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

INTERNATIONAL JOURNAL OF COMPUTERS AND THEIR APPLICATIONS

TABLE OF CONTENTS

Page

Guest Editorial: Special Issue from ISCA CAINE-2017 53
Gongzhu Hu and Takaaki Goto

Efficient PTS Algorithm of PAPR Reduction for improving OFDMA Wireless Communication System Behavior 54
Majed Albogame, Junghwan Kim and Mohammad Niamat

Applying Task Models from Human Computer Interaction to Support and Improve Usage Based Statistical Testing for Web Applications 64
Gity Karami and Jeff Tian

An Empirical Algorithm for Turn Detection from Vehicle Data 76
Jennifer Knull, Steven Beauchemin, and Michael Bauer

Using Orbital Network for Scalable Multi-Core 84
Nagi Mekhiel

Use of Ethereum Blockchain for Authentication, Access Control, and Data Sharing in Untrusted Environments 91
Syed Hasnain, Abhinav Kalra, Peter Bodorik, Dawn Jutla, and Sandeep Kuri

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

Volume 25, No. 2, June 2018 **ISSN 1076-5204**

International Journal of Computers and Their Applications

A publication of the International Society for Computers and Their Applications

ISCA Headquarters···•···P. O. Box 1124, Winona, MN 55987 USA···•···Phone: (507) 458-4517
E-mail: isca@ipass.net • URL: http://www.isca@isca-hq.org.

Guest Editorial
Special Issue from ISCA CAINE 2017

This Special Issue of IJCA is a collection of five refereed papers selected from the 3*0th International Conference on Computer Applications in Industry and Engineering* (CAINE 2017).

Each paper submitted to the conference was reviewed by at least two members of the International Program Committee and additional reviewers, judging the originality, technical contribution, significance and quality of presentation. After the conference, a number of high quality papers were recommended by the Program Chair 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, five papers were accepted for publication in this issue of the journal.

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

Majed Albogame, Junghwan Kim and Mohammad Niamat of University of Toledo, USA, described performance of OFDMA using PAPR-PTS reduction technique in their paper "*Efficient PTS Algorithm of PAPR Reduction for improving OFDMA Wireless Communication System Behavior*." They discussed OFDM technology, the OFDMA technique concepts with Ns point's subcarriers and its transmission block diagrams, and the improvement OFDMA-PAPR reduction by applying PTS technique.

Gity Karami of Southern Methodist University, USA, and and Jeff Tian of Northwestern Polytechnical University, China, in the paper "*Applying Task Models from Human Computer Interaction to Support and Improve Usage Based Statistical Testing for Web Applications*, " proposed a method utilizing task models to maintain accuracy of the Markov OP, and evaluated the effectiveness of the proposed method in case studies.

Jennifer Knull, Steven Beauchemin, and Michael Bauer of The University of Western Ontario, Canada, presented a method for detecting turns from vehicle data with two turn detection algorithms in the paper "*An Empirical Algorithm for Turn Detection from Vehicle Data*". One of the two algorithms relies on vehicular data and GPS coordinates and the other is based solely on vehicular data.

Nagi Mekhiel of Ryerson University, Canada, in the paper "*Using Orbital Network for Scalable Multi-Core*, " proposed an Orbit Network on a Chip (NoC) approach that addressed the problem of waiting time of messages between processors on a multi-core. The proposed approach is to map message to time and using a shared orbit for processors.

Syed Hasnain, Abhinav Kalra, Peter Bodorik, Sandeep Kuri of Dalhousie University, Canada, and Dawn Jutla of Saint Mary's University, Canada, in the paper "*Use of Ethereum Blockchain for Authentication, Access Control, and Data Sharing in Untrusted Environments*," examined the various architectures of incorporating blockchain to existing data sharing solutions while also establishing a need for blockchain-based data sharing solutions. The experiments show the advantage of the proposed method.

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

Guest Editors:
Gongzhu Hu, Central Michigan University, USA, CAINE 2017 Conference Chair
Takaaki Goto, Ryutsu Keizai University, Japan, CAINE 2017 Program Chair

May 2018

Efficient PTS Algorithm of PAPR Reduction for Improving OFDMA Wireless Communication System Behavior

Majed Albogame*, Junghwan Kim*, and Mohammed Niamat*
University of Toledo, Toledo, OH, 43606, USA

Abstract

Wireless telecommunications have been growing significantly due to the need of a high data rate during recent years. The basic Orthogonal Frequency Division Multiplexing Access (OFDMA) concepts are introduced in an easy way to be understood. OFDMA is an advanced technique which plays a key role in the 4G and 5G systems for future wireless communication. OFDMA combines Orthogonal Frequency Division Multiplexing (OFDM) with Frequency Division Multiplexing Access (FDMA) so as to exploit the advantages of both for supporting multiuser wireless communications. It is characterized by a high spectral efficiency, which is required for high data rate transmissions. OFDMA with different scenarios are tested by MATLAB programs and the results indicate that the OFDMA technique will be the advanced technology providing the need of high data rate and high performance in wireless communication uses. Furthermore, the complexity of OFDMA technology can be reduced and the capacity is increased by using Peak to Average Power Ratio Reduction (PAPR-R) associated with the Partial Transmit Sequences (PTS) method. However, this method promised the best performance of OFDMA in efficient manner and high quality for different M-QAM modulation. Increased capacity, coverage, and reliability are clearly evident from the test results presented in this work.

