_name, c_address,c_nationkey,c_phone, c_acctbal,c_mktsegment,c_comment from tpch10g.customer join s_tpch10g.orders on customer.c_custkey = orders.o_custkey;`

Creating Sample Nation Table

- `create table s_tpch10g.nation as select distinct n_nationkey, n_name, n_regionkey,n_comment from tpch10g.nation join s_tpch10g.customer on nation.n_nationkey = customer.c_nationkey;`

Creating Sample Region Table

- `create table s_tpch10g.region as select distinct r_regionkey, r_name, r_comment from tpch10g.region join s_tpch10g.nation on region.r_regionkey = nation.n_regionkey;`

B Testing Queries HQL Code

Query 1

- `select count (*) from lineitem,partsupp where l_partkey = ps_partkey and l_suppkey = ps_suppkey;`

Query 2

- `select count (*) from lineitem,partsupp where l_partkey = ps_partkey and l_suppkey = ps_suppkey and ps_availqty > 9000;`

Query 3

- `select count (*) from lineitem,partsupp where l_partkey = ps_partkey and l_suppkey = ps_suppkey and ps_supplycost < 100;`

Query 4

- `select count (*) from lineitem,partsupp where l_partkey = ps_partkey and l_suppkey = ps_suppkey and l_quantity >= 20;`

Query 5

- `select count (*) from lineitem,partsupp where l_partkey = ps_partkey and l_suppkey = ps_suppkey and l_extendedprice >= 40000;`

Query 6

- `select count (*) from lineitem,orders where l_orderkey = o_orderkey;`

Query 7

- `select count (*) from lineitem,orders where l_orderkey = o_orderkey and o_totalprice >= 400000;`

Query 8

- `select count (*) from lineitem,orders where l_orderkey = o_orderkey and l_quantity < 20;`

Query 9

- `select count (*) from lineitem,orders where l_orderkey = o_orderkey and l_discount = .04;`

Query 10

- `select count (*) from lineitem,orders where l_orderkey = o_orderkey and l_shipmode = 'AIR';`

Query 11

- `select count (*) from lineitem,orders,customer where l_orderkey = o_orderkey and o_custkey = c_custkey;`

Query 12

```
• select count (*) from
  lineitem,orders,customer where l_orderkey
  = o_orderkey and o_custkey = c_custkey and
  c_acctbal > 500;
```

Query 13

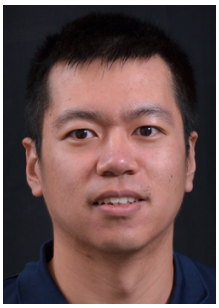
```
• select count (*) from
  lineitem,orders,customer where l_orderkey
  = o_orderkey and o_custkey = c_custkey and
  c_mktsegment
  = 'AUTOMOBILE';
```

Query 14

```
• select count (*) from
  lineitem,orders,customer where l_orderkey
  = o_orderkey and o_custkey = c_custkey and
  l_returnflag = 'A';
```

Query 15

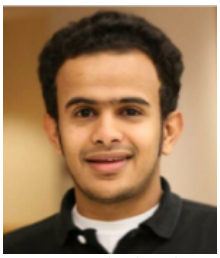
```
• select count (*) from
  lineitem,orders,customer where l_orderkey
  = o_orderkey and o_custkey = c_custkey and
  l_returnflag = 'N'
  and c_mktsegment = "HOUSEHOLD";
```



Feng Yu is an Associate Professor of Computer Science and Information Systems at Youngstown State University. He received his Ph.D. in Computer Science from Southern Illinois University, Carbondale, IL in 2013. His primary research interests include database systems, big data, and cloud computing. He conducts a research lab called Data Lab focusing on data-oriented sciences. He has served as a reviewer for many international conferences, such as DEXA and SSDBM, and scholarly journals, such as ACM TODS, Information Sciences, and DKE. He serves on the committee board of many international conferences, such as DBKDA, CAINE, and CATA. He is a campus champion for NSF XSEDE. His research has been supported by Computer Research Association and Amazon.

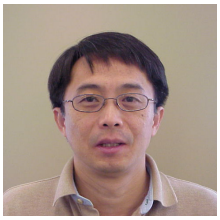
David S. Wilson received his Master's Degree of Computing and Information Systems and Bachelor's Degree of Computer Information Systems at Youngstown State University, Youngstown, OH in 2018 and 2016, respectively. His research interest lies in database systems. His master thesis is on approximate query processing.

Tasha M. Wells obtained her Master of Computing and Information Systems at Youngstown State University, Youngstown, OH in 2018. Her area of interest for research is in database systems. Her work experience includes business intelligence data development and engineering of IT infrastructure systems. She currently works as the Manager of Campus Technology Support for Youngstown State University.



Mohammed Hamdi is an Assistant Professor of Computer Science and Information Systems at Najran University, Saudi Arabia. He received his PhD and MSc in Computer Science from Southern Illinois University-Carbondale in 2013 and 2018, respectively. His main research interests are databases, query optimization, data mining, big data, and crowd-sourcing systems.

Xiaolan Huang is an Assistant Professor in the School of Computing at Southern Illinois Un iversity Carbondale. She received her Ph.D. in Computer Science from Southern Illinois University Carbondale, her M.S. degree in Electrical and Computer Engineering from The University of Texas at Austin, and her B.S. degree from Tsinghua University (China). Her research interests include bioinformatics, big data analytics, machine learning, and high-performance computing. Dr. Huang has published her work in research publications including RNA, Journal of Biomolecular Structure and Dynamics, etc. Her research is funded by NIH. Before pursuing her doctorate, she worked at Broadcom, Cisco Systems in California.



Wen-Chi Hou received his M.S. and Ph.D. degrees in Computer Science and Engineering from Case Western Reserve University, Cleveland Ohio, in 1985 and 1989, respectively. He is a Professor of Computer Science at Southern Illinois University, Carbondale, IL, USA. His interests include statistical databases, mobile databases, XML databases, and data streams.

Evaluation of Game-Theme Based Instructional Modules for Data Structure Concepts

Sarika Rajeev*
American University, Washington, DC, 20016, USA

Sharad Sharma†
Bowie State University, Bowie, MD, 20715, USA

Abstract

Motivation plays a key role in effective learning but motivation needs to be sustained through feedback responses, reflection, and active involvement in order for learning to take place. This work focuses on incorporating active methods of teaching such as game-based learning and simulation-based learning. Game theme-based instructional (GTI) modules produce better results than traditional learning techniques because it increases motivation and engagement of students as they learn interactively. This work is aimed at assessing the motivation and engagement of undergraduate students for GTI modules in introductory data structure courses. This paper discusses the design, implementation, and evaluation of a GTI module to teach the linked list and binary tree data structure. For the design and development of GTI modules, we have incorporated FDF (four-dimensional framework) and constructive approach for learning. The GTI modules were evaluated based on the five components of the Science Motivation Questionnaire II (SMQII). We have also performed an ANOVA test for evaluating the motivation based on familiarity with other virtual reality games and the t-test for evaluating the motivation of students when they use GTI modules as a learning tool. The results of the evaluation of GTI modules shows that instructional modules can efficiently promote learning by encouraging the students' participation.

1 Introduction

With the fast development of computer science and technology, computer games have become one of the integral parts of modern way of life, especially for youngsters. Research studies have conveyed that educational games are motivating, engaging and provide a reliable learning context. The various studies reveal that undergraduate computer science students, using the traditional method of teaching, are not producing better learning outcomes. So, there is a consistent decline in the number of students choosing computer science courses at the undergraduate level [10-11]. Due to the expansion of the educational transform, game theme-based learning methodology has become one of the current research focuses. The educational technologist wants to exploit digital/computer games to create software which can be used for learning purposes. Many researchers suggest that when the learner is actively engaged in the learning process, then it is beneficial to the learning outcomes. The engagement with learning promotes exploration, and conversely, exploration promotes engagement with learning [18].

There is a need to change the traditional passive method of teaching to an active method of teaching such as game and simulation-based learning. Many students choose to use unethical means (cheating, plagiarism, collaboration on assignments) to get a good grade. So, we need to provide a better learning environment by increasing the student's motivation towards learning. The game theme-based instructional (GTI) modules prepare the learners to think critically, and the students can adopt new challenges of the relevant knowledge. Many researches indicate that students are more tempted towards unethical means in education because of a few reasons, which includes: (1) excess workload, (2) trying to get a good grade at any cost (3) introduction to new concepts, and (4) time pressure placed upon students [1, 6]. This paper presents a novel and exciting methodology of learning which includes ethics in future education by motivating the students towards learning. Game theme-based learning helps to improve the problem-solving skills of students and produce better learning outcomes [16, 21]. We had designed and developed GTI modules to teach the linked list and binary tree data structure. Sharma [19. 20, 22], indicates that GTI modules help the students to have a better understanding of concepts by engaging them more towards learning. Woods [25], states that the simulation element of games has educational potential. Squire [24], mentioned that the games have educational potential regarding both subjective and social perspectives.

We have used a proven four-dimensional framework [7] to assess the perceived learning outcomes and effectiveness of GTI modules. The first dimension of the framework emphasizes on "context" of learning. The educational benefits that the students receive in one context may differ from another context. So, the context is an essential factor to consider during the design and development of GTI modules. The context for learning depends on both the micro-level and macro-level factors. The micro-level factors include (1) access to equipment, (2) technical support, and (3) availability of specific resources. The macro-level factors for GTI modules include (1)

*Department of Computer Science. E-mail: rajeev@american.edu,
† Department of Computer Science. E-mail: ssharma@bowiestate.edu.

historical, (2) political and, (3) economical. The context for GTI modules is classroom-based or lab-based. However, it is not limited to use in classrooms; the students can use the GTI modules for self-study.

The second dimension of the framework focuses on the attributes of learning, such as the learner's background, age, learning style, and preferences. For the GTI modules, the learners are undergraduate and graduate students of the computer science department. The age of learners (users of GTI modules) should be between 18 to 26 years. Many studies indicate that the adolescent spends more time with software games than an adult [3]. We have developed GTI modules to motivate young learners towards learning and nurturing them into an intellectual adult. The third dimension of the framework emphasizes the approach of learning. The GTI modules are designed using the constructive approach of learning. The learning process is fair and unbiased when the students construct their knowledge through experience with GTI modules. The fourth dimension of the framework emphasizes the internal representation of the GTI modules. The GTI module is highly interactive and non-immersive. The students' hesitation (shyness) to ask their queries to a real instructor is a major cause of poor learning outcomes and student's unethical approach to learning. We have included a virtual instructor (Figure 1) in the GTI modules which help the students during gameplay.

Figure 1: Virtual instructor in GTI module for teaching linked list

We have evaluated GTI modules using Science Motivation Questionnaire II (SMQII). SMQII is a questionnaire to find the motivation of students. It has five components which include: (1) Intrinsic motivation, (2) Self-determination, (3) Self-efficacy, (4) Career motivation and, (5) Grade motivation. The results of the user study indicate that the students get motivated towards learning and produce better learning outcomes.

3 Related Work

2.1 Motivation and Enthusiasm in Education

Recent studies show that the idea which states that, "there is no relation between emotion and learning," becomes obsolete [4, 8]. According to Damasio [4], emotions not only determine the intelligence but also provide a means to intelligence. According to Goleman [8], if the students are angry, uneasy or distressed, then they will not learn efficiently. So, for better learning outcomes it is necessary to evaluate the motivation and enthusiasm of the learners. Using a self-report approach, we can evaluate the motivation and enthusiasm of students for a specific teaching methodology. Del Soldato [23] introduced a motivational modeler and a motivational planner to measure the level of motivation when students used a traditional intelligent tutoring system for learning new concepts.

De Vincente [5] states that motivation is an essential aspect of the educational software, and it can be analyzed by using a method named 'motivational diagnosis.' Matsubara and Nagamachi [14], state about instructional planning to increase motivation. The likeability is considered as a factor of motivation. According to Keller [12], it is not always possible that if the instruction (traditional lecture by the instructor) is of good quality, then it will motivate the students. There are many pieces of research have been done to deal with motivation [5, 14, 23]. Vincente and Pain [5] presented an empirical study, which showed a significant amount of knowledge related to the motivational diagnosis. As per Nicholls [15], maintaining the optimum motivation in all learners leads to quality and equality in intellectual development.

In AI and educational research, the pupil's motivation is virtually unexplored. In the traditional method of teaching teachers or instructors get cues from the students through a different communication medium such as posture and facial expression. However, many of these cues are unrelated which makes it challenging to evaluate the motivation of students. Vincente and Pain [5] designed a study in which an instructor is asked to conclude pupils' motivational state. For this study, they used the prerecording of students' interaction with the system. The researchers concluded that the evaluation of motivation would be easier if they use the video recording of the students' interaction [5]. The duration of time the students spend with the software is also noteworthy to evaluate the motivation.

2.2 A Constructive Approach to Learning

Constructivism refers to the belief that the learners construct knowledge for themselves via interaction. Constructivists focus more on an understanding of knowledge through experience and less on verifying the concept [17]. The learners are the devoted creator of their knowledge based on the previous idea and new experience. Constructivists believe that the learners are not an empty container to be filled with loads of knowledge; instead, they can make the meaning of the knowledge. The consequence of the theory is twofold.

1. Learners in thinking mode for learning.
2. Learners know that it's not related to their experience.

The constructivist theory appears in response to behaviorism and cognitivism. Researchers believe that a human is not a robot so, he could not be programmed [9]. Constructivists

assert that the human brain plays an active role in learning. In other words, the learners are much more actively involved than teachers or peers to learn or understand a particular concept. Constructivism refers to both learning theory and epistemology [17]. Regarding learning theory, the constructivists suggest that learners construct meaning. In epistemology, constructivists suggest that knowledge is constructed through discussion or interaction with peers.

In constructivism, students construct and reconstruct the concept with an attempt to match with the experience. The new concept is matched with the student's prior knowledge, pre-conception, and misconception. If the new material is consistent, then it is considered to be learned, and if it is contradictory, then it is considered as "it cannot be learned." Per the principles of constructive alignment, as described by Biggs and Tang [2], the assessment tasks and the learning activities are directly related to the learning outcomes of the students.

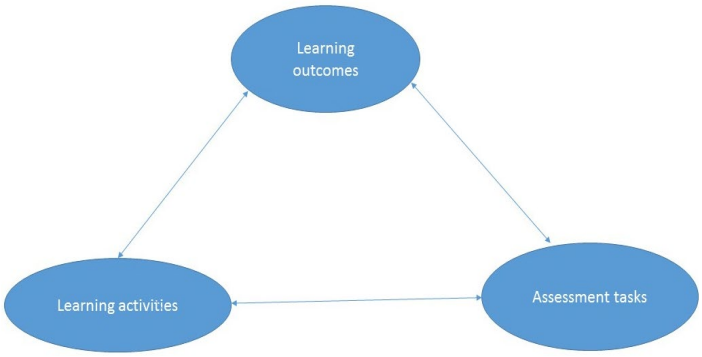


Figure 2: Constructive alignment (Biggs and Tang, 2007) [2]

3 Experimental Design of GTI Modules

This section discusses a brief overview of the activity system and four-dimensional framework, which we have used to design the GTI modules.

3.1 Activity System for GTI Modules

According to Kuutti [13], activity is a form which leads to an object. Various activities are distinguished from each other based on their objects. We have incorporated a primary individual activity system, in our proposed GTI modules. We have utilized the Kuutti's activity system [12] in GTI modules as follows:

1. Tool: GTI modules have a non-immersive 3D-interface. It has a high level of interactivity. In GTI modules users can interact with the virtual instructor to clarify the concepts.
2. Subject: Undergraduate students from the computer science department who have taken introductory programming courses.
3. Object: The objective of GTI modules is to clarify the concept (binary tree and linked list). We want to motivate and engage the students in learning linked lists and binary tree concepts.
4. Rules: We have utilized a constructive approach to learning, in GTI modules. The students construct their knowledge using a gaming metaphor. We have used Bloom's taxonomy in the design of GTI modules.
5. Community: Our proposed GTI modules do not support group study.
6. Division of labor: We have designed GTI modules to be used in the classroom or computer lab. However, students can use GTI modules at their home during leisure time.