Key Words: Orthogonal frequency division multiplexing (OFDM), orthogonal frequency division multiplexing access (OFDMA), fast fourier transform (FFT), peak to average power ratio reduction (PAPR-R), inter symbol interference (ISI), inter carrier modulation (ICI), complementary cumulative distribution function (CCDF), partial transmit sequences (PTS).

1 Introduction

Multiplexing is effective for transmission efficiency. It is used to allocate the total capacity of a transmission medium among a number of users. OFDMA is also known as a multiuser of OFDM multicarrier modulation technique, which has been the key behind the most advances and achievements associated with high data rate communications. OFDMA nowadays has become so popular due to flexible and efficient management of Inter-Symbol Interference (ISI). Moreover, OFDMA grants the high spectral efficiency and mitigates the multipath environment effects. These positive consequences happened as a result of OFDMA concept which divides the data stream of each OFDM symbol into multiple sub-streams and sending them through multiple orthogonal sub-channels for each user. Generally, the overall system performance and communication link quality will be improved by OFDMA technology [4, 14] and [16].

In this journal, we will discuss OFDM technology in detail, the OFDMA technique concepts with Ns point's subcarriers and its transmission block diagrams, and finally the improvement OFDMA-PAPR reduction by applying PTS technique. The simulation examines the results of using MATLAB to prove the ability of OFDMA technique having capacity and validity for different scenarios of M-QAM modulation.

2 Related Work

Parallel data transmission was proposed to increase data processing and transmission traffic due to the desired demand of high data rate in communication networks. In 1960 Frequency Division Multiplexing (FDM) was introduced as a technique in parallel data transmission [3].

FDM divides the frequency spectrum into slots we called transmission channels. Also, each slot should be assigned to one transmission channel. These frequency sub-bands cannot be overlapped nor can be placed adjacently, which means that multicarrier FDM communication link designers should not only assign non-over-lapped frequency bands for transmission channels but also leave frequency guard bands in between those assigned frequency bands [14]. Consequently, multicarrier modulation technique, had mainly two dis-advantages. The first one was the implementation complexity and the second one was the spectral inefficiency.

The solution came up officially in 1970 when the OFDM patent was issued in the United States of America. OFDM, at that time, handled one of the conventional FDM multicarrier problems which is spectral inefficiency. The final method was presented by Weinstein and Ebert in 1971 when they proposed a multicarrier system which works with Discrete Fourier Transform (DFT) [5]. DFT was implemented as a part of the modulation and demodulation process in the multicarrier

* Department of Electrical Engineering and Computer Science, (Majed.Albogame, Jung.Kim, Mohammed.Niamat)@utoledo.edu.

communication systems to reduce complexity. Inverse Discrete Fourier Transform (IDFT) and DFT, according to Weinstein and Ebert proposed approach, are assigned to perform the modulation and demodulation processes, respectively. Many research efforts have been made to improve multicarrier communication system complexity. The most successful improvement is the use of Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) instead of DFT and IDFT respectively. By that time, both system complexity and bandwidth inefficiency problems were handled through modifications being added one after another and the last one was introduced by Weinstein and Ebert as mentioned before. Besides handling system complexity and spectral inefficiency, the other advantages of OFDM system are free ISI and flat frequency fading communication link.

Recently, the multiple access schemes have attracted the communication technologies. They are allowed multiple users to communicate by simultaneously sharing the same communication channel. Actually, the way to have multiple users is by using the frequency, time or code multiplexing, so there are three basic principles in multiple access, Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), and Code Division Multiple Access (CDMA). However, OFDMA uses OFDM symbols to divide the high rate input data stream into lower rate multiple sub-streams. Then, those sub-streams are modulated on orthogonal subcarriers offering high spectral efficiency, frequency flat fading sub-channels, free ISI communication link, and robust modulation for each user [15] and [17]. In fact, OFDMA is a capable technique that makes the system more flexible, reliable and stronger.

The OFDMA multiple access provides orthogonally between the users, reducing the interference and improving the network capacity [13]. OFDMA can attain the best features than different multiple access such as TDMA, FDMA and CDMA [16]. The OFDMA system has many significant features that work perfectly with the recent wireless communication standards. Those features are spectral efficiency, flexibility in adapting different modulations, and simpler channel equalization.

3 OFDM Technology

OFDM is a multi-carrier communication system. It divides the high rate data stream into lower rate sub-streams and sends them over multiple parallel sub-channels [7]. It has fantastic properties which will be discussed in the next sub-sections that make its advanced system.

3.1 OFDM Orthogonality

OFDM satisfies orthogonality of communication system subcarriers which means that a mathematical relationship between these subcarriers is developed to allow these subcarriers to overlap without having any adjacent sub-channels interference [6]. In an OFDM system, we have specified frequency spectrum within N equally spaced subcarriers will be satisfied the orthogonality property when the frequency spacing Δf equals to:

$$\Delta f = 1/(N.T_S) \quad (1)$$

Where $N.T_S$ = Symbol duration [6].

This mathematical relationship creates a sine frequency response for all these N subcarriers where each one of them has maximum amplitude at a point, whereas others have nulls [6]. Figure 1 shows this unique OFDM aspect clearly for 4 OFDM subcarriers.

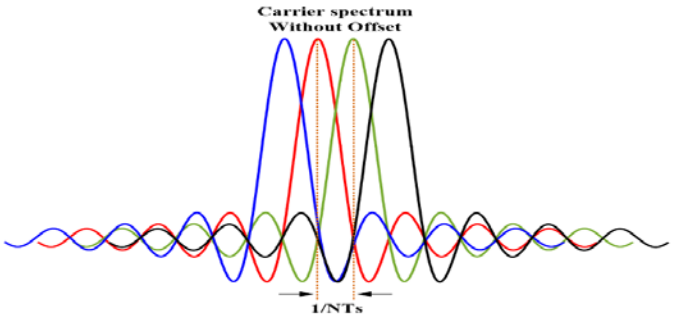

Figure 1: OFDM orthogonality property for four subcarriers [5]

Also, Equation (2) is a mathematical proof, which shows there is no interference between signals if spaced as in Equation (1), is provided in [12] as the following:

$$\int_0^{N.Ts} X_K(t)\, X_L{}^*(t) = \begin{cases} 0, K \neq L \\ C, K = L \end{cases} \quad (2)$$

Where $X_K(t)$ and $X_L(t)$ are two interference signals, over time $N.T_S$ and described the orthogonality if there is or not between them depending on the integration result either zero or Constant (C) value.

Because of this orthogonality aspect, OFDM system occupies the frequency spectrum efficiently and wisely as it has been shown in Figure 2. According to [11], OFDM multi carrier system occupies 50% of the bandwidth that regular FDM

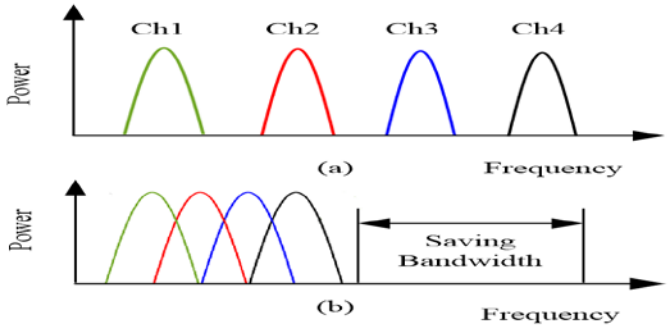

Figure 2: Shows what the differences between the FDM and OFDM in spectral efficiencies

multicarrier system does.

3.2 ISI, ICI, guard time, CP in OFDM system

OFDM mitigates the effects of multipath delay spread significantly. By dividing the data stream into N sub-streams, the symbol duration time of each sub-stream is smaller than the original one by a factor of (1/Ns) [11]. This shorter symbol duration helps reduce the effects of multipath delay spread by a factor of (1/Ns) as well.

ISI is an unwanted phenomenon which happens during the communication process when one symbol interferes with the adjacent symbols [1]. In addition, ISI becomes an ever more serious issue with high data rate communication systems so this phenomenon needs to be overcome to maintain communication quality. OFDM has handled the ISI problem by dividing high data rate stream wanted to be transmitted into (L) lower rate sub-streams such that each one has symbol time (Ts/L) >> channel delay spread time (τ). This property gives almost free ISI communication, as a result of ISI criterion mentioned above [1]. Moreover, OFDM can eliminate ISI problem and improve the results even more by inserting guard time slots between OFDM symbols as it is shown in Figure 3. These guard slots should be chosen to be larger than the expected symbol delay spread time. Doing so, OFDM ensures that multiple components received of one symbol due to multipath environment do not interfere with the adjacent symbols components [11].

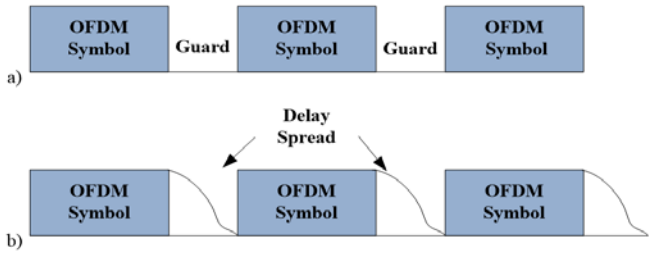

Figure 3: (a) and (b) shows how the guard band is left in between OFDM symbols [11]

ICI is the consequences of losing subcarriers orthogonality in the OFDM system and that happens when the receiver tries to retrieve one of the subcarriers while a portion from the adjacent symbol is being added causing ICI [9]. This simply means when OFDM subcarriers are not properly synchronized, the problem of ICI would arise, or i.e. when having frequency offset between OFDM subcarriers as it is illustrated in Figure 4.

The proposed solution for this problem is adding Cyclic Prefix (CP). CP is a copy of the last portion of OFDM symbol being added in front of that symbol. CP was proposed first in 1980 by Peled and Ruiz as in [12]. Their idea was to occupy the time guard slot of any OFDM symbol with the last portion of that symbol as it is shown in Figure 5. That portion duration chosen to have Ng samples.

This time duration should be longer than the longest expected symbol delay spread. Therefore, the transmitted signal will have (Ng + N) samples. The advantage of CP is to create an integer number of cycles of each subcarrier when being passed through FFT process or simply to maintain subcarriers orthogonality [11].