3.2 Framework for the Evaluation of GTI Module

We have used a proven framework for finding the effectiveness of GTI modules. The first dimension of the framework focuses on the context where the learning takes place. We want to use GTI modules as a supplement in classroom-based learning. The context for GTI modules is undergraduate classes of computer science. We need limited technical support to use the GTI modules. We also considered the macro-level factors of the context, such as (1) Historical, (2) Political and, (3) Economical. The instructors/tutors must have a specific background such as (1) minimal technical knowledge of computer usage and (2) familiar with virtual reality games. We have developed the GTI modules for classroom-based context, but the use of GTI modules is not limited to the classroom. This educational software tool (GTI module) is platform-independent. So, this software (GTI module) can be used outside the classroom. In other words, the GTI modules are outdoor accessible and; are not limited to classroom or computer lab

The second dimension of the framework, which we have used to identify the effectiveness of the GTI module, focuses on the attributes of learners. From the phrase "learner's specification," the first question that arises in our mind is: Who is the learner? Before designing or developing the GTI modules (educational software), we need to clarify the target audience or users of the GTI modules. The users or learners of GTI modules are undergraduate students from the computer science department. However, the GTI modules can be used in graduate classes of introductory programming. GTI modules do not support group study. The GTI modules are aimed to support individual learners only. According to Provenzo [15], children and adolescents are more fascinated by games. We developed GTI modules for an adolescent of ages 18 years to 26 years. The background and the learning history of the students is also an important factor which affects the effectiveness of GTI modules. The GTI modules support the students of different learning backgrounds/style. There is the various level of games in each GTI modules (binary trees or linked list). The students can choose the game according to their preference and learning style. There is a range of differential learners. The game theme-based instructional module is basically for college students, but this tool can be used by others who want to learn programming concepts (binary trees and linked list) in an informal setting.

The third dimension of the framework focuses on pedagogic

consideration. We need to decide the pedagogic models and approaches to learning before we design the GTI modules. The game theme-based instructional modules are developed with constructive theory in mind. The constructive learning theory states that learners construct knowledge by experiencing it in the real world. When the undergraduate students of the computer science department use GTI modules for learning the concepts of binary trees and linked lists; they interact with the game theme-based instructional module and learn through their experience with the gaming metaphor. The learning activities of GTI modules emphasis on the clarification of the concepts of binary trees/linked list and to produce better learning outcomes. The learning activities include: (1) Interacting with the virtual instructor, (2) Navigating through the GTI modules and, (3) Playing the games of GTI modules. We have used briefing/non-briefing to reinforce learning outcomes.

The fourth dimension of the framework emphasizes the mode of representation or tool for use. Undeniably, this dimension focuses on the processes of learning. This dimension considers models, approaches, and theories to support learning. GTI modules use a medium level of fidelity based upon the use of non-immersive 3D interfaces. The game theme-based instructional modules have a high level of interactivity. GTI modules have buttons through which the learners interact with the virtual environment. Students or learners use mouse clicks and keyboard commands for interacting with the GTI modules. The way the students can interact with the GTI modules are listed as follows:

1. Students can ask for help using the buttons. They can go to the main menu and switch to another game anytime.
2. During gameplay, if the students need to learn about the dedicated concept, then the students can end game ask the virtual instructor (using mouse and keyboard command) for a better understanding of the concepts they want to learn.
3. Students can skip the tutorial if they are already familiar with the concept and directly move to the game of their choice.
4. Students can also answer the popped-up questions during gameplay. These questions are rewarding, and there is a scoreboard that displays the student's score. So, the students can evaluate themselves while learning.

The learning activities and outcomes are achieved partly by playing games and partly by interacting with the virtual instructor. The GTI modules use a medium level of realism regarding the 3D non-immersive virtual environment and 3D model of virtual instructors. The GTI modules can be easily converted into an immersive virtual environment having a high level of realism using virtual reality hardware devices such as Oculus Rift.

4 GTI Implementation

The GTI modules were implemented in two phases, such as: (1) We created the 3D-models using 3d-max and sketch-up. (2) We exported the 3D models in Vizard and added game mechanics using Python.

4.1 Implementation of the Linked List and Binary Tree Module

In GTI modules, the user is greeted with a brief overview of the concepts related to the linked list/binary trees. A 3D avatar of the virtual tutor explains the basic concepts related to a linked list/binary tree in brief. Users are guided on the screen to proceed to the next steps. The students can press the "enter key" to go to the next screen. There is a real-life example of the train to show the linked list. Students can add cars to the trains and construct their knowledge of how to create a linked list. We also implemented a real-world example of the circular linked list by creating a group of a woman standing in a circle. As shown in Figure 3, all-women held each other's hand; students can visualize there is no null pointer (defined end) and can construct their knowledge about the difference in a linked list and circular linked list. We have used assorted interface elements to make the linked list/binary tree modules more attractive.

Figure 3: A real-life example of a circular linked list

The games of GTI modules are rewarding so; the students get some idea of his/her learning progress. The award system of GTI modules can motivate the students towards learning. There is a steady virtual instructor in the GTI modules who support the learners to navigate through the GTI modules and provide help in the understanding of definition and example of the binary tree and linked list. The virtual instructor aimed to support the learning of various concepts of binary trees and linked list. The students can ask for help anytime from the virtual instructor. We have included a high-level view of gameplay by adding a pseudo-code on the screen. Students can visualize the programming concept while they use the GTI modules (Figure 4).

4.2 GUI for a Linked List and Binary Tree Modules

The GUI for gameplay in the GTI module to teach linked list is given in Figure 5. We have utilized the following GUI to make our modules more appealing:

• Buttons: We have used a range of buttons to make the gameplay more revealing, such that (1) GO button, (2) Help button, (3) Next button and (4) End button. We have used the

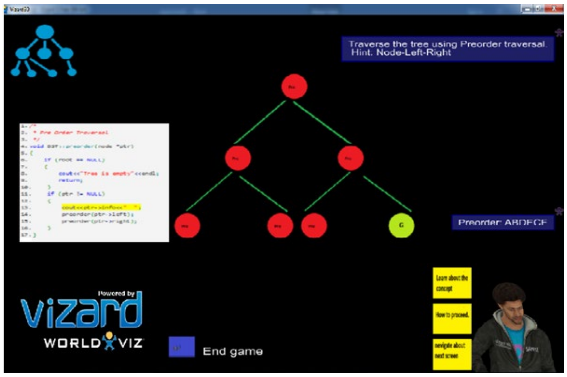


Figure 4: In-order traversal of binary tree

"GO button" for choosing between given options. The "help" button is used for clarifying the queries of the users. We have added the "help" button to make the GTI modules more usable. We have exploited the "Next" button to go on the next screen and; the "End" button to end the gameplay and move onto the main menu page/screen.

• Boxes: We have incorporated boxes (VR logo) to shows the linked list/binary tree module is virtual reality-based gameplay.

• 3D-models: We have exploited 3D-models to make the linked list and binary tree modules more appealing. We have created the 3D models using 3D max and SketchUp and imported these models in Vizard. We have also used in-built avatars of vizard to make the gameplay more likable.

• Text: We have incorporated various "text boxes" to make the GTI modules more informative. The "text boxes" make the GTI modules more usable as the user can take help from the "text." The users also receive a compliment using the text boxes such as: "good job" and "congratulation."

• Scoreboard: There is a scoreboard incorporated in the gameplay to display the score of the users. When the users complete an activity or challenge, then the score is incremented by "50" points, and the scoreboard displays the score of the candidate.

• Images of pseudo-code: The pseudocode of the linked list is displayed on the screen. The goal is to provide the learners with a high-level view of the fundamentals of the linked list. The GTI modules should motivate and engage the students in learning linked list concepts.

5 Evaluation and Results

Our primary goal of this research is to provide a better learning environment by increasing the motivation of the students toward learning. 57 college students (42 undergraduate and 15 graduate students) helped us in evaluating the GTI modules in the spring semester. We demonstrated the technique to use the GTI modules to the students before the evaluation process. Then the students played with the module and completed a post-survey. The survey questions were based on the SMQII questionnaire. The survey contained five questions each for each of the five components of SMQII.

5.1 Statistical Analysis of Motivation Using T-Test

For the statistical analysis of motivation towards learning (the concepts of the binary tree and linked list) with the GTI modules, we have integrated the T-test. We have collected the data using pre- and post-survey from introductory programming classes. We compare the data collected from pre- and post-survey using the t-test. We have used a five-point Likert scale ranging from "strongly agree" to "strongly disagree". Table 1 shows the mean value of motivation of students (for both the pre- and post-survey in the learning of binary tree and linked list concepts. We have analyzed the collected data (Table 2) using t-test with the hypothesis.

H1: There is a significant difference between the motivation of students when they use the GTI module to learn the binary tree and linked list.

We have tested our hypothesis is true using the null hypothesis H0.

Figure 5: Assorted interface elements of GTI modules to teach linked list

Table 1: Mean value of motivation to learn in pre-and post-survey

Motivation	Pre-survey	Post-survey
The Linked list and binary tree concepts I learn are relevant to my life.	2.65	3.40
Learning Linked list and binary tree concept is interesting.	2.61	3.59
I am curious about the applications of the Linked list and binary tree concept.	2.39	3.55
I enjoy learning Linked list and binary tree concept.	2.58	3.51
I am confident I will do well on the Linked list & binary tree concept tests.	2.52	3.63
I am confident I will do well on labs and projects based on the Linked list and binary tree concept.	2.42	3.56
I believe I can master knowledge and skills based on the Linked list and binary tree concept.	2.19	3.69
I am sure I can understand the Linked list and binary tree concept.	2.10	3.63
I use strategies to learn Linked list and binary tree concept well.	2.45	3.46
I spend a lot of time learning Linked list and binary tree concept.	2.58	3.41
I prepare well for Linked list and binary tree concept tests and labs.	2.55	3.41
I study hard to learn the Linked list and binary tree concept.	2.35	3.44
Getting a good grade is important to me.	1.43	4.28
I think about the grade I will get.	1.57	4.22
Scoring high on tests and labs matters to me.	1.37	4.31
Knowing the concepts in depth will give me a career advantage.	1.73	3.85
I will use problem-solving skills in my career.	1.33	3.85

H0: There is no significant difference between the motivation of students in pre-and post-survey.

The result of the above t-test for motivation to learn binary tree and the linked list is summarized in Table 2. The total number of items that we need to compare using the T-test was 17. Table 2 illustrates the mean values of the motivation to learn the binary tree and linked list for each of the groups (pre-and post-survey). The mean value of motivation to learn in post-survey was 3.69 whereas the mean value of motivation to learn in pre-survey was 2.16.

Additionally, it includes the respective results of the variances of both pre- and post-survey based on motivation to learn the linked list and binary tree. The result involves the degree of freedom, the T values and the critical value of the t-tests. Results show that the degree of freedom (df) for the t-test is 27, t-value (t Stat) is 11.0349 and the critical value (t Critical two-tail) is 2.051831. The results of the t-test for the motivation have shown that the t-value (11.0349) is greater than the critical value (2.051831). Additionally, the p-value (1.65E-11) of the t-test is lesser than the hypothesized p-value (0.05). So, we can reject the null hypothesis and we accept the alternate hypothesis such that there is a significant difference in motivation to learn when we use the GTI module for learning binary trees and linked lists.

5.2 Statistical Analysis of Motivation Using ANOVA Test

For the quantitative analysis of the data collected from the user study, we divided the student's data into three categories (expert, intermediate and novice user) based on the familiarity of the students with the software games. We randomly took 16 students from each category of students for ANOVA analysis. We did an ANOVA analysis of the data for the three categories of students. We have done the ANOVA test for all the five types of motivation individually.

Table 2: t-Test: Two-sample assuming unequal variances

	post-survey	pre-survey
Mean	3.692804	2.165908
Variance	0.094652	0.230832
Observations	17	17
Hypothesized Mean Difference	0	
Df	27	
t Stat	11.03492	
P(T<=t) one-tail	8.24E-12	
t Critical one-tail	1.703288	
P(T<=t) two-tail	1.65E-11	
t Critical two-tail	2.051831	

We did the ANOVA test for the "Intrinsic Motivation" of the students after interaction with GTI modules based on the hypothesis (H2).

- H2: the students who had more exposure to other virtual reality games are more intrinsically motivated after interacting with GTI modules.

The results of the ANOVA test show that the difference was statistically not significant ($p = 0.08>0.05$). So, the results do not support our hypothesis, which greater familiarity with software games results in more intrinsic motivation. The graph (Figure 6) shows that the mean value for all three categories of students range from 60.3% to 78.75%. The error bar on bar graphs shows the intrinsic motivation with a 95% confidence level. So, most of the students (mean value 66.88%) got intrinsically motivated toward learning when they use GTI modules for learning.

We performed the ANOVA test for the "self-efficacy" of the students after interaction with GTI modules based on the hypothesis.

- H3: the students who had more exposure to other virtual reality games have more self-efficacy than the students having less exposure to VR games, after interacting with GTI modules.

The results of the ANOVA test show that the difference was statistically significant ($p = 0.02<0.05$). So, the results support our hypothesis, in which greater familiarity with software games results in more self-efficacy.

The graph (Figure. 7) shows that the mean value for all three categories of students' ranges from 62.19% to 84.69%. The error bar on bar graphs shows the mean values self-efficacy with a 95% confidence level. So, most of the students (mean value 70.64%) have improved self-efficacy toward learning when they use GTI modules for learning

We did the ANOVA test for the "self-determination" of the students after interaction with GTI modules based on the following hypothesis.

- H4: the students had more exposure to other virtual reality games have more self-determination after interacting with GTI modules.

The results of the ANOVA test show that the difference was statistically not significant ($p = 0.07>0.05$). So, the results do not support our hypothesis, which greater familiarity with software games results in more self-determination.

The graph (Figure. 8) shows that the mean value for all three categories of students range from 60% to 79.6%. So, most of the students (mean value 67.29%) got self-determination toward learning when they used GTI modules for learning.

We did the ANOVA test for the "Grade Motivation" of the students after interaction with GTI modules based on the hypothesis.

- H5: the students had more exposure to other virtual reality games have more "Grade motivation" after interacting with GTI modules.

The results of the ANOVA test show that the difference was statistically not significant ($p = 0.7>0.05$). So, the results do not support our hypothesis, which greater familiarity with software games results in more grade motivation.

The graph (Figure. 9) shows that the mean value for all three categories of students range from 81.25% to 86.67%. The error

Table 3: ANOVA analysis of Intrinsic motivation

Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Novice	16	48.25	3.015625	1.66224		
Intermediate	16	49.25	3.078125	2.039323		
Expert	16	63	3.9375	1.229167		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	8.492188	2	4.246094	2.583448	0.086675	3.204317
Within Groups	73.96094	45	1.643576			
Total	82.45313	47				

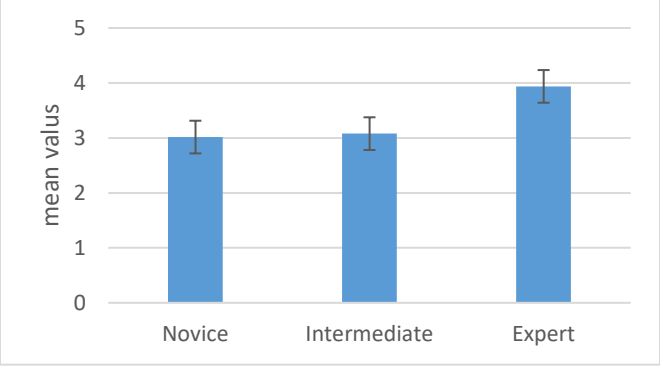


Figure 6: Mean value of Intrinsic motivation for the three categories

Table 4: ANOVA analysis of self-efficacy

Anova: Single Factor				
SUMMARY				
Groups	*Count*	*Sum*	*Average*	*Variance*
Novice	16	49.75	3.109375	1.691406
Intermediate	16	52	3.25	1.525
Expert	16	67.75	4.234375	1.26224

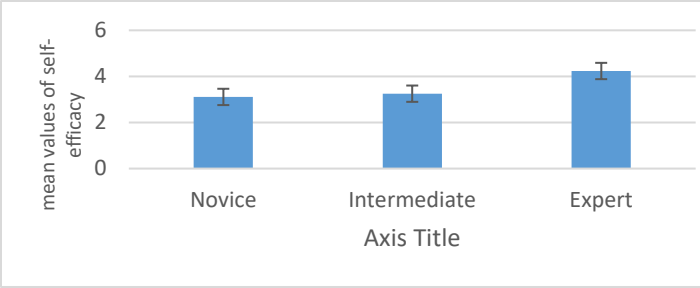


Figure 7: Mean value of self-efficacy for the three categories

bar on bar graphs shows the grade motivation with a 95% confidence level. So, most of the students (mean value 84.3%) got grade motivation toward learning when they used GTI modules for learning.

We did the ANOVA test for "Career Motivation" of the students after interaction with GTI modules based on the following hypothesis.

- H6: the students who had more exposure to other virtual reality games have more Career motivation after interacting with GTI modules.