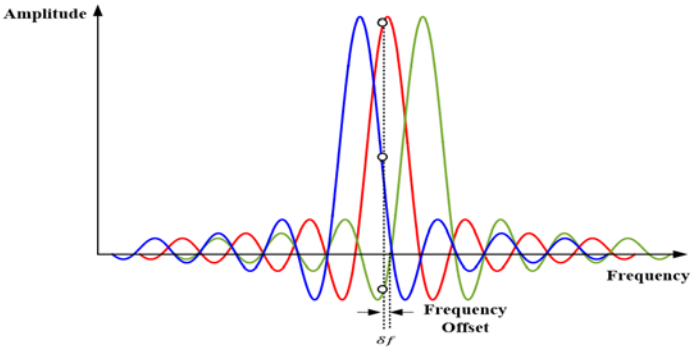

Figure 4: Shows frequency offset problem which yields to ICI

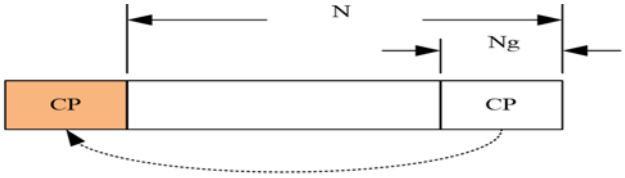

Figure 5: Shows CP procedure

3.3 OFDM Frequency Flat Fading

OFDM has been the most known modulation scheme candidate for high data rate wireless communications. OFDM does group high data rate input streams into smaller lower rate multiple sub-streams to ensure that each sub-stream would have a flat fading not a frequency selective one [2]. Selective frequency fading takes place when having a wide range of frequencies and some of these frequencies are by nature more vulnerable to be faded than other frequencies are. Therefore, when OFDM divides the wideband frequencies into narrower frequency sub-bands, those narrow bands will experience the same fading effects along that sub-band and that what is called flat frequency fading. It is known that flat frequency fading consequences are lighter and easier to be taken care of than the selective frequency fading consequences are. One of the most valued benefits of not having frequency selective fading is that the receiver no longer needs complex equalizers and RAKE receivers [2]. To be more specific, a communication system with flat frequency fading needs only a simple equalizer at the receiver side.

4 OFDM in Multiple Access Techniques

OFDM is used for transmitting the symbols for a single user. However, OFDM can be operated with multiple access schemes such as TDMA, FDMA and CDMA which are all used to allow many users to share simultaneously a finite amount of radio

spectrum. Consequently, all subcarriers can be shared by multiple users in the forms of (OFDM-TDMA), OFDMA (OFDM-FDMA), and MC-CDMA (OFDM-CDMA) as illustrated in Figure 6.

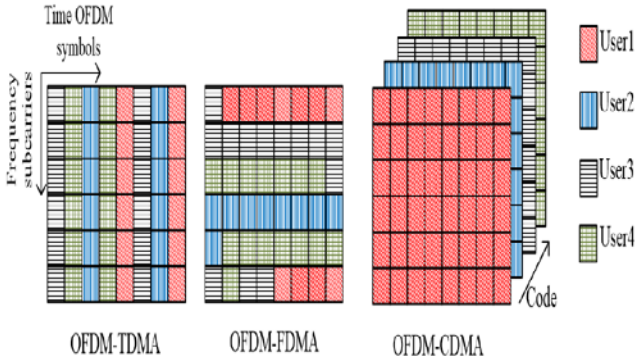

Figure 6: Multiple access techniques used in OFDM systems [2]

In the OFDM-TDMA system, all subcarriers assign to a single user for duration of several OFDM symbols and each user can be adaptively changed by the OFDM symbols in each frame. Those frames are orthogonal in time. OFDM-FDMA (OFDMA) system, some of the subcarriers allocated to each user and user's subcarriers can be adaptively varied in each frame. In other words, the subcarriers in each OFDM symbol are orthogonally divided among the multiple users. But, in the MC-CDMA (OFDM-CDMA) system, all users can share both time and subcarriers (not in orthogonal manner). The subset of orthogonal codes is assigned to each user and the information symbols are spread in the frequency domain [2]. Among these multiple access techniques associated with OFDM, OFDMA performs the best and has incredible approaches in the desired communication applications. The OFDMA scheme will be discussed in details next sections.

4.1 OFDMA Concepts

OFDMA is a multi-user version of the OFDM. In general, OFDMA is a hybrid of FDMA and TDMA users dynamically assigned subcarriers FDMA in different time slots TDMA [1]. So, the multi-user is manageable by the assignment of subsets of subcarriers to users individually. The OFDMA provides orthogonally between the users, reducing the interference and improving the network capacity [13]. The main idea of the OFDMA is to divide the one high data rate stream into several low data rate streams and transmit all into parallel to reach the desired user.

OFDMA is clearly a multiple access/multiplexing scheme that provides multiplexing operation of user data streams into the downlink sub-channels and uplink multiple access by means of uplink sub-channels. Therefore, there are closely separated multiple subcarriers of OFDMA system. The subcarriers are divided into groups of subcarriers. Each group is named a sub-channel and its subcarriers need not to be adjacent. To assign the subcarriers or sub-channel, there are three kinds of Carrier Assignment Schemes (CAS) used in OFDMA, called sub-band CAS, interleaved CAS and generalized CAS [8]. Generalized CAS supplies more flexibility than sub-band or interleaved CAS. The three CAS strategies for four users are shown in Figure 7.