The results of the ANOVA test show that the difference was statistically not significant ($p = 0.9 > 0.05$). So, the results do not support our hypothesis, which greater familiarity with software games results in more Career motivation.

The graph (Figure. 10) shows that the mean value for all three categories of students ranges from 73.75% to 77.5%. The error bar on bar graphs shows the career motivation with a 95% confidence level. So, most of the students (mean value 75.63%) got Career motivation toward learning when they used GTI modules for learning.

6 Conclusions

We have designed and implemented GTI modules to motivate the students towards learning. We have used the four-dimensional framework with minor extension for the design of GTI modules. The GTI modules can be included in the regular classroom as a supplement of teaching. We have evaluated GTI modules using the SMQII questionnaire. The GTI modules were evaluated in the spring semester with 57 students (15 graduate and 42 undergraduate). We have done statistical analysis using ANOVA test and T-test. The ANOVA test for the four types ("Intrinsic Motivation"," self-determination", "Grade motivation", "Career motivation") of motivation, showed that the difference between the three groups (novice, intermediate and expert player) was not statistically significant. So, the results, conclude that greater familiarity doesn't affect the motivation of the students towards learning.

The outcomes of the statistical analysis (t-test) for the motivation have shown that the p-value (1.65E-11) of the t-

Table 5: ANOVA analysis for self-determination

Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Novice	15	45	3	1.508929		
Intermediate	16	49.75	3.109375	1.341406		
Expert	15	59.75	3.983333	1.932738		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	8.777095	2	4.388547	2.762742	0.074339	3.21448
Within Groups	68.30443	43	1.588475			
Total	77.08152	45				

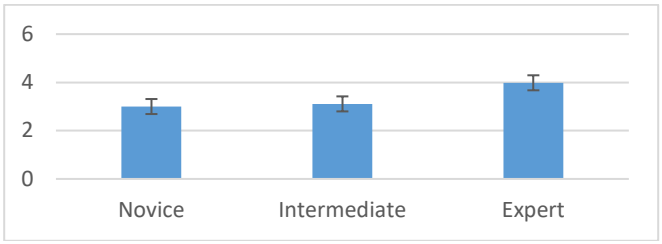


Figure 8: Mean value of self-determination for the three categories.

test is lesser than the hypothesized p-value (0.05). So, we can conclude that students felt motivated and engaged in learning. The result of the empirical evaluation shows that 74% of students (refer Figure 11) got motivated when they used GTI modules for learning.

Acknowledgments

This work is funded in part by the ARL Award: W911NF1820224 and NSF award #1923986.

Table 6: ANOVA analysis for Grade motivation

Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Novice	16	68	4.25	1.207407		
Intermediate	16	65	4.0625	1.277315		
Expert	16	69.33333	4.333333	1.185185		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.615741	2	0.30787	0.251672	0.778587	3.204317
Within Groups	55.04861	45	1.223302			
Total	55.66435	47				

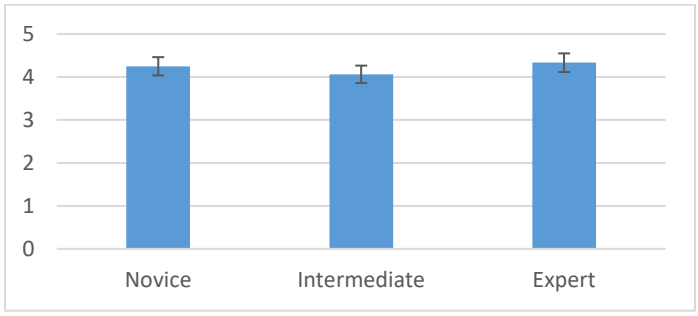


Figure 9: Mean value of grade motivation for the three categories.

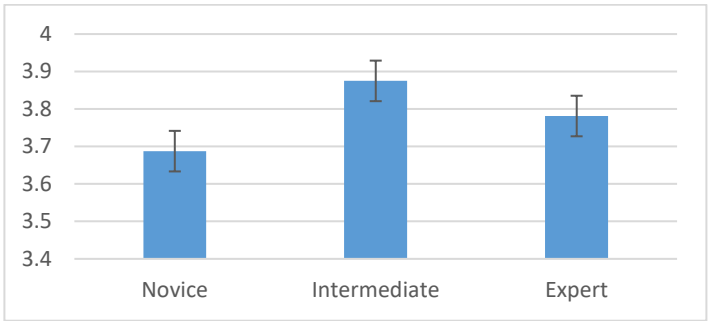


Figure 10: Mean value of Career motivation for the three categories.

Table 7: ANOVA analysis for career motivation

Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Novice	16	59	3.6875	2.495833		
Intermediate	16	62	3.875	1.483333		
Expert	16	60.5	3.78125	1.765625		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.28125	2	0.140625	0.073436	0.929307	3.204317
Within Groups	86.17188	45	1.914931			
Total	86.45313	47				

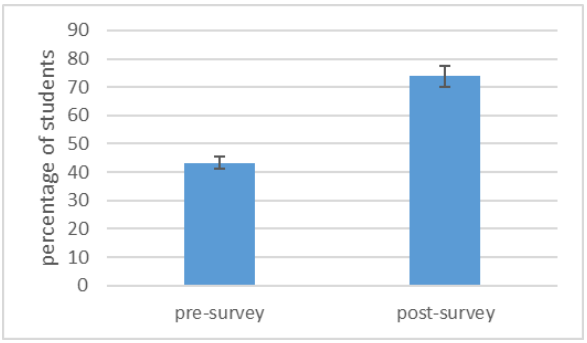


Figure 11: Mean value of motivation for pre- and post-survey

References

[1] Z. Alam, Y. Mehmood, I. Naseem, A. Rustam, and. A. Waqar, "Unethical Practices in University System: Pattern and Causes," Investigation from Pakistan (KPK), *Science Series Data Report*, 4(23), 2012.

[2] J. Biggs and C. Tang, *Teaching for Wuality Learning at University: What the Student Does*, (3rd edition), Open University Press, Buckingham, UK, 2007.

[3] A. Brandlow and, A. Bruckman, *HCI for Kids*, J. Jacko & A. Sears (Eds.), Human-Computer Interaction Handbook, Lawrence Erlbaum, Hillsdale, NJ, pp. 428–440, 2003.

[4] A. Damasio, *Descartes' Error: Emotion, Reason, and the Human, Brain*, Grosset/Putnam, New York, 1994.

[5] A. De Vincente and H. Pain, "Informing the Detection of the Students' Motivational State: An Empirical Study," S. A. Cerri, G. Gouarderes, and F. Paraguacu (Eds.), *Intelligent Tutoring Systems 2002*, Springer-Verlag, Heidelberg, Berlin, LNCS, 2363:932-943, 2002.

[6] L. Dömeová1 and A. Jindrova, "Unethical Behavior of the Students of the Czech University of Life Sciences," International Education Studies; 6(11):ISSN 1913-9020-E-ISSN 1913-9039, *Published by Canadian Center of Science and Education*, 2013.

[7] S. Freitas and M. Oliver, "How Can Exploratory Learning with Games and Simulations within the Curriculum be Most Effectively Evaluated?" *Computers & Education* 46:249-264, 2006.

[8] D. Goleman, *Emotional Intelligence*, Bantam Books, New York, 1995.

[9] L. Harasim, Learning Theory and Online Technologies, Routledge**,** 2012.

[10] D. Heersink and B. Moskal, "Measuring High School Students' Attitudes Toward Computing," *Proceedings of the 41 ACM Technical Symposium on Computer Science Education (SIGCSE '10)*, ACM New York, NY, USA, pp: 446-450, 2010.

[11] A. Hoegh and B. Moskal, "Examining Science and Engineering Students' Attitudes Toward Computer Science", *Proceedings of the 39th IEEE International Conference on Frontiers in Education Conference (FIE'09)*, IEEE Press, Piscataway, NJ, USA, pp: 1306-1311, 2009.

[12] J. M. Keller, "Motivational Design of Instruction," C. M. Reigeluth (Ed.), *Instructional-Design Theories and Models: An Overview of their Current Status*, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 386-434, 1983.

[13] K. Kuutti, "Activity Theory as a Potential Framework for Human-Computer Interaction Research," B. A. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*, The MIT Press, Cambridge, MA, pp. 17-44, 1996.

[14] Y. Matsubara and Y. Nagamachi, "Motivation System and Human Model for Intelligent Tutoring," C. Frasson, G. Gauthier, and A. Lesgold (Eds.), *Proceedings of the 3rd International Conference on ITSs*, Springer-Verlag, Berlin, pp. 139-147, 1996.

[15] J. G. Nicholls, "Quality and Equality in Intellectual Development: The Role of Motivation in Education," American Psychologist, 34(11):1071-1084, 1979.

[16] E. Ossuetta and, S. Sharma, "Virtual Reality Instructional Modules in Education Based on Gaming Metaphor," IS&T International Symposium on Electronic Imaging (EI 2017), *The Engineering Reality of Virtual Reality Proceedings Papers*, Hyatt Regency San Francisco Airport, Burlingame, California, pp. 11-18, 29 January- 2 February 2017.

[17] S. Papert, "Mindstorms: Children, Computers, and Powerful Ideas," *Basic Books*, New York, 1980.

[18] S. Price, Y. Rogers, H. Neale, M. Scaife, and D. Stanton, "Using Tangibles to Promote Novel Forms of Playful Learning, Interacting with Computers," Interacting with Computers, 15(2):169-185, 2003.

[18] S. Price, Y. Rogers, H. Neale, M. Scaife, and D. Stanton, "Using Tangibles to Promote Novel Forms of Playful Learning, Interacting with Computers," Interacting with Computers, 15(2):169-185, 2003.

[19] S. Rajeev, A. Sahu, and S. Sharma, "Game Theme Based Instructional Module to teach Binary Trees Data Structure," *Proceedings of ISCA 26th International Conference on Software Engineering and Data Engineering*, San Diego, CA, USA, pp. 13-18, October 2-4, 2017.

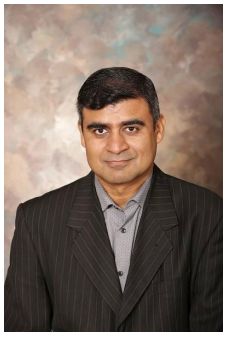
[20] S. Rajeev, S. Sharma, and, J. Stigall, "Game–Theme Based Instructional Module for Teaching Object-Oriented Programming," *Proceedings of the IEEE International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, USA, DOI 10.1109/CSCI.2015.3, pp. 252-257, December 7-9, 2015.

[21] S. Sharma and J. Stigall, "Virtual Reality Instructional Modules for Introductory Programming Courses," *Proceedings of IEEE Integrated STEM Education Conference (ISEC)*, DOI:978-1-5090-5379-7/17, Princeton, New Jersey, Saturday, pp.: 33-41, March 11, 2017.

[22] S. Sharma and, J. Stigall, "Usability and Learning Effectiveness of Game-Themed Instructional (GTI) Module for Teaching Stacks and Queues," *Proceedings of IEEE SoutheastCon 2018*, Hilton Bay Front, St.

Petersburg, FL, USA, pp. 1-6**,** April 19 - 22, 2018.
[23] T. Soldato, "Detecting and Reacting to the Learner's Motivational State," C. Frasson, G. Gauthier & G. I. McCalla (Eds.), *Intelligent Tutoring Systems: Proceedings of ITS'92*, Montréal, Canada, Springer-Verlag, New York, pp. 567-574, 1992.
[24] K. Squire, "Cultural Framing of Computer/Video Games," Game Studies, 2(1), See http://www.game studies.org/ 0102/squire/, Last accessed 15th March 2005.
[25] S. Woods, *Loading the Dice: The Challenge of Serious Video Games, Game Studies*, 4(1)http://www.game studies. org/0401/woods/, Last accessed 15th March 2005.

Sarika Rajeev (photo not available) is an Adjunct Professorial Lecturer in the Department of Computer Science at American University. She also worked as an Adjunct faculty in the Computational and Data Science department at George Mason University. She has received D.Sc. in Computer Science Bowie State University in 2018. She has published many papers under Dr. Sharma, Director of the Virtual Reality Laboratory, at College of Arts and Science at Bowie State University. The Virtual Reality Laboratory applies virtual reality and augmented reality as a tool for learning, training, and education. Sarika's research focus is on instructional modules, software development, SDLC and Data Science. Her work is motivated by the need of research in game theme based instructional modules to teach introductory programming.



Sharad Sharma is a Professor in the Department of Computer Science at Bowie State University. He has received a Ph.D. in Computer Engineering from Wayne State University, Detroit, MI in 2006 and an M.S. from University of Michigan, Ann Arbor, MI in 2003. He has won the "Outstanding Researcher Award" in 2011 and 2013, "Outstanding Faculty Award" in year 2012, "Outstanding Publication Award" in year 2010, and "Outstanding Young Faculty Award" in year 2009 at College of Arts and Science at Bowie State University. Dr. Sharma is the Director of the Virtual Reality Laboratory at Bowie State University. The laboratory applies virtual reality and augmented reality as a tool for learning, training, and education. Dr. Sharma's research focus is on Modeling and Simulation of Emergency Evacuation, Human Behavior Modeling, Agent-Based Modeling, Multi-Agent System, Gaming, Fuzzy Logic, Data Science and Data Visualization.

A Hardware and Software Prototype of the CTAR All-Star

Terri Heglar* Andrew Penrose† Austin Yount‡
Kristine Galek§ Yantao Shen¶ Sergiu M. Dascalu‖ Frederick C. Harris, Jr.**
University of Nevada Reno, NV 89557, USA

Abstract

The CTAR All-Star is a system consisting of a rubber ball, a pressure sensor, and a bluetooth transmitter paired with a cross-platform mobile application. The device is used as a rehabilitation tool for people with dysphagia in a similar fashion to the traditional chin tuck against resistance (CTAR) exercise by squeezing a ball between the chin and upper chest. The mobile device monitors and displays the pressure inside the ball on a real-time graph allowing the patient to follow exercise routines set by Speech-Language Pathologists. Additionally, the application stores exercise data that can be used to both monitor the patient's progress over time and provide objective data for future research purposes.

keywords: chin tuck against resistance, dysphagia, rehabilitation.

1 Introduction

Dysphagia is a medical symptom with a neurological underpinning that is characterized by difficulty swallowing. Primarily seen in stroke victims and the elderly, this condition is estimated to affect roughly 9 million adults of which approximately 37 percent are diagnosed with dysphagia in the US [3]. Therefore, it can be estimated that the number of people undergoing medical treatment for dysphagia is around 3.3 million people each year. For the purpose of this paper the focus will be on dysphagia, which is caused by upper esophageal sphincter dysfunction. Dysphagia is caused by sphincter (UES) dysfunction and lack of bolus flow into the esophagus and can result in serious health consequences including malnutrition, dehydration, aspiration pneumonia, and death [8].

One common exercise for strengthening the suprahyoid muscles and improving oral feeding is the chin tuck against resistance (CTAR) which has been reported to be an effective treatment for dysphagia. [11] This exercise involves squeezing a ball between the chest and chin in order to strengthen the muscles in the neck, and improve bolus flow into the esophagus. Furthermore, it is known that patients perform better and work harder with visual feedback. The CTAR All-Star aims to improve on the traditional rehabilitation experience for patients and Speech-Language Pathologists (SLPs). The system capitalizes on the increasing prevalence of mobile smartphones and tablets in modern society to create a package that is both affordable and feature-rich. The CTAR All-Star is composed of a cross-platform mobile application and a wireless exercise device that can be used by both parties for creating, assigning, and performing exercise routines as well as monitoring the results of rehabilitation sessions. Using this system, the SLP can create personalized exercise routines and assign the exercises as home practice to the patients. During the home practice exercise routines, the application guides the patient through the exercise step by step while plotting progress, pressure, and pressure thresholds inside the ball in real time.

The rest of this paper is structured as follows: Section 2 discusses the software design of the mobile application, Section 3 goes over the hardware design of the exercise device, Section 4 walks through the implementation of the CTAR All-Star, and Section 5 reviews the conclusions and future work associated with the CTAR All-Star.

2 Software Design

This work began with a brainstorming meeting between several of the co-authors of this paper. During this meeting we wanted to leverage our previous work on tool development for doctors in the field of Speech Pathology and Audiology. Our previous work had led to an open source game for dysphagia therapy [7]. We came up with the idea of constructing the hardware and software for the device described in this paper.

The requirements detailed in the following subsections include descriptions of the functional and nonfunctional requirements used by the system. The construction of these requirements was developed through interviews with stakeholders that had a hand in the project. Besides the requirements we also present use cases, the traceability matrix, and the database design of the application. The requirements and use cases for the application were realized through interviews with multiple stakeholders including SLPs.

2.1 Stakeholder-Client Interviews

Once the development team was put together, the first process was constructing the requirements. This was done by interviewing several of our clients. These were interviewed in person and via email. In order to collect consistent information

*theglar@nevada.unr.edu
†apenrose@nevada.unr.edu
‡austinyount11@gmail.com
§kgalek@med.unr.edu
¶ytshen@unr.edu
‖dascalus@cse.unr.edu
**fred.harris@cse.unr.edu

we put together a set of questions to ask. The first ten questions we asked of all clients are as follows:

1. How would your profession benefit from this project?
2. What in your view would make this a successful project?
3. What functions are desired?
4. What is the priority of each feature or function?
5. What are the business requirements?
6. What results are required/desired?
7. What are the metrics to define success?
8. Are there any other requirements we should be aware of?
9. Are there any products/projects related to this one?
10. Is there anything we didn't discuss?

Additionally we asked Client 1 several more questions:

- What is the current process?
- What type of exercises would this project be used for?
- What kind of people will the primary users be?
- What are the pitfalls/what is inhibiting success with the current process?
- What's been done to solve this already?
- How much history would you like stored & who should have access to it?
- What level of security is required?
- What kind of training do you anticipate needing?
- Is there anyone we need to speak with who isn't on our list?

Client 1: Our first client spoke about the need for an effective way to measure the results of the chin tuck against resistance exercise. Currently, they have no way of knowing if the patient is actually putting any amount of force into the exercise. With proper measuring techniques, they can gather better data for research and determine if the exercise is indeed an effective rehabilitation technique.

She said her profession would benefit in several ways from this project. It will help speech pathologists know more about swallowing conditions and rehabilitating them. Additionally, it will help a speech pathologist give more specialized treatment to patients. She also said several times that patients perform better with visual feedback.

The business requirements she spoke about are a product that works and works reliably. Data gathered from the device should be repeatable and precise. She would also like the price point to be low enough that a patient can pay cash for it, no more than $200 dollars, making it affordable and accessible. This price would not include the phone or tablet required to run the application.

The functions she would like the project to encompass are a visualization of the exercises, a way to set thresholds, a hold time with a target, and a repetition counter. She would like separate training and exercise modules, with different modules for isometric and isotonic exercises. She would also like the visualization neutral enough to appeal to multiple demographics, although a large part of our users will be older and many of them under long term care.

As for security, in order to meet HIPAA requirements we can't use any identifying information. She said we can use their medical records number for our application, and can upload a pdf or histogram to their electronic medical record (EMR). Currently, there are no competitors in the CTAR exercise devices. However, there are several for similar rehabilitation exercises. These exercises include sEMG, surface electromyography, with the leading competitor being Synchrony by ACP (Accelerated Care Plus), and an exercise that strengthens the tongue using a bulb, which is led by IOPI Medical. Both of these companies have effective ways to measure rehabilitation results and somewhat interactive visualizations.

Client 2: Our second client was a graduate research assistant at the University of Nevada, Reno. She works as a clinician for the Northern Nevada Voice and Swallow Clinic. She treats patients with dysphagia and will be a user of our product when it is available, which makes her a good person to interview. We asked her the same first ten questions we asked Client 1. We interviewed her via email and the following is a summary of her answers.

Like Client 1, she expressed the need for a measurable system, to broaden their understanding and research, which will lead to improved patient outcomes. She said they currently use a rolled-up towel or an inflated ball under the chin. They have no way to quantify the resistance. They also cannot change that resistance. Her measurement of our success is if we are able to create something that she will be able to use in practice. Integrating a way to change the resistance would be helpful as well. As for business requirements, she suggests that it has to be portable and cost effective, and depending on our target audience, the cost can vary.

The primary function she would like to see is a way to measure how much pressure the patient is exerting. In addition to measurement, she is interested in knowing what level of pressure a healthy person is capable of exerting as a baseline. Some additional functionality requirements are portability, accuracy, data storage, visual representation, and sterilizability or disposable parts to prevent spreading of infections. She also commented that we need to consider the constant changes of the medical field. We should be able to easily update our software, possibly remotely, when the products are out in the field.

Client 3: Our third client is a Lecturer in Speech Pathology and Audiology at the University of Nevada, Reno. She is a clinician at the Northern Nevada Voice and Swallow Clinic,

which makes her an ideal candidate to interview. We gathered her answers via email and asked her questions 1-10.

She expressed that her profession would benefit from this device because it would allow them to integrate additional objective data into their rehabilitation program. She would like the device to measure pressure and provide visual feedback as well as track duration. She would consider us successful if we can achieve these functionalities.

For business requirements, she said the most important thing to consider is cost. The product should be inexpensive so that each patient can purchase one. If we aren't able to keep the cost down, we should incorporate materials that are tolerant to a high level of disinfecting or integrate the use of disposable covers. She mentioned that there is research and data that support the use of the CTAR ball, and there is an ideal size for the ball that we should look into. She is unaware of any competitors that quantify the pressure component.

2.2 Functional Requirements

Functional Requirements are "statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do"[9]. The following functional requirements describe the behavior of the CTAR All-Star. There are several more than the ones listed, but these are the most critical and give a good sense of the design.

1. The CTAR All-Star stores a user profile.
2. The CTAR All-Star identifies users only by EMR numbers.
3. The CTAR All-Star forces the user to log in for the first time.
4. The CTAR All-Star connects via Bluetooth 4.0 using an HM-10
5. The CTAR All-Star measures the gauge pressure between the ambient air and the inside of the ball.
6. The CTAR All-Star broadcasts pressure readings at least once every 100ms.
7. The CTAR All-Star allows the user to drop/change the current bluetooth connection.
8. The CTAR All-Star allows the user to start a new session.
9. The CTAR All-Star allows the user to stop the current session.
10. The CTAR All-Star offers Isometric and Isotonic exercise routines.
11. The CTAR All-Star provides a visual representation of the pressure during an exercise.
12. The CTAR All-Star stores the 1-Rep Max value used for calculating a threshold.
13. The CTAR All-Star stores a repetition count value.
14. The CTAR All-Star allows the user to view data from previous sessions.
15. The CTAR All-Star provides a countdown timer.
16. The CTAR All-Star timer pauses and restarts if the pressure falls below the threshold.
17. The CTAR All-Star allows the user to view aggregate data.

2.3 Non-functional Requirements

Non-functional requirements are "constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole rather than individual system features or services"[9]. The following list contains the Non-Functional requirements for the CTAR All-Star application.

1. The CTAR All-Star runs on Android and iOS platforms.
2. The CTAR All-Star communicates between the device and ball via Bluetooth 4.0.
3. The CTAR All-Star is implemented using the Xamarin framework.
4. The CTAR All-Star is written in C, C#, XAML
5. The CTAR All-Star provides Visual feedback in real time.
6. The CTAR All-Star has a user friendly interface.
7. The CTAR All-Star protects the security of each user.
8. The CTAR All-Star visualization is neutral to appeals to multiple demographics.

2.4 Use Cases

Use cases describe on a high-level how the users should interact with an application. Figure 1 shows the interaction between the CTAR All-Star and the patient or doctor. The major use cases for this application are as follows:

1. **CreateUser:** Patients and medical professionals should be able to make a patient user account from a mobile device using the patient's EMR number. Medical professionals should be able to make an account which allows them to view their patient's information.
2. **LogIn:** Patients should be able to log in to the mobile app the first time the app opens. The home screen is displayed after successful login.
3. **PairDevice:** Users should be able to connect their mobile device to the exercise ball by selecting from a list of available Bluetooth devices. The app will prompt this when a device is not connected or allow the user to change the connection through settings.
4. **CreateExercise:** Medical professionals should be able to make a patient exercise plan that the patient can work on between appointments. Plans would include specifying the

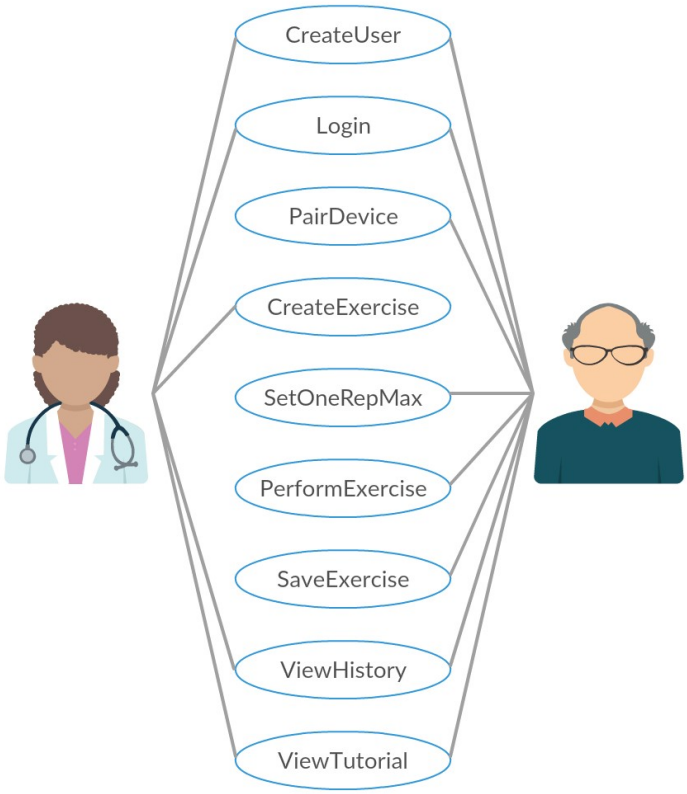


Figure 1: The Use Case Diagram shows the patient and doctor interactions with the CTAR All-Star.

exercise module and the number of times or frequency that the module should be completed during a specified time period.

5. **Set1_RepMax:** The app guides the patient through setting a 1-rep max at the beginning of each exercise session. The app prompts the user to squeeze the ball as hard as they can and stores the maximum pressure inside the ball.
6. **PerformExercise:** Patients should be able to follow an exercise routine. The program sets various pressure thresholds based on the patients 1 rep max. The program then guides the patient through the exercise routine where the patient squeezes and releases the ball while a counter counts the repetitions for a certain threshold.
7. **SaveExercise:** The patient's exercise should be saved at the end of a session so that the results can be viewed at a later date.
8. **ViewHistory:** Patients and medical professionals should be able to view historical exercise data.
9. **ViewTutorial:** The patient or medical professional should be able to view various tutorials with screenshots instructing them on how to use the various features of the app.

Each Use Case had a detailed template that described several items. An example of this detailed use case for the Create Exercise use case can be seen in Figure 2.

Use Case: CreateExercise	
Use Case ID	UC4
Brief Description	Medical professionals should be able to make a patient exercise plan that the patient can work on between appointments. Plans would include specifying the exercise module and the number of times or frequency that the module should be completed during a specified time period.
Primary Actors	Clinician/Medical Professional
Secondary Actors	Patient
Precondition(s)	1. Clinician/Medical Professional has an account 2. Patient has an account
Main Flow	1. Medical Professional selects one or more modules to be completed 2. Medical Professional selects the frequency or number of times the exercise should be completed 3. Medical Professional saves the exercise plan 4. Medical Professional assigns the exercise plan to selected patients
Postcondition(s)	1. The user is reminded about new or incomplete exercises through push notifications
Alternative Flow	None

Figure 2: Detailed Use Case for Create Exercise

2.5 Traceability Matrix

Traceability policies "define the relationships between each requirement and between the requirements and the system

Requirements	Use Cases							
	1	2	3	4	5	6	7	8
1	X	X						
2	X							X
3	X	X						
4			X					
5					X	X	X	
6						X	X	
7			X					
8					X	X	X	
9					X	X	X	
10				X		X	X	
11					X	X	X	
12					X			
13				X		X	X	
14								X
15						X	X	
16						X	X	
17								X

Figure 3: The Traceability Matrix shows the relationships between each functional requirement and use case.

design that should be recorded.[9]" The CTAR All-Star Traceability Matrix, as seen in Figure 3, is the tool used to track these relationships. Organizing these connections is essential in analyzing proposed changes and the impact they have on other parts of the system.

2.6 Database Design

The main data structures used for the CTAR All-Star are tables in a SQLite[5] local database. Each patient is only identifiable by their Electronic Medical Record (EMR) number to avoid any HIPAA violations. The EMR is used along with their password to login when the patient initially opens the application. Table 1 depicts the class tables, the primary keys, and the class attributes. The User class is role-based with their role being stored in the userType attribute. The UserName is the patient's EMR or another unique identifier for the doctors. An auto generated unique Id serves as the primary key. The DocID identifies the patient's corresponding SLP.

There is a one-to-many relationship between the User table and the Workout table. In the Workout table, there is a WorkoutID, WorkoutName, PatientEmrNumber, DoctorID, and several other attributes that allow the details of the workout to be stored, such as the number of reps, sets, and the duration of each step in the exercise. The WorkoutID specifies and uniquely identifies the workout number. The PatientEmrNumber is used to link each workout to the corresponding patient. The DoctorId corresponds to the proper medical professional. Additionally, there is a measurement table. This table is used to record the measurement data from each exercise session. The measurement table contains the Id, UserName, DocID, SessionNumber, a timestamp with various display formats, Pressure, and OneRepMax. The Id is the primary key which is auto generated and uniquely identifies measurements from the ball. UserName and DocID are linked to the User table in a one-to-many relationship. The pressure field is used to record the pressure reading as an integer. The OneRepMax is used to track the patient's current capabilities at the time the exercise was completed.

The tables are related to display the history inside the application for both the patient and the medical professionals to review. If a medical professional's UserName is linked with a patient's DocID, then they have access to that patient's session history. The Patient class is designed to assist in this linkage, specifically for allowing a doctor to create workouts for a patient before they have created their own account. By linking the tables, the medical professional is able to analyze a patient's data and determine if they are making positive progress. If the patient is not making progress, a workout adjustment may be needed.

The remaining tables, GraphMeasurement, NativeDevice, and Tutorial, are used for various other specific features in the application. The GraphMeasurement stores the data that will be displayed in the graph on the exercise screen. The NativeDevice stores the available bluetooth devices used when connecting to the hardware. Finally, Tutorial contains the individual instruction guides for the Tutorial's page.

3 Hardware Design

A major goal of the hardware was to make it affordable. A cost-effective device will enable patients to purchase their own personal device and perform exercises from within their home. Figure 4 shows the initial prototype and the hardware design can be seen in Figure 5. The patient uses the device by squeezing the pneumatic ball between the chin and upper chest. A pressure transducer outputs an analog voltage to the Arduino before the analog reading is converted to digital and transmitted over Bluetooth 4.0.

There are several articles which were consulted in Bluetooth software design. [2] explores the needs of wireless medical sensors. [4] gives an introductory tutorial on how to configure an arduino to communicate with an iOS device via Bluetooth. It lists the hardware needed, a great deal of information relating to code on the arduino side, details on wiring, and some information on development of the mobile app for iOS.

The hardware components used in the prototype include the following:

- **Pneumatic ball:** This ball is appropriately sized to fit between an adults chin and upper chest. It has a valve for an inflation needle which makes probing the internal pressure simple.
- **Inflation needle and plumbing:** The inflation needle and plumbing allow the pressure sensor to read the pressure inside the ball.
- **Pressure Transducer:** The pressure transducer reports the gauge pressure inside of the ball as an analog voltage value between 0 and 4.5 volts. The pressure transducer used in the prototype has a range of 0 to 7.25 psi.
- **Arduino Uno:** The Arduino Uno[1] reads the analog pressure from the pressure transducer every 50 ms.