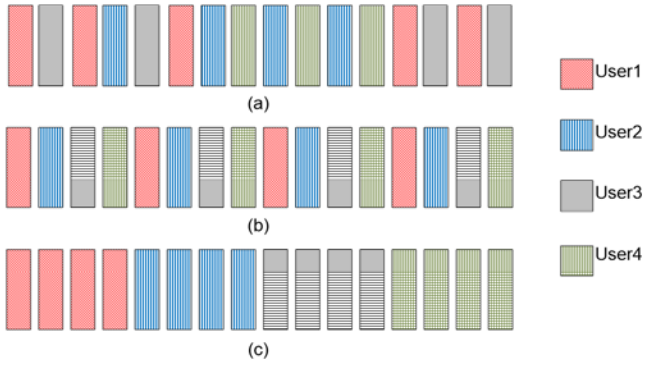

Figure 7: CAS Schemes: (a) Generalized (b) Interleaved (c) Sub-band for four users

The multiple access is achieved by assigning different OFDM sub-channels to different users. In the downlink, a sub-channel may be intended for different receivers. In the uplink, a transmitter may be assigned one or more sub-channels as are shown in Figure 8.

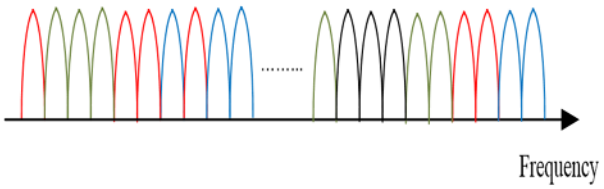

Figure 8: OFDMA-Subcarriers with the same color represent a sub-channel

Principally, each user has to know what frequency band it is allowed to use and maps its data symbols onto the matching subcarriers. Also, each subcarrier can independently assign its own modulation scheme. In OFDMA system, the famous modulation schemes are Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), and Mary-Quadrature Amplitude Modulation (M-QAM). Among those modulation types, we will implement (M-QAM) modulation because it has provided the best performance than others.

4.2 OFDMA Transmission Block Diagram

In the case of OFDMA, subcarriers for several users instead of a single user - are multiplexed or combined into a larger set of subcarriers. The block diagrams of OFDMA transmitter and receiver sides explain briefly how the OFDMA system works as shown in Figure 9 below. OFDMA block diagram are similar to

OFDM system, except the additional subcarriers mapping and the position of some blocks.

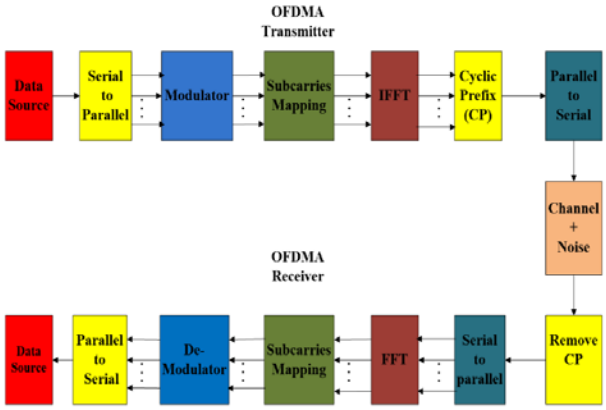

Figure 9: Block diagram of an OFDMA system

In the OFDMA transmitter, all parallel sequences of modulated symbols is mapped with N subcarriers for each user to which the IFFT is applied. The output samples signal is converted into serial sequences to add CP, consisting of an OFDM symbol. Obviously, CP is added to the OFDM symbol to protect against interference between OFDM symbols and against the loss of orthogonality due to the multipath channel. Then after some additional processing (windowing,Digital-to-Analog(D/A) conversion, frequency up-conversion, RF processing, etc.) the OFDM signal is transmitted over the radio channel.

In the OFDMA receiver, after some conventional processing not explicitly shown, for each user the cyclic prefix is removed from the OFDM symbol, the signal is converted into a parallel set of samples, and the FFT is applied to these samples. Using the reference or pilot signals, the equalizer is able to remove the amplitude and phase distortion of the signal-carrying subcarriers. Finally, these subcarriers are de-mapped and converted into the original serial symbol sequences.

5 Peak to Average Power Ratio (PAPR) in OFDMA

One of the OFDMA implementation challenges will be addressed in this section, and that challenge is the PAPR. The major challenge of implementing OFDMA system is the high PAPR. The relationship between the OFDMA subcarriers and PAPR is proportional to each other. Increasing the number of subcarriers will increase PAPR value. As we know, the IFFT procedure in OFDMA system involves adding up N subcarriers which are all sinusoids. This summation of N subcarriers, at some points gives high peak values compared to the average ones and that is well known as high PAPR. This high PAPR is simply described as an envelope fluctuation which causes many issues that degrade the system performance as described in [1], [4], and [7]. In fact, high PAPR has a direct impact on the Bit Error Rate (BER) causing more likely errors to happen. It also causes ISI at the receiver side when trying to get the received data decoded.

Generally, the PAPR of OFDM signals x(t) is defined as the ratio between the maximum instantaneous power and its average power as shown in Equations (3) and (4):

$$\text{PAPR}[\,\text{x(t)}] = \frac{P_{Peak}}{P_{Average}} = 10\ log_{10}\frac{max[|x(n)|^2]}{E[|x(n)|^2]}\ \text{dB} \quad (3)$$

And $x(n)$ is expressed as:

$$x(n) = \frac{1}{\sqrt{N}}\sum_{K=0}^{N-1} X_K\ W_N^{nk} \quad (4)$$

Where P_{Peak} represents peak output power, $P_{Average}$ means average output power. E[.] denotes the expected operator, $x(n)$ represents the transmitted OFDM signals which are obtained by taking IFFT operation on modulated input symbols X_K. N is the total number of subcarriers. $W_N^{nk} = e^{-j2\pi\frac{nK}{N}}$ is the twiddle factors.