Table 1: Data Structures

Class	Key	Attributes
User	Id	Username, Password, userType, isLoggedIn, OneRepMax, Session, DocID
GraphMeasurement	Id	Time, Pressure
Measurement	Id	UserName, DocID, SessionNumber, TimeStamp, DisplayTime, DisplayDate, Pressure, OneRepMax
NativeDevice	Id	Name
Patient	PatientId	PatientEmrNumber, DoctorName
Tutorial	Id	Topic, Description, ImageName, isVisible, URL
Workout	WorkoutId	WorkoutName, PatientEmrNumber, DoctorID, NumReps, NumSets, ThresholdPercentage, HoldDuration, RestDuration, Type

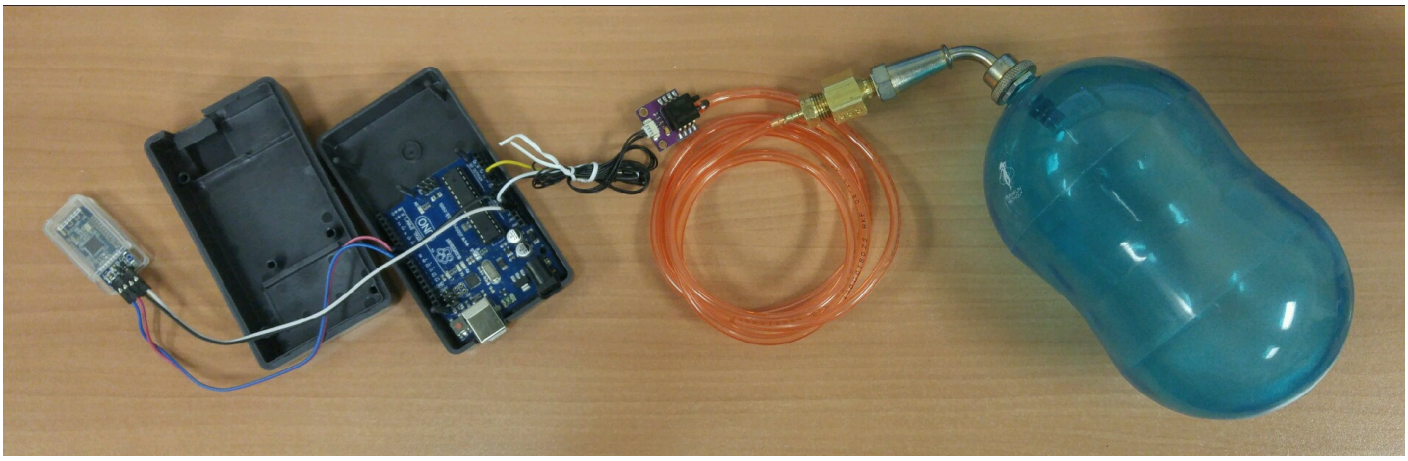


Figure 4: The initial prototype of the CTAR All-Star was powered by a PC through a USB cable.

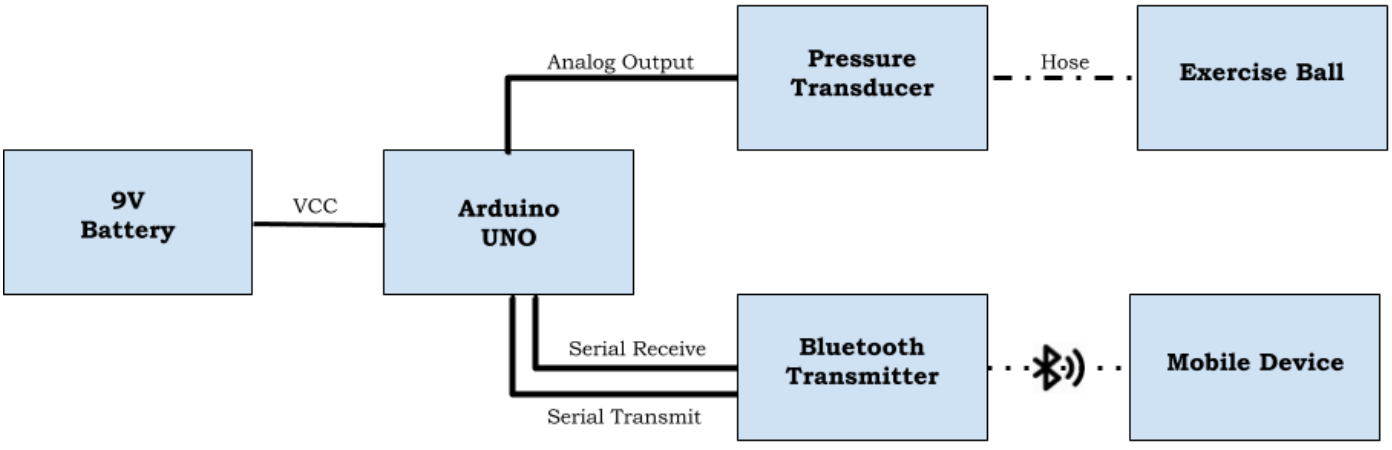


Figure 5: The Hardware Diagram defines how the components connect and interact.

Figure 6: A patient learns how to use the CTAR All-Star.

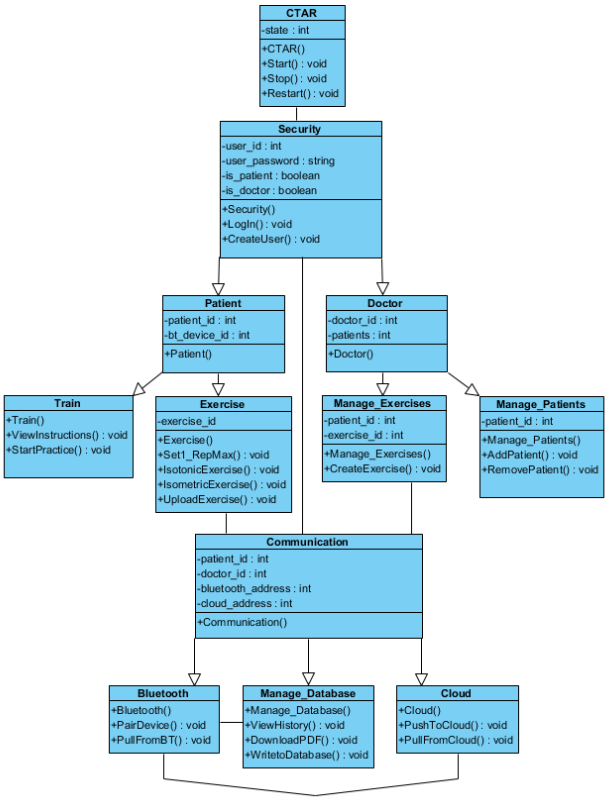


Figure 7: Class Diagram for CTAR All-Star

The analog value is converted to digital and then sent to the Bluetooth transmitter via universal asynchronous receiver/transmitter (UART).

- **Bluetooth Transmitter:** The Bluetooth transmitter receives readings via UART and then transmits them to the mobile device.

The pressure inside the ball is sampled continuously and each reading is transmitted over Bluetooth 4.0 to the mobile app where the pressure is recorded and displayed for the user. Figure 6 shows the device in action.

4 Software Implementation

The implementation of the CTAR All-Star was built around various classes in C#. These classes can be seen in the Class Diagram in Figure 7. The application was designed with a user friendly interface. It allows doctors to create or modify exercise plans (Figure 8) while patients can complete their prescribed exercises with visual feedback (Figure 9). The exercise history is stored for monitoring improvements and will additionally be used for future research purposes (Figure 10). It is a cross platform application, working on both Android and iOS tablets and mobile devices (Figure 11).

The mobile application is built using the Xamarin.Forms [6] framework in the Microsoft Visual Studio IDE. All of the code is written using a combination of C# and XAML, the Plugin.BLE package is utilized for management of Bluetooth 4.0 connections, the Syncfusion[10] plugin is utilized for plotting the ball pressure, and the mobile app is built for use on both iOS and Android devices.

5 Conclusions and Future Work

5.1 Conclusions

The CTAR All-Star allows medical professionals and patients with dysphagia to track CTAR workouts over time. This application is significant because with technology incorporated into the ball, doctors are able to accurately track patient progress. The real time graph is interactive, gives instant feedback, and guides patients through their workouts. This first-of-it's-kind cross-platform application is necessary to gauge the success of both the CTAR exercise and the rehabilitation progress of the patient. With future work such as an online server and embedding the hardware inside of the ball, the CTAR All-Star can become a marketable product.

5.2 Future Work

This software can be modified to be used with different types of devices such as expiratory/inspiratory muscle trainers, IOPI®, or SEMG. Although aimed at medical rehabilitation

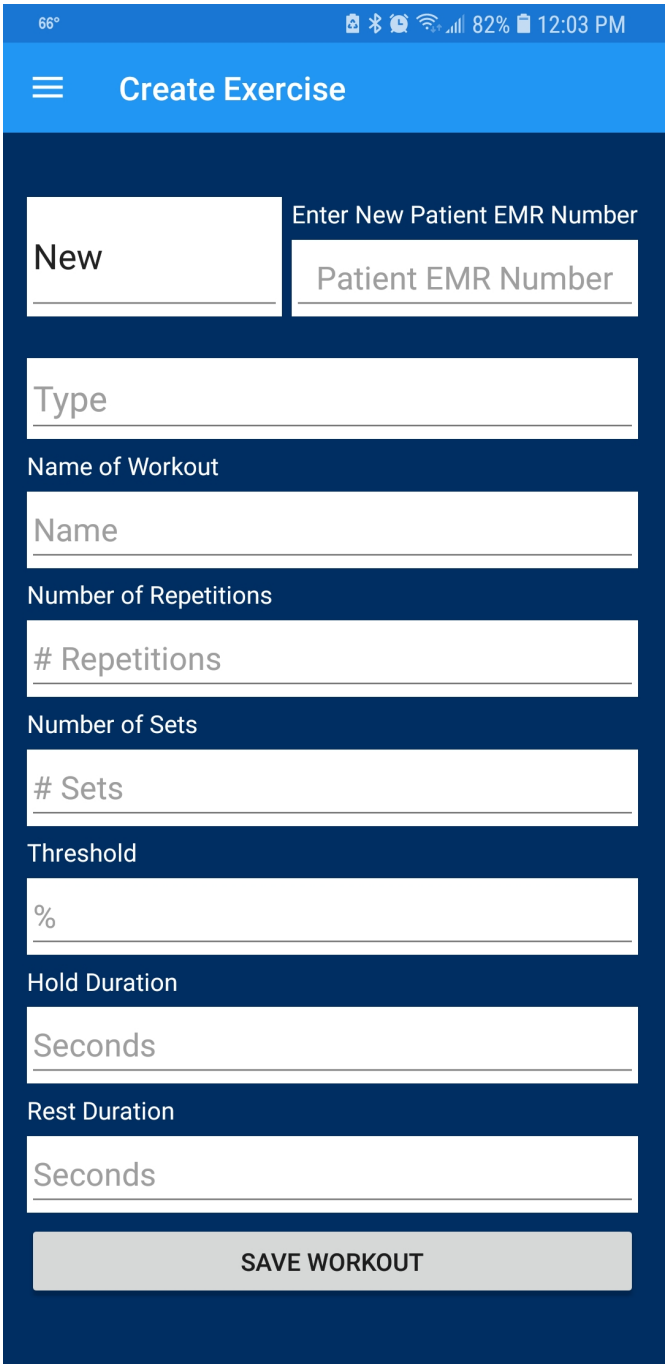


Figure 8: Doctors can create an exercise.

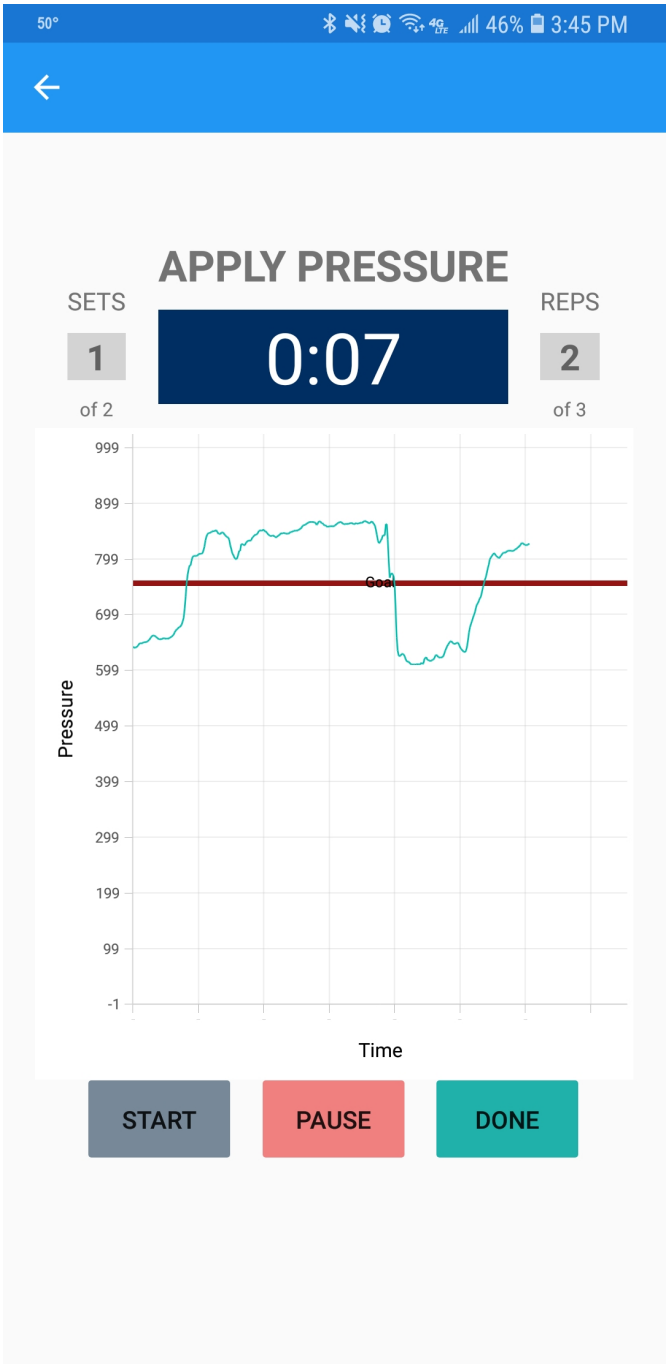


Figure 9: Patients are led through the exercise.

exercises, it may also be useful in other fields. This specific project could also integrate a game to make rehabilitation exercises more fun and interactive. Creating a mobile game, or incorporating a game already created for dysphagia rehabilitation, such as Avaler's Adventure, to work inside the CTAR mobile application would be a great option [7]. Additionally, a server hosted database back end with an API would make communication across multiple devices possible. Finally, refining the design of the hardware so that the electronics can be embedded inside the ball would greatly improve the reliability and ease of use of the device.

6 Acknowledgments

This material is based upon work supported by the National Science Foundation under grant number IIA1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not

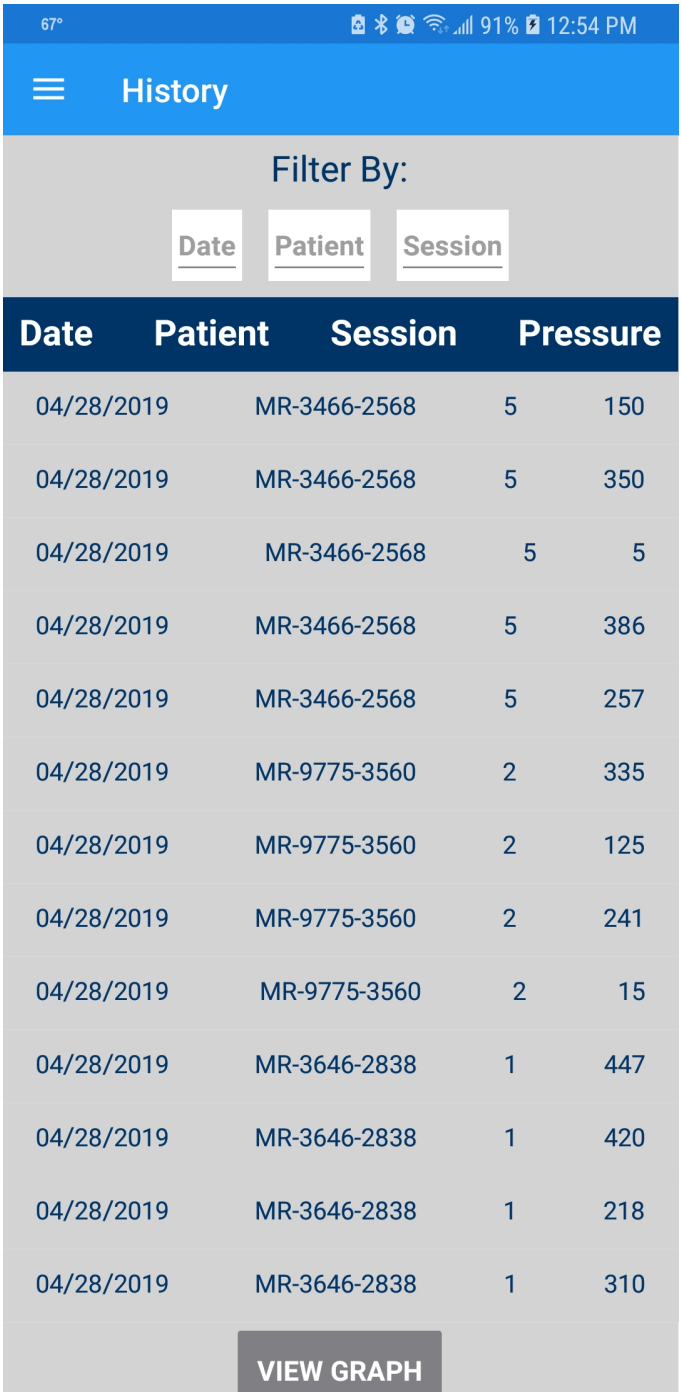


Figure 10: History is stored for future research.

necessarily reflect the views of the National Science Foundation.

References

[1] Arduino. Arduino Language Reference. `https://www.arduino.cc/reference/en/`. (Last Accessed: 2019-06-26).

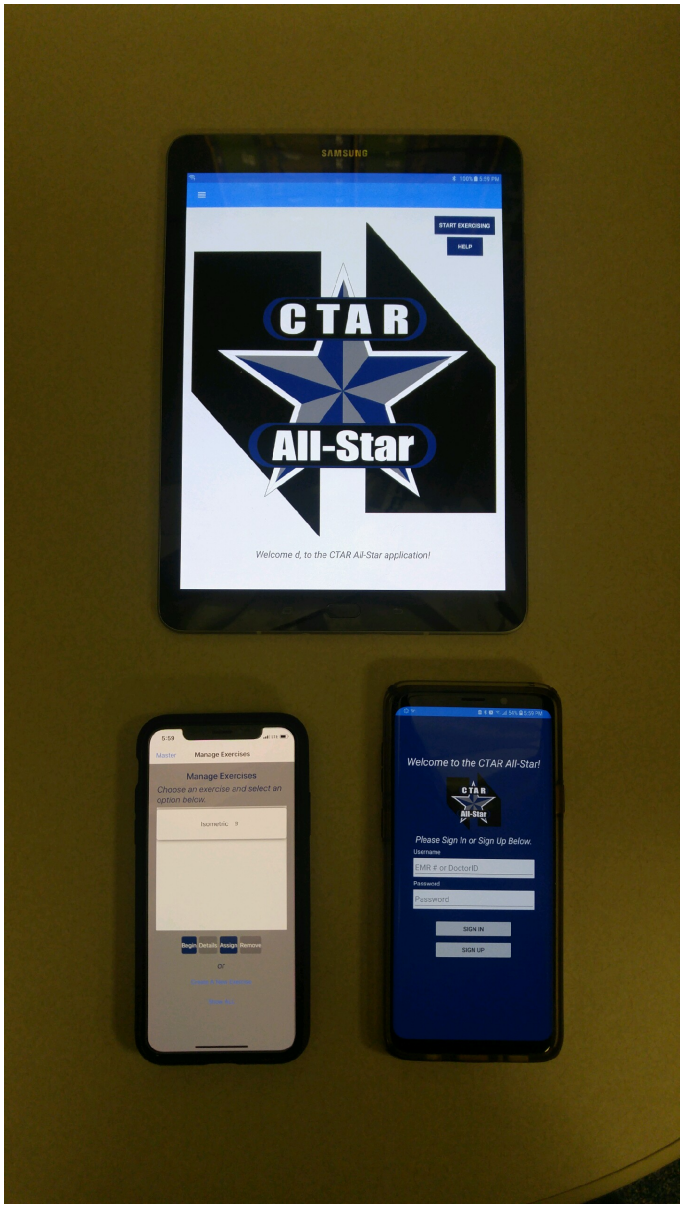


Figure 11: The CTAR All-Star can be run on multiple devices and operating systems including Android and Apple's iOS.

[2] P. Bai, J. Li, Y. Li, and X. Duan. Application of Mobile Bluetooth Based on Human Physiological Parameters Wireless Sensor. In *2016 IEEE International Conference on Consumer Electronics-China (ICCE-China)*. pp. 1-3, Dec 2016, DOI:10.1109/ICCE-China.2016.7849740.

[3] Neil Bhattacharyya. The Prevalence of Dysphagia Among Adults in the United States. *Otolaryngology–Head and Neck Surgery*, 151(5):765–769, 2014. PMID: 25193514, `https://doi.org/10.1177/0194599814549156`.

[4] Owen L Brown. Arduino Tutorial: Integrating Bluetooth LE and iOS. `https://www.raywenderlich.com/`

2295-arduino-tutorial-integrating-bluetooth-le-and-ios. Last accessed: 2020-03-01.

[5] SQLite Consortium. About SQLite. https://www.sqlite.org/about.html. Last accessed: 2019-06-26.

[6] Charles Petzold. *Creating Mobile Apps with Xamarin.Forms*. Microsoft Press, 2016.

[7] Catherine R. Pollock, Daneil A. Lopez, Wambaugh Gunnar, Luis Almanzar, Amanda Morrissey, Kathryn Krings, Kristine Galek, and Frederick C. Harris Jr. Avaler's Adventure: an Open Source Game for Dysphagia Therapy. *Proceedings of the ISCA 26th International Conference on Software Engineering and Data Engineering (SEDE 2017)*, 26:25–30, 2017. https://www.cse.unr.edu/~fredh/papers/conf/180-aaaosgfdt/paper.pdf.

[8] Nicole Rogus-Pulia, Georgia Malandraki, Joanne Robbins, and Sterling Johnson. Understanding Dysphagia in Dementia: The Present and the Future. *Current Physical Medicine and Rehabilitation Reports*, 3:86–97, 01 2015. DOI: 10.1007/s40141-015-0078-1.

[9] Ian Sommerville. *Software Engineering*. Pearson, 2016.

[10] Syncfusion. Syncfusion.com. https://www.syncfusion.com/. Last accessed: 2019-06-26.

[11] Wai Lam Yoon, Jason Kai Peng Khoo, and Susan J. Rickard Liow. Chin tuck against resistance (ctar): New method for enhancing suprahyoid muscle activity using a shaker-type exercise. *Dysphagia*, 29(2):243–248, Apr 2014. https://doi.org/10.1007/s00455-013-9502-9.

Terri Heglar graduated with a BS in Computer Science and Engineering with a Minor in Mathematics from the University of Nevada, Reno in 2019. She currently works as a Software Engineer at Sierra Nevada Corporation. In her free time, she coaches a high school robotics team at the Boys and Girls Club of Truckee Meadows. Her interests include mobile application development, robotics and machine learning.

Andrew Penrose (photo not available) completed his BS in Computer Science and Engineering at the University of Nevada, Reno in 2019. He is currently working as a Software Engineer. His research interests are in tool design and construction.

Austin Yount completed his BS in Computer Science and Engineering at the University of Nevada, Reno in 2019. He is currently working as a Software Engineer on the design and implementation of epithelial barrier research equipment. His research interests are in software engineering, tool design, and tool construction.

Kristine Galek received her bachelor's degree from the University of Arkansas. She earned her master's degree from the University of Arkansas for Medical Sciences and her doctoral degree from the University of Nevada, Reno School of Medicine. She holds the academic rank of assistant professor. She is also the Co-Director of the Northern Nevada Voice and Swallow Clinic and Director of the RAVSS Research Laboratory. Dr. Galek teaches undergraduate, and graduate courses and supervises clinical practicum in the areas of swallowing disorders, craniofacial disorders, resonance, endoscopy, pediatric feeding disorders, manometry, and voice disorders. Her research interests include swallowing disorders, voice disorders, and craniofacial disorders. She has published her research in scholarly journals and given oral presentations at the American Speech and Hearing Association annual convention, and the Florida Dysphagia Institute. Dr. Galek is a clinician who sees patients and works with students individually to develop clinical skills.

Yantao Shen received his Ph.D. degree in Mechanical and Automation Engineering from The Chinese University of Hong Kong in 2002. He is currently an associate professor with the Department of Electrical and Biomedical Engineering at the University of Nevada, Reno. Dr. Shen's research is in the areas of Bio-instrumentation & Automation, Biomechatronics/robotics, Sensors and Actuators, and Tactile/Haptic Interfaces. His research has been supported by federal agencies such as the National Science Foundation (NSF), National Institute of Health (NIH), as well as the state's and local agencies. Dr. Shen has published more than 120 research papers in the fields. Several publications have been nominated for or have won the Best Paper Awards, including in the IEEE international conferences ICRA, IROS, ROBIO, AIM and RO-MAN. In addition, Dr. Shen is a recipient of NSF CAREER Award.

Sergiu Dascalu is a Professor in the Department of Computer Science and Engineering at the University of Nevada, Reno, USA, which he joined in 2002. In 1982 he received a Master's degree in Automatic Control and Computers from the Polytechnic University of Bucharest, Romania and in 2001 a Ph.D. in Computer Science from Dalhousie University, Halifax, NS, Canada. His main research interests are in the areas of software engineering and human–computer interaction. He has published over 180 peer-reviewed papers and has been involved in numerous projects funded by industrial companies as well as federal agencies such as NSF, NASA, and ONR.

Frederick C. Harris Jr. received his BS and MS degrees in Mathematics and Educational Administration from Bob Jones University, Greenville, SC, USA in 1986 and 1988 respectively. He then went on and received his MS and Ph.D. degrees in Computer Science from Clemson University, Clemson, SC, USA in 1991 and 1994 respectively.

He is currently a Professor in the Department of Computer Science and Engineering and the Director of the High Performance Computation and Visualization Lab at the University of Nevada, Reno, USA. He is also the Nevada State EPSCoR Director and the Project Director for Nevada NSF EPSCoR. He has published more than 250 peer-reviewed journal and conference papers along with several book chapters. He has had 14 PhD students and 77 MS Thesis students finish under his supervision. His research interests are in parallel computation, computational neuroscience, computer graphics, and virtual reality. He is also a Senior Member of the ACM, and a Senior Member of the International Society for Computers and their Applications (ISCA).

Design Patterns in a Machine Learning Framework for Medical Diagnostics

Corey M. Thibeault*, Samuel G. Thorpe, Kian Jalaleddini,
Nicolas Canac, Robert B. Hamilton
Neural Analytics, Inc., Los Angeles, CA, 90064, USA

Abstract

This work presents the use of creational and structural design patterns in a generalized software framework intended to alleviate some of the difficulties in employing machine learning models in clinical applications. The design entails a configurable pipeline that supports not only the experimentation and development of diagnostic machine learning algorithms, but also their transition into production level systems. The resulting framework provides the foundation for the development of unique tools by both novice and expert users.

Key Words: Design patterns; software engineering; machine learning framework.

1 Introduction

The impact of machine learning in medicine has arguably lagged behind its commercial counterparts. This may be partly attributable to the generally slower pace and higher costs associated with the development of clinical applications, but may also reflect the conflicting constraints and requirements of learning from data in a highly regulated industry that introduce levels of complexity unique to the medical space. Because of this, the balance between innovation and controlled development is challenging. Adding to this are the multiple signal modalities found in many clinical applications for which the application of traditional machine learning preprocessing and cross-validation techniques can be precarious. Here we present the application of several classic design patterns in a generic machine learning framework, named Atlas, aimed at easing the burden of model development for clinical diagnostics. The codebase was created to assist data scientists developing machine learning algorithms for medical diagnostics. This unique combination of complex data analysis and varying levels of developer experience – data scientists do not always have extensive software engineering exposure – presents a difficult set of constraints when designing a generalized framework. This paper outlines what makes Atlas unique and how some of these classic patterns – creational and structural – were applied.

Design patterns are an important method for communicating and reusing architectural knowledge in software systems. Although the implementations are not inherently reusable, the formalized definitions provide concepts that increase quality, comprehensibility, maintainability, and testability [11]. However, amongst developers the use of patterns can be divisive – as some patterns can result in an increase in complexity and a reduction in understandability [27, 29]. This is partly due to the cognition involved in appropriately conceptualizing a particular pattern [12], as well as the contextual mismatch between software domain and pattern definitions that can result in poor design choices and reduced maintainability [3]. This is particularly true when dealing with novice developers [5].

In general, a pattern-dependent design is more complex than its pattern-free counterpart [29]. In a study of open source games Ampatzoglou et al. (2011) [1] found that the application of design patterns could significantly change defect frequency – the direction of that change was dependent on the pattern selected. This was similarly found in Vokac (2004) [26] in a study of industrial code. Regardless, patterns are an important aspect of any large-scale software project, and can in general promote good coding practices [7].

Creational patterns are a basic form of dependency inversion. Rather than relying on a concrete class definition, the instantiating class can defer the specific implementation to a subclass at runtime. This creates a mechanism for decoupling code in an application – resulting in a codebase that can be easier to extend, and more importantly, easier to test. Similarly, structural patterns rely on composition to alter the functionality of an object dynamically. A particularly useful structural design element is the decorator pattern[8]. This provides a mechanism for adding functionality to a class or instantiated object dynamically.

Atlas is the realization of several different patterns. Developed in Python [18], the framework is constructed around a dynamic class import system – allowing the instantiation of data processing blocks at runtime based on user defined configurations. The unique aspect of this is that new building blocks can be added into the processing pipeline by changing the text based configuration. This provides a dynamic development framework for data scientists, and facilitates experimentation without consequences to the overarching production level system – the first motivating factor behind developing a proprietary platform, as opposed to using off-the-shelf solutions.

Although Atlas is solving a unique problem, there are several libraries that share similarities. For example, NiftyNet [9], Nuts-ML [15], and DeepNeuro [2] were all developed to handle the preprocessing of diagnostic images exclusively for use in deep learning applications. Similarly, the Deep Learning Toolkit [16] was developed specifically for prototyping deep learning models and modules. Whereas libraries such as NifTK [6] focus on compatibility and interoperability of

*Neural Analytics, Inc. Email: corey@NeuralAnalytics.com

Figure 1: The three different stages of algorithm workflow defined here visualize the model life cycle. The experiment stage requires the most flexibility, as different methods of data extraction and machine learning are attempted. The evaluation stage allows candidate models to be retested as new data arrives – this is most often useful during long-term clinical trials. Finally, the production stage represents the deployment of the framework to end-users.

transferring imaging data. Unfortunately, there were no general purpose machine learning packages available for medical applications at the time Atlas development began.

The remainder of this paper presents the requirements that motivated the development of Atlas along with the configuration system that is the core of the framework. This is followed by an explanation of the creational and structural patterns that are employed. These are accompanied by a motivating example of how an Atlas experiment is invoked. Finally, the existing uses of Atlas and concluding remarks close out the discussion.

2 Software Requirements

There are three model stages in the workflow Atlas supports – experimentation (selection), evaluation, and production (distribution), see Figure 1. These have motivated a system that uniquely fulfills the non-functional requirements of usability, extensibility, and flexibility. While many organizations have departmental separation of these states – where data scientists develop prototype models and software engineers create structured implementations of them – a finite resource pool drove the need for a consistent interface between stages. Unfortunately, these can often introduce requirements in conflict with one another. Atlas was conceived as a way to mitigate those competing requirements while allowing data scientists to write flexible experiments that could be more readily refined for production.

2.1 Experiment Stage

The experiment stage of model development necessitates flexibility and extensibility. The generalized flow of development, outlined in Figure 2, includes preprocessing, cross validation, and evaluation. With this, Atlas facilitates the rapid creation of modeling experiments as well as a mechanism for systematically tuning analysis and model parameters. As experiments are created and released, they will be released to the evaluation system.

2.2 Evaluation Stage

The evaluation phase is the initial model release state and provides a framework for reevaluating them and their corresponding experiments as new data is collected. The basic

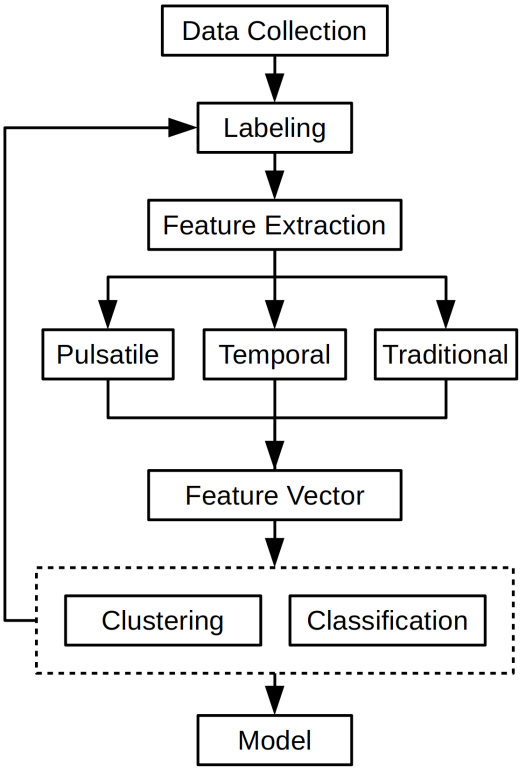


Figure 2: General experiment flow. This stage is used to design experiments and develop candidate models.

pipeline, illustrated on the left in Figure 3 , provides the infrastructure for fully exploring the potential of a candidate model by supplying mechanisms for full parameter sweeps at all stages of an experiment. This provides a way for completing these computationally intensive tasks relatively quickly. The models can be continually updated, but it requires a consistent shared interface with the experimental framework. The evaluation framework however, adds the additional requirements of tested, peer-reviewed, and documented code.