The PAPR can also be described in terms of their magnitudes (not power) by defining the Crest Factor (CF), which is defined as the ratio between maximum amplitude of OFDM signal x(t) and Root-Mean-Square (RMS) of the waveform. The CF is defined in Equation (5):

$$\text{CF} = \frac{max[|x(t)|^2]}{E[||x(t)|^2|]} = \sqrt{PAPR} \quad (5)$$

The most commonly used technique to evaluate the PAPR system is the Cumulative Distribution Function (CDF). In the most recent publications, the Complementary Cumulative Distribution Function (CCDF), which can be derived from the CDF, is also used to evaluate the PAPR system. The CCDF function shows that the system probability of PAPR would exceed certain threshold values known as PAPR_0. The Equations (6) and (7) are used to get the CDF and CCDF functions respectively. To get the CDF, the following Equation is applied:

$$\text{F(z)} = 1 - \exp(\text{z}) \quad (6)$$

Then, to derive the CCDF, the following set of Equations are applied:

$$\begin{aligned}\text{P(PAPR} > \text{z)} &= 1 - \text{P(PAPR} \le \text{z)} \\ &= 1 - \text{F(z)}^N \\ &= 1 - (1 - \exp(-\text{z}))^N \end{aligned} \quad (7)$$

Where, N is the number of subcarriers and z is the given reference level.

6 Partial Transmit Sequences (PTS) in OFDMA

PTS is one of the best techniques that reduces the effect of the

PAPR in OFDMA scheme. In fact, for this technique, the input data block is divided into multiple non-overlapping M sub-blocks and each one of these sub-blocks has multiple subcarriers. For each sub-block, the subcarriers are weighted by a phase factor. These phase factors are developed in a way to get the PAPR of the combined OFDMA signal reduced [10]. The block diagram of PTS is shown in Figure 10.

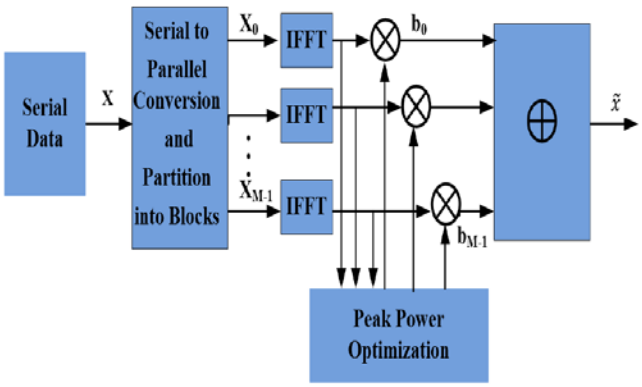

Figure 10: PTS system block diagram [7]

Clearly, from the Figure 10 above, the input data block in X is partitioned into M disjoint sub blocks which can be represented as vectors as is shown in Equation (8):

$$\mathbf{X} = \sum_{m=0}^{M-1} \mathbf{X}^{(m)} \tag{8}$$

Where $\mathbf{X}^{(m)} = \left[X_0^{(m)}\ X_1^{(m)}\ \ldots\ X_{N-1}^{(m)}\right]$

With $\boldsymbol{X}_K^{(m)} = \boldsymbol{X}_K\ or\ 0\ , and\ (0 \le m \le M-1)$

Then, the sub blocks are transformed $\mathbf{X}^{(m)}$ into M time-domain partial transmit sequences L time oversampling as is shown in Equation (9):

$$\mathbf{x}^{(m)} = \left[x_0^{(m)}\ x_1^{(m)}\ \ldots\ x_{LN-1}^{(m)}\right] = IFFT_{\ LN\times N}\left[\boldsymbol{X}^{(m)}\right] \tag{9}$$

These partial sequences are independently rotated by phase factors $\mathbf{b} = \{b_{\boldsymbol{m}} = \boldsymbol{e^{j\theta_m}},\ \ m = 0, 1, \ldots, M-1\}$. The combined sub blocks are optimized obtaining the time domain OFDMA signals with the lowest PAPR as is shown in Equation (10):

$$\tilde{\mathbf{x}} = \sum_{m=0}^{M-1} b_{\boldsymbol{m}}\mathbf{x}^{(m)} \tag{10}$$

The OFDMA system can be evaluated using the PTS-OFDMA efficiency between new PTS-OFDMA and the original PTS-OFDMA as is illustrated in Equation (11):

$$\text{PAPR Efficiency} = \frac{New\ PTS{-}OFDMA}{Original\ PTS{-}OFDMA} \tag{11}$$

In the PTS PAPR reduction scheme, there are several methods for the partition of the data sequence into multiple sub-blocks, including adjacent partition, interleaved partition and pseudorandom partition [4]. Among them, pseudo-random partitioning has been found to be the best choice.

7 Results

The simulation results show the response of OFDM signals by calculating its PAPR based on different IFFT sample points Ns, 64, 128,256,512 and 1024 as shown in Figure 11. The x-axis represents the diverse OFDM values, while y-axis represents the CCDF values. Also, the simulations from Figure 12 to Figure 16 show that OFDMA scheme becomes an efficient manner when some PTS are chosen for different sub blocks points from M = 1, 2, 4, 8, and 16 and various M-QAM symbols. The x-axis represents the PAPR (dB) values, while y-axis represents the probability of (PAPR > $PAPR_0$) values. Figure 17 represents the relation between the PTS-OFDMA system efficiency and the probability depending on the results extracting from Table 1 to Table 5.