A logical extension of the evaluation system that the Atlas framework needs to support is an ensemble classifier framework. With this, as each of the different models is explored, those with promising performance or discriminatory

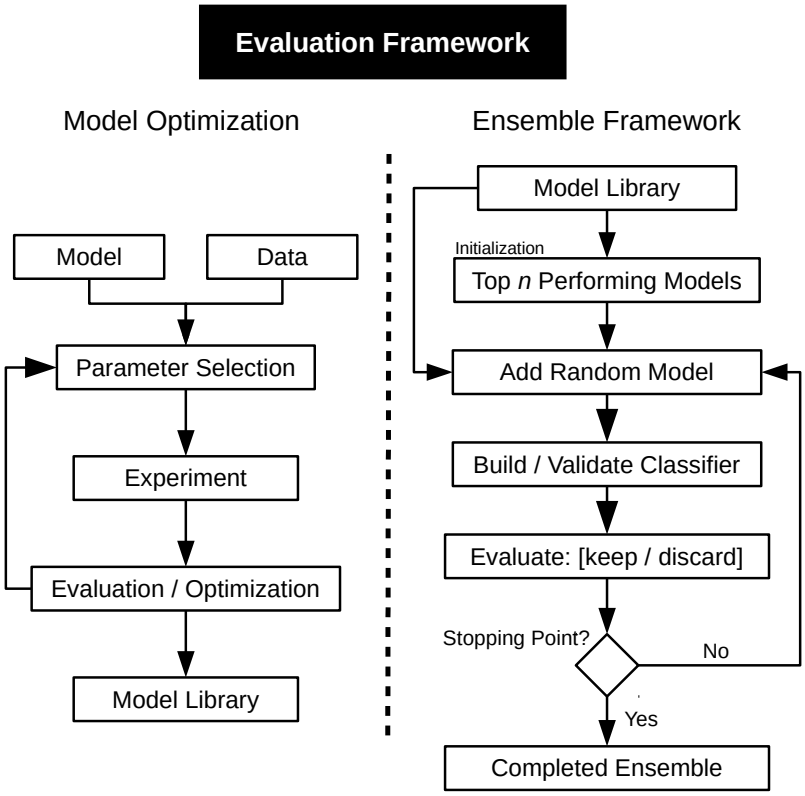


Figure 3: Evaluation Stage. (left side) Initially, the candidate models can be explored and optimized using the framework. The result of this search will be a library of successful models. (right side) The model library can then be included in an ensemble framework to create a multiple classifier system.

capabilities can be added into a library of models that are utilized in the construction of the multiple classifier system. Core to ensemble methods is the concept of hypothesis overlap – where the results of multiple models are combined to improve the overall prediction accuracy. For most well-posed problems, ensemble methods outperform individual models [10, 14]. For example, the most successful submissions, including the winning team, in the Netflix Challenge utilized ensemble approaches [13, 25]. There are a number of ways to construct ensemble systems; in this example an ensemble selection procedure presented by Caruana et al. (2006) is outlined in Figure 3 (right-side). This is a greedy method of construction – iteratively selecting a subset from a library of diverse models and combining results. The ensemble is first initialized with models that have the best independent performance. The construction is then completed by adding models, with replacement, from the library that increase performance [4].

2.3 Overall Requirements

The third and final stage, the production system, introduces regulatory, clinical, and marketing requirements. Each of these stages increases the testing and documentation burden of any project. Here, the framework and included models have to be stable and documented through a well-defined product development process. The design elements described in this paper cover the shared core of these three states but the focus and examples are on the initial evaluation stage. Each of these include a different compromise in the balance between flexibility and determinism.

Although there are a myriad of machine learning frameworks available, the development of Atlas began out of the necessity to meet the requirements of the specific three stage model process. One of the major difficulties in providing a system for developing and validating ML models, was in the preprocessing of the data. This is certainly not unique in data science – where formatting and sanitizing data is a significant step in any machine learning application [28]. However, in the case of diagnostic algorithm development, a different set of obstacles emerges. First are the multiple modalities present in most of the clinical studies generating data. The primary data source in our case is transcranial Doppler (TCD) ultrasound. But most studies include other biological signals, such as end-tidal CO_2 (the amount of carbon dioxide exhaled), absolute blood pressure, EKG, or intracranial pressure. The difficulty in incorporating all of these signals into a ML model is defining a unique uncorrelated feature set. The second source of difficulty arises during cross-validation. In most cases multiple datasets may be generated from a single subject. This is often the case for

Listing 1: Basic configuration syntax

```
module_list = module1, module2

[general]
general_bool = False
general_float = 0.65

[module1]
    [[SpecificModule1Implementation]]
    implementation_specific_int = 10

[module2]
module2_variable = String
    [[SpecificModule2Implementation]]
    implementation_specific_int = 125,
    implementation_specific_float = 10.2
```

traumatic brain injuries (TBIs), where a subject is tracked along their recovery. During cross-validation, a simple leave-one-out turns into a leave-one-subject-out – making most off-the-shelf ML packages insufficient.

3 Software Design

3.1 Configuration System

The configuration system, AtlasConfig, is built around the ConfigObj Python library. Employing a singleton pattern, AtlasConfig stores the requested processing blocks and parameters that can be accessed from within the namespace. The configuration files take the general form of Listing 1. These are dependent on the experiment and the particular processing elements, but as these are developed they become a significant part of the experiment documentation.

The configuration begins with a list of requested modules. These are core elements specific to Atlas that delineate a functional separation or data encapsulation along the processing pipeline. Not only can this list be used by the validation system for ensuring the config meets the formatting specification, but it also provides a mechanism for breaking from the traditional flow of Figure 2 without modifying the core code. The list corresponds to module definitions, further down in the configuration file, that specify the requested implementation and any sub-sections or variables required, as illustrated in Listing 1. These are fixed and have corresponding factories associated with them, but the specific implementation can be from outside of the core codebase – detailed further below.

3.2 Atlas BaseFactory

The factory method [8], is a creational pattern for abstracting away the instantiation of a concrete class to another class or subclass. The pattern defers the decision of which implementation to invoke to the factory at run-time. Unique to this work, is the utilization of the Python import framework to completely remove any prior awareness of the instantiable classes. Using this with the configuration system, data scientists can replace specific blocks of the experiment pipeline with

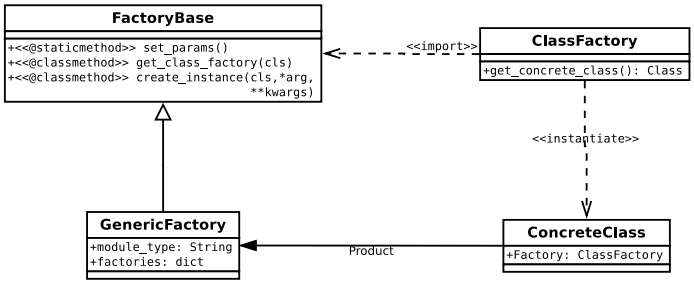


Figure 4: Atlas factory pattern class diagram.

their own implementations, without ever changing the Atlas core. This provides flexibility and encourages exploration, without risking untested or immature code diluting the shared codebase. As components mature and evolve, they can be incorporated into the common framework through the traditional development/test lifecycle. In Atlas, the factory class diagrams take the form of Figure 4. The `BaseFactory` class is inherited by all `ModuleFactories`. This contains the specific import mechanisms unique to Atlas.

An example of how this pattern is implemented is presented in Listing 2. This function takes advantage of the `@classmethod` decorator provided by Python and allows it to modify the instantiating class definition, rather than an instance. When a specific module instance is requested, this method will search for the corresponding implementation factory. If a file path is included in the configuration file, then that module will be imported and the corresponding implementation will be returned. If a path is not specified, then the function will search the associated directory structure of the `ModuleFactory` object. This pattern is utilized throughout the Atlas pipeline.

3.3 Atlas Exam Decorators

The motivation for employing a decorator pattern in an already complex framework was driven by the `Exam` classes. The different clinical studies result in a surprising number of analysis configurations. This is best exemplified in the different beat processing methods used in the creation of feature vectors. Physiologically, TCD beats correspond to the pulsatile blood flow measured in the vasculature. In Atlas feature generation, this includes everything from raw beats truncated to a common size, normalized beats, or beats averaged over each exam segment. The decorator pattern provides a mechanism for adding components and functionality at run-time – meaning that different beat processing mechanisms can be swapped into an exam by changing the configuration.

The implementation in Atlas utilizes the built-in Python `Metaclass` abstraction. Additionally, the `six` package is incorporated to allow cross-functionality between Python 2 and 3. The use of the somewhat controversial – at least in the Python community – `Metaclass` as opposed to the built-in `@decorator` keyword, came from the need for adding functionalities dynamically at run-time. Because of this, similar

Listing 2: Example of a dynamic import function in Python.

```
@classmethod
def get_class_factory(cls):
    # Get classname and path from params (AtlasConfig
        Object).
    classname, module_path = FactoryBase.__params.
        get_module_type(cls.module_type)
    # If the class has already been registered.
    if classname in cls.factories:
        return cls.factories[classname]
    # Was a path to the implementation given?
    if module_path is None:
        if not cls.initialized:
            # Search the sub-directories of the
                ModuleFactory for implementations.
            cls._get_module_paths()
            cls.initialized = True
        modules = cls.modulePaths
    else:
        # Split path on extension.
        filename, _ = os.path.splitext(module_path)
        module_name = os.path.basename(filename)
        modules = [(module_path, module_name)]
    # Search through the list of possible modules.
    for import_path, module_name in modules:
        try:
            # Try to import the module.
            mod = imp.load_source(module_name,
                import_path)
            if hasattr(mod, classname):
                # Get the implementation factory.
                module_class = getattr(mod, classname)
                module_class_factory = module_class.
                    Factory()
                # Register the factory so future
                    searches are not required.
                cls.factories[classname] =
                    module_class_factory
                # Return the factory.
                return module_class_factory
        except ImportError:
            print("Error importing {} from {}".format(
                module_name, import_path))
    # All of modules were searched and none returned the
        requested class.
    raise Exception("Module:{}, class:{}".format(
        module_path, classname))
```

to the factory methods, the use of the decorator pattern in Atlas relies on the `AtlasConfig` class to define which functionalities will be attached.

Listing 3 presents the super class implementation for decorating `exam` classes with functionalities – comparable to the classic `Component` object. The requested functionalities are first pulled from the `AtlasConfig` object and imported with the static method `import_module`. The functionality definitions are added to the concrete exam class definition and the init functions are added to the `registered_inits` list. None of the imported functions are called at this point, only the class definition is modified. An example functionality is included in Listing 4. Only two functions are implemented here, `init_beats`, which is added to the registered initializers, and `get_beats`, which is automatically added by the `MetaDecorator` class. All of this is then employed in the inheriting class illustrated in Listing 5 – this is comparable to the classical `ConcreteComponent`. In addition to loading the basic raw data, the `MetaExam`

Listing 3: Metaclass used for decorating inheriting classes with defined functionalities.

```
from atlas.config import AtlasConfig

class MetaDecorator(type):
    def __new__(mcs, name, bases, dct):
        params = AtlasConfig()

        functionalities = params.config["exam"]["
            MetaExam"].sections

        for mod_name in functionalities:
            mod_path = params.config["exam"]["MetaExam"
                ][mod_name].get(
                "module_path", None)
            module = mcs.import_module(mod_name,
                mod_path)
            explicit_decorations = getattr(module, "
                DECORATIONS", None)

            if explicit_decorations is not None:
                for func_name in explicit_decorations:
                    func = getattr(module, func_name)
                    dct[func.__name__] = func
            else:
                func = getattr(module, "get_{}".format(
                    mod_name))
                dct[func.__name__] = func

            init_method = getattr(module, "init_{}".
                format(mod_name), None)

            if init_method is not None:
                dct["__"+init_method.__name__] =
                    init_method
                dct["registered_inits"].append("__"+
                    init_method.__name__)
            else:
                print("init_method {} is None:".format(
                    mod_name))

        return super(MetaDecorator, mcs).__new__(mcs,
            name, bases, dct)

    @staticmethod
    def import_module(mod_name, mod_path):

        if mod_path is None:
            curr_dir = dirname(abspath(__file__))
            mod_path = join(curr_dir, "functionalities",
                "{}.py"
                    .format(mod_name))

        module = imp.load_source(mod_name, mod_path)
        return module
```

class loops through all of the registered functionalities and initializes them at runtime. This pattern allows for a component or abstract-oriented approach to the data containers. It has encouraged reuse of the core functionalities and added the necessary flexibility to the machine learning experiments.

3.4 Motivating Example

Running an Atlas experiment generally involves 2 steps. The first is constructing the configuration file – as illustrated in Listing 6. The second step is the basic Python code to instantiate the `Experiment` object and run the ML experiment – given in Listing 7.

After the `[general]` section in the configuration file,

Listing 4: Example implementation of a dynamic functionality.

```
from atlas.experiment_framework.preprocessing.beats
     import BeatsFactory

def init_beats(exam, dec_params):
    beats = {}
    segments = ["baseline"]
    # loop through all of the segments and create the
        beats.
    # Normally this would include a larger list of exam
        segments.
    for segment in segments:
        beats[segment] = BeatsFactory.create_instance(
            dec_params, segment)
    # add the beats as an exam attribute
    exam.beats = beats

def get_beats(exam, segment):
    return exam.beats.get(segment)
```

Listing 5: Example exam base class implementing the Metaclass.

```
import six
from atlas.config import AtlasConfig

class MetaExam(six.with_metaclass(MetaDecorator, object)
    ):
    """ This will be populated by the MetaDecorator. """
    registered_inits = []

    class Factory(object):
        """Used by the ExamFactory to generate MetaExam
            instances."""
        def create(self, *args, **kwargs):
            """Return an instance of MetaExam"""
            return MetaExam(*args, **kwargs)

    def __init__(self, params, subid, examid_string,
        sensor, is_kit):
        """Initialize the object since this does not
            happen at instantiation."""

        """
           [Raw data and other basic processing may
               occur here.]
        """

        # Run initialization for decorated
            functionalities
        for init_method in MetaExam.registered_inits:
            dec_params = self.params.config["exam"]["
                MetaExam"].get(
                init_method.split('_')[-1], {})

            try:
                dec_params.pop("module_path")
            except KeyError:
                pass

            getattr(self, init_method)(dec_params)
```

the requested modules are defined. Figure 5 illustrates the relationships between these modules. In the Preprocessing blocks the instantiated modules have a compositional relationship, with `exam` containing multiple `beats` and the `exam_list` organizing all of the `exam` objects. In the Machine Learning block however, the diagram illustrates a behavioral or data dependency interaction – with the output from a module feeding the subsequent module. These are

Listing 6: Motivating Example.

```
module_list =
    experiment, exam_list,
    beats, exam,
    ml_framework, data_set_gen,
    subspace_decomp, classifier
[general]
search_path = /data/
output_path = ~/test
extract_peaks = True
[experiment]
    [[SimpleExperiment]]
[exam_list]
    [[ExamList]]
    module_path = ~/example_examlist.py
[exam]
    [[MetaExam]]
        extract_peaks=True
        [[[beats]]]
        [[[events]]]
[beats]
    [[RawBeats]]
[ml_framework]
    [[LeaveOneOutml]]
[data_set_gen]
    [[AverageDataSetGen]]
    train_segments = baseline,
    test_segments = baseline,
[subspace_decomp]
    [[SciKitPCADecomp]]
    subspace_n_components = 5
    random_subspace = False
    random_subspace_samples = 10
[classifier]
    [[SciKitSVMClassifier]]
    theta0 = 1e-2
    svm_C = 100
```

Listing 7: Running the Experiment.

```
params = AtlasConfig("./example.cfg")
FactoryBase.set_params(params)
exp = ExperimentFactory.create_instance(params)
exp.initialize(params)
exp.run_experiment()
```

contained within a single `experiment` object that controls the instantiation and execution of the experiment.