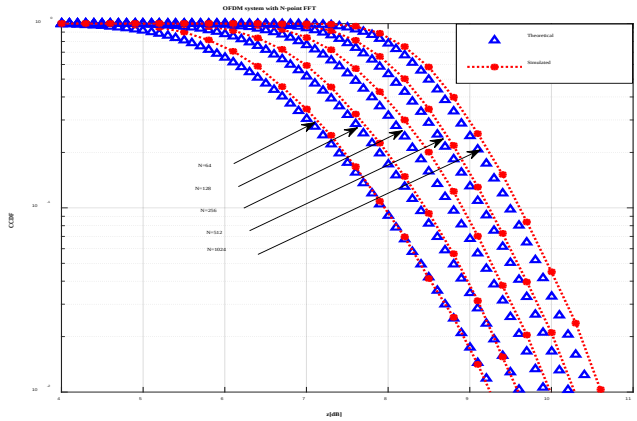

Figure 11: The relation between OFDM and CCDF for different points with Ns = 64, 128, 256, 512, and 1024

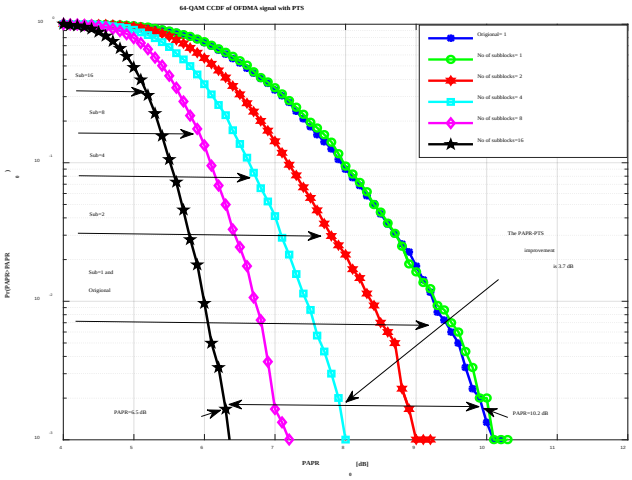

Figure 12: The relation of 64 QAM CCDF signal with PTS for different sub blocks points at M = 1, 2, 4, 8, and 16

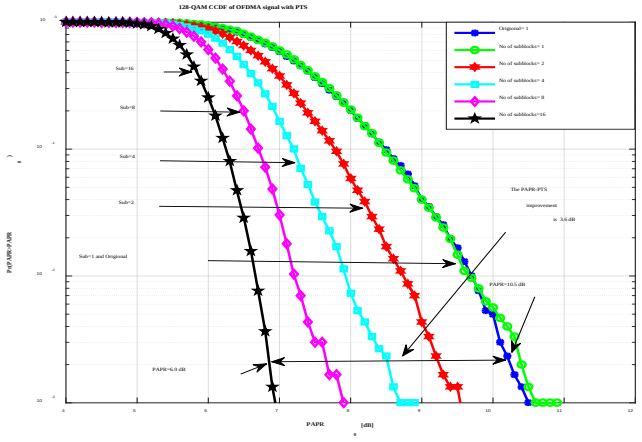

Figure 13: The relation of 128 QAM CCDF signal with PTS for different sub blocks points at M =1, 2, 4, 8, and 16

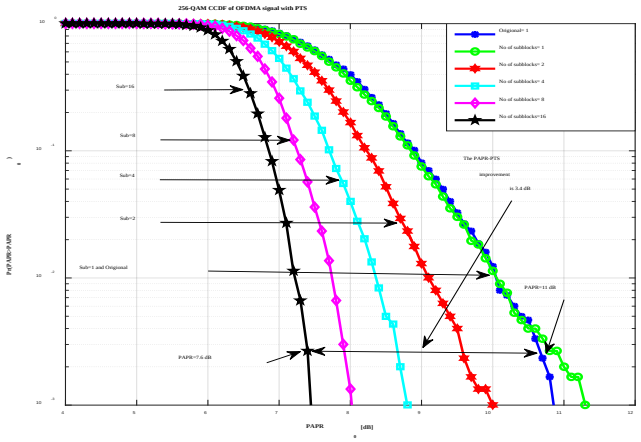

Figure 14: The relation of 256 QAM CCDF signal with PTS for different sub blocks points at M = 1, 2, 4, 8, and 16

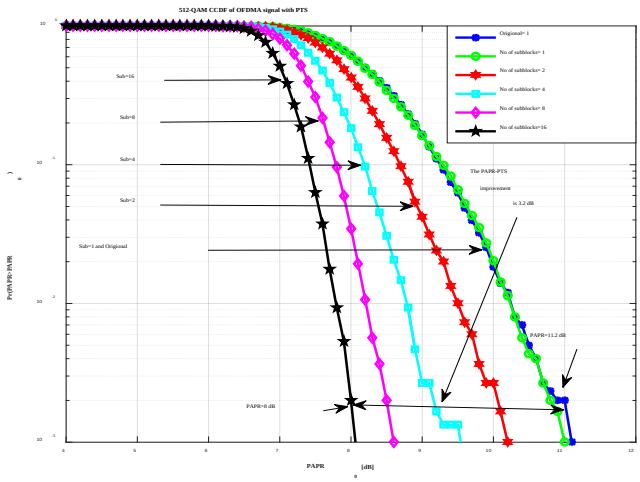

Figure 15: The relation of 512 QAM CCDF signal with PTS for different sub blocks points at M = 1, 2, 4, 8, and 16