In this example a `SimpleExperiment` is requested with a `LeaveOneOutml` cross validation. The experiment first collects the requested exams – Preprocessing block. A single subject is then removed from `exam_list` – using `AverageDataSetGen`. A subspace decomposition is then constructed using the training data with a PCA from SciKit-learn [17] – `SciKitPCADecomp`. A classifier is then constructed using an SVM model, also from SciKitLearn. The classifier is then used to predict the subject that was removed for testing. The experiment continues until the cross-validation has completed. The ability to wrap other packages – SciKitLearn in this example – is another benefit of the creational pattern. This provides an important mechanism for encouraging the inclusion of external libraries in the pipeline.

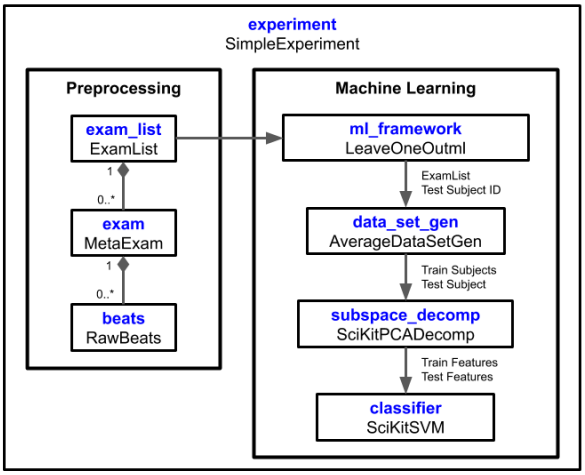


Figure 5: Motivating Example.

4 Discussion

The ultimate question of any generic framework is in how useful it actually is. Atlas has supported a number of different studies, ranging from mild traumatic brain injuries [21, 20, 22], to stroke [24, 19, 23]. However, what is most interesting about those publications is that most are based on more traditional analysis using pooled statistics or statistical models, rather than machine learning. The functional decoupling designed into Atlas has supported use cases that fall outside of the initial software requirements. This is probably the best illustration of the true extensibility that the architecture provides.

The decision to incorporate the design patterns presented here was not made without hesitation. As discussed in the introduction, including this level of extensibility and flexibility comes at a cost of reduced code readability and a level of architectural complexity that can be unnecessary at times. This framework has been through several design iterations before arriving at its current state and it could be argued that there is a reduction in the understanding of the codebase. This is particularly true for the creational aspects of Atlas. The use of the decorator pattern has not generated as much confusion – this has been the case even when the developer had no prior exposure to patterns. This observation is consistent with more rigorous studies of design patterns – where the decorator in particular has also been shown to have generally positive effects on extensibility and developer comprehension [27]. There is a learning curve to fully utilizing Atlas for novice developers. However, this only seems to hold true when trying to extend the framework, rather than during its general use. Regardless, the benefits it offers more than outweigh the developer start-up costs.

The aspect-oriented approach to exam composition has allowed for efficient changes in feature vector extraction and the module design has supported use outside of its original design. However, one important concept not discussed by these choices is performance. These design considerations can in many cases result in a significant performance drop. In this instance, the patterns are applied in places where either performance is not a concern, or other processing dominates the total computational costs – for example, the actual machine learning training and testing. If performance becomes a concern in the future, the modularized design of Atlas will allow for precise profiling and identification of bottlenecks. Furthermore, these modules can be more readily replaced with optimized implementations.

5 Conclusion

This work describes the use of creational and structural patterns in the Atlas framework as a way to mitigate the conflicting requirements of machine learning in the medical space. Outside the scope of this work are the model exploration and production systems that employ the resulting models from the experimental framework. In addition, the report generation and resulting experiment database deserve mention as well. All of these have resulted in a stable and usable system for machine learning model development. The patterns presented here are all class-scope rather than object-scope. Modifying the framework away from a class-based Adapter method would indeed be a substantial undertaking. However, the need for an object-scoped adapter has not arisen in the current use of the framework.

Funding/Support

This work was supported by the National Institute Of Neurological Disorders And Stroke of the National Institutes of Health under award numbers 1R43NS092209-01 and 2R44NS092209-02. The NIH had no role in the design and conduct of the study; collection, management, analysis, and interpretation of the data; preparation, review, or approval of the manuscript; and decision to submit the manuscript for publication. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Institutes of Health.

References

[1] Apostolos Ampatzoglou, Apostolos Kritikos, Elvira-Maria Arvanitou, Antonis Gortzis, Fragkiskos Chatziasimidis, and Ioannis Stamelos. ”An Empirical Investigation on the Impact of Design Pattern Application on Computer Game Defects”. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pp. 214–221, 2011.

[2] Andrew Beers, James Brown, Ken Chang, Katharina Hoebel, Elizabeth Gerstner, Bruce Rosen, and Jayashree Kalpathy-Cramer. ”DeepNeuro: an Open-Source Deep Learning Toolbox for Neuroimaging”. *arXiv preprint arXiv:1808.04589*, 2018.

[3] Marc Boyer and Vojislav B Mišić. "Generic Patterns: Bridging The Contextual Divide". *Grace Technical Rreports*, p. 20, 2009.

[4] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. "Ensemble Selection from Librairies of Models". In *International Conference on Machine Learning*, pp. 1–12, 2004.

[5] Alexander Chatzigeorgiou, Nikolaos Tsantalis, and Ignatios Deligiannis. "An Empirical Study on Students Ability to Comprehend Design Patterns". *Computers & Education*, 51(3):1007–1016, 2008.

[6] Matthew J. Clarkson, Gergely Zombori, Steve Thompson, Johannes Totz, Yi Song, Miklos Espak, Stian Johnsen, David Hawkes, and Sébastien Ourselin. "The NifTK Software Platform for Image-Guided Interventions: Platform Overview and NiftyLink Messaging". *International Journal of Computer Assisted Radiology and Surgery*, 10(3):301–316, Mar 2015.

[7] Daniel Feitosa, Apostolos Ampatzoglou, Paris Avgeriou, Alexander Chatzigeorgiou, and Elisa Y Nakagawa. "What Can Violations of Good Practices Tell About The Relationship Between GoF Patterns and Run-Time Quality Attributes?". *Information and Software Technology*, 105:1–16, 2019.

[8] Erich Gamma. *"Design Patterns: Elements of Reusable Object-Oriented Software"*. Pearson Education India, 1995.

[9] Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Marc Modat, Dean C. Barratt, Sebastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. "NiftyNet: A Deep-Learning Platform for Medical Imaging". *Computer Methods and Programs in Biomedicine*, 158:113–122, 2018.

[10] Isabelle Guyon, Amir Saffari, Gror Dror, and Gavin Cawley. "Model Selection: Beyond the Bayesian / Frequentist Divide". *Journal of Machine Learning Research*, 11:61–87, 2010.

[11] Günter Kniesel, Tobias Rho, and Stefan Hanenberg. "Evolvable Pattern Implementations Need Generic Aspects.". *RAM-SE*, 4:111–126, 2004.

[12] Christian Kohls and Katharina Scheiter. "The Relation Between Design Patterns and Schema Theory". In *Proceedings of the 15th Conference on Pattern Languages of Programs*, p. 15. ACM, 2008.

[13] Yehuda Koren. "The BellKor Solution to the Netflix Grand Prize". Netflix prize documentation August, 2009.

[14] Alexandre Lacoste, Hugo Larochelle, Mario Marchand, and Francois Laviolette. "Agnostic Bayesian Learning of Ensembles". In *International Conference on Machine Learning*, pp. 1–9, 2014.

[15] Stefan Maetschke, R Tennakoon, Christian Vecchiola, and Rahil Garnavi. "Nuts-Flow/ml: Data Pre-Processing for Deep Learning". *arXiv preprint arXiv:1708.06046*, 2017.

[16] Nick Pawlowski, S. Ira Ktena, Matthew C.H. Lee, Bernhard Kainz, Daniel Rueckert, Ben Glocker, and Martin Rajchl. "DLTK: State of the Art Reference Implementations for Deep Learning on Medical Images". *arXiv preprint arXiv:1711.06853*, 2017.

[17] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and douard Duchesnay. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[18] Guido Rossum. "Python Reference Manual". *CWI (Centre for Mathematics and Computer Science)*, 1995.

[19] C Thibeault, S Thorpe, S Wilk, T Devlin, and R Hamilton. "Using transcranial doppler ultrasound for the objective evaluation and prediction of endovascular treatment outcomes", 2018.

[20] Corey M Thibeault, Samuel Thorpe, Nicolas Canac, Michael J OBrien, Mina Ranjbaran, Seth J Wilk, and Robert B Hamilton. "A Model of Longitudinal Hemodynamic Alterations After Mild Traumatic Brain Injury in Adolescents". *Journal of Concussion*, 3:2059700219838654, 2019.

[21] Corey M Thibeault, Samuel Thorpe, Michael J OBrien, Nicolas Canac, Mina Ranjbaran, Ilyas Patanam, Artin Sarraf, James LeVangie, Fabien Scalzo, Seth J Wilk, Ramon Diaz-Arrastia, and Robert B Hamilton. "A Cross-Sectional Study on Cerebral Hemodynamics After Mild Traumatic Brain Injury in a Pediatric Population". *Frontiers in Neurology*, 9:200, 2018.

[22] Corey Michael Thibeault, Samuel Garrett Thorpe, Nicolas Canac, Seth J Wilk, and Robert Benjamin Hamilton. "Sex-Based Differences in Transcranial Doppler Ultrasound and Self-Reported Symptoms After Mild Traumatic Brain Injury". *Frontiers in Neurology*, 10:590, 2019.

[23] Samuel G Thorpe, Corey M Thibeault, Nicolas Canac, Seth J Wilk, Thomas Devlin, and Robert B Hamilton. "Decision Criteria for Large Vessel Occlusion Using Transcranial Doppler Waveform Morphology". *Frontiers in Neurology*, 9:847, 2018.

[24] Samuel G Thorpe, Corey M Thibeault, Seth J Wilk, Michael OBrien, Nicolas Canac, Mina Ranjbaran, Christian Devlin, Thomas Devlin, and Robert B Hamilton. "Velocity Curvature Index: a Novel Diagnostic Biomarker for Large Vessel Occlusion". *Translational Stroke Research*, pp. 1–10, 2018.

[25] Andreas Toscher, Michael Jahrer, and Robert M Bell. "The BigChaos Solution to the Netflix Grand Prize". Technical report, 2009.

[26] Marek Vokáč. "Defect Frequency and Design Patterns: An Empirical Study of Industrial Code". *IEEE Transactions on Software Engineering*, 30(12):904–917, 2004.

[27] Marek Vokáč, Walter Tichy, Dag IK Sjøberg, Erik Arisholm, and Magne Aldrin. "A Controlled Experiment Comparing the Maintainability of Programs Designed With and Without Design Patternsa Replication in a Real Programming Environment". *Empirical Software Engineering*, 9(3):149–195, 2004.

[28] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *"Data Mining: Practical machine learning tools and techniques"*. Morgan Kaufmann, 2016.

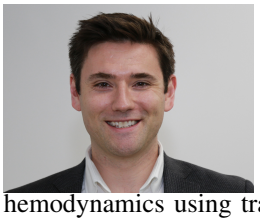
[29] Liguo Yu, Yingmei Li, and Srini Ramaswamy. "Design Patterns and Design Quality: Theoretical Analysis, Empirical Study, and User Experience". *International Journal of Secure Software Engineering (IJSSE)*, 8(2):53–81, 2017.

Corey M. Thibeault is Director of Data Science at Neural Analytics and has been working with the company for the last 5 years to understand changes in cerebral hemodynamics after brain injury including TBI and stroke. Prior to Neural Analytics, Dr. Thibeault was a postdoctoral fellow at UCLA and HRL Laboratories. He holds advanced degrees in mechanical engineering, computer engineering, and biomedical engineering.

Samuel G. Thorpe is a Senior Data Scientist at Neural Analytics, Inc. in Los Angeles, CA. He received his PhD in 2012 from the Institute for Mathematical Behavioral Science at the University of California, Irvine. His areas of expertise include cerebral hemodynamics, and cognitive/computational neuroscience. His research uses multi-modal physiological time series analysis (TCD, EEG/MEG, ECG, fMRI) to study hemodynamic pathologies such as Acute Ischemic Stroke and Traumatic Brain Injury, as well as human spatial attention and motor development.

Kian Jalaleddini is currently a Senior Data Scientist at Neural Analytics, Inc and his research is focused on advanced analytical techniques to understand pathophysiology of cerebral hemodynamics. He was a Post-Doctoral Scholar with the Division of Biokinesiology and Physical Therapy, University of Southern California from 2015 to 2017. He received his Ph.D. degree in Biomedical Engineering from McGill University, Montreal, Canada and MSc and BSc degrees in Electrical Engineering. His research interests include analysis of Biomedical signals and systems, development and application of system identification tools, transcranial Doppler ultrasound, and neuromechanics of human joints and spinal reflexes. Dr.Jalaleddini has served as the Chair of the IEEE Engineering in Medicine and Biology Chapter, Montreal Section from 2011 to 2015, and Secretary of the IEEE Montreal Section from 2010 to 2013.

Nicolas Canac received his B.S. from the University of Texas at Austin (2010), and an M.S. (2014) and Ph.D. (2016) in astrophysics from the University of California, Irvine. He joined Neural Analytics in 2016, where he has been involved in the research and development of novel science and algorithms aimed at improving the assessment, diagnosis, and tracking of brain health through the use of Transcranial Doppler ultrasonography. His primary research interests include applying signal processing methods and machine learning algorithms to cerebral blood flow velocity measurements in order to aid in the diagnosis of neurovascular conditions such as stroke, traumatic brain injury, and intracranial hypertension.



Dr. Robert Hamilton is Co-founder and Chief Scientific Officer at Neural Analytics and has been leading the company for the past 7 years in their mission to preserve brain health by analyzing changes in cerebral hemodynamics using transcranial Doppler ultrasound. Before co-founding Neural Analytics, Dr. Hamilton performed his graduate work in the Department of Neurosurgery at UCLA. Robert holds advanced degrees in biomedical engineering and mathematics.

Journal Submission

The International Journal of Computers and Their Applications is published four times a year with the purpose of providing a forum for state-of-the-art developments and research in the theory and design of computers, as well as current innovative activities in the applications of computers. In contrast to other journals, this journal focuses on emerging computer technologies with emphasis on the applicability to real world problems. Current areas of particular interest include, but are not limited to: architecture, networks, intelligent systems, parallel and distributed computing, software and information engineering, and computer applications (e.g., engineering, medicine, business, education, etc.). All papers are subject to peer review before selection.

A. Procedure for Submission of a Technical Paper for Consideration

1. Email your manuscript to the Editor-in-Chief, Dr. Ziping Liu at: zliu@semo.edu.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page (or include in the email message) the preferred author and address for correspondence. Also, please include email, telephone, and fax information should further contact be needed.
4. **Note**: Papers shorter than 10 pages long will be returned.

B. Manuscript Style:

1. The text should be **double-spaced** (12 point or larger), **single column** and **single-sided** on 8.5 X 11 inch pages.
2. An informative abstract of 100-250 words should be provided.
3. At least 5 keywords following the abstract describing the paper topics.
4. References (alphabetized by first author) should appear at the end of the paper, as follows: author(s), first initials followed by last name, title in quotation marks, periodical, volume, inclusive page numbers, month and year.
5. The figures are to be integrated in the text after referenced in the text.

C. Submission of Accepted Manuscripts

1. The final complete paper (with abstract, figures, tables, and keywords) satisfying Section B above in **MS Word format** should be submitted to the Editor-in-Chief. If one wished to use LaTex, please see the corresponding LaTex template.
2. The submission may be on a CD/DVD or as an email attachment(s). **The following electronic files should be included:**
   - Paper text (required).
   - Bios (required for each author).
   - Author Photos (jpeg files are required) or photos can be integrated into the text.
   - Figures, Tables, and Illustrations. These should be integrated into the paper text file.
3. Reminder: The authors photos and short bios should be integrated into the text at the end of the paper. All figures, tables, and illustrations should be integrated into the text.
4. The final paper should be submitted in (a) pdf AND (b) either Word or LaTex. For those authors using LaTex, please follow the guidelines and template.
5. Authors are asked to sign an ISCA copyright form (http://www.isca-hq.org/j-copyright.htm), indicating that they are transferring the copyright to ISCA or declaring the work to be government-sponsored work in the public domain. Also, letters of permission for inclusion of non-original materials are required.

Publication Charges

After a manuscript has been accepted for publication, the contact author will be invoiced a publication charge of **$500.00 USD** to cover part of the cost of publication. For ISCA members, publication charges are **$400.00 USD** publication charges are required.

Revised 2019