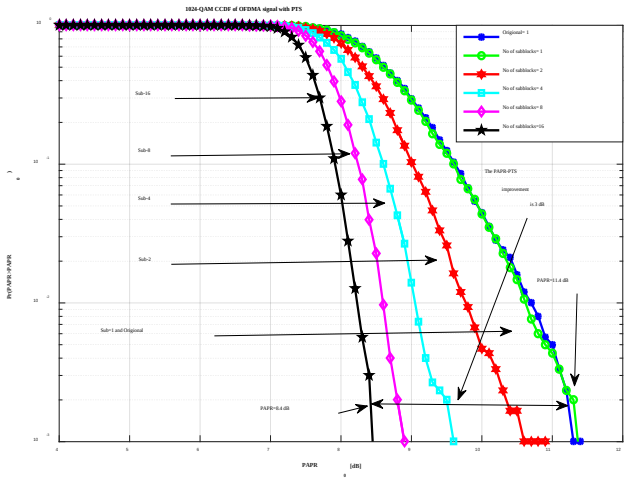

Figure 16: The relation of 1024 QAM CCDF signal with PTS for different sub blocks points at M = 1, 2, 4, 8, and 16

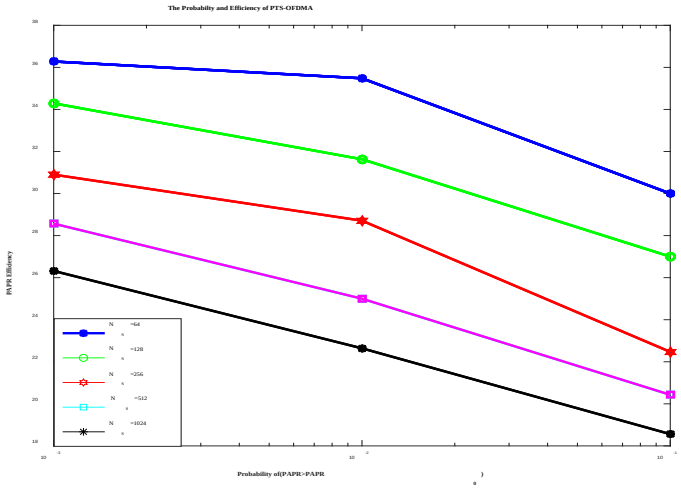

Figure 17: The relation between Efficiency and Probability of PTS-OFDMA

8 Discussions

The result of OFDMA when PAPR is considered shows that CCDF will increase when the number of chosen points Ns of OFDM are increased. To make a fast and more applicable OFDMA system, another MATLAB program is written to compute the PAPR reduction efficiency of the PTS algorithm. The OFDMA system performance by PAPR-PTS has clear improvement and strong ability. Also, the results show that probability of (PAPR > $PAPR_0$) is increased from 10^{-3} to 10^{-1}. Moreover, the theoretical and simulation CCDF of OFDMA signal become more accurate when the number of subblocks points are increased. FFT, M-QAM techniques are supported

Table 1: The PAPR efficiency and the PTS-PAPR values at *Ns* = 64 for different sub blocks points at *M* = 1, 2, 4, 8, and 16

Probability of PAPR > PAPR$_0$	**PAPR Value (dB) at *M* = 1**	**PAPR Value (dB) at *M* = 2**	**PAPR Value (dB) at *M* = 4**	**PAPR Value (dB)at *M* = 8**	**PAPR Value (dB) at *M* = 16**	**PAPR Efficiency Improving**
10^{-3}	10.2	9	8	7.1	6.5	36.28
10^{-2}	9.3	8.3	7.5	6.7	6	35.48
10^{-1}	8	7.2	6.7	6.1	5.6	30

Table 2: The PAPR efficiency and the PTS-PAPR values at *Ns* = 128 for different sub blocks points at *M* = 1, 2, 4, 8, and 16

Probability of PAPR > PAPR$_0$	**PAPR Value (dB) at *M* = 1**	**PAPR Value (dB) at *M* = 2**	**PAPR Value (dB) at *M* = 4**	**PAPR Value (dB)at *M* = 8**	**PAPR Value (dB) at *M* = 16**	**PAPR Efficiency Improving**
10^{-3}	10.5	9.6	8.9	7.9	6.9	34.29
10^{-2}	9.8	8.8	7.7	7.3	6.7	31.63
10^{-1}	8.5	7.8	7.2	6.7	6.2	27

Table 3: The PAPR efficiency and the PTS-PAPR values at *Ns* = 256 for different sub blocks points at *M* = 1, 2, 4, 8, and 16

Probability of PAPR > PAPR$_0$	**PAPR Value (dB) at *M* = 1**	**PAPR Value (dB) at *M* = 2**	**PAPR Value (dB) at *M* = 4**	**PAPR Value (dB)at *M* = 8**	**PAPR Value (dB) at *M* = 16**	**PAPR Efficiency Improving**
10^{-3}	11	10	8.9	8.2	7.6	30.9
10^{-2}	10.1	9.3	8.4	7.8	7.2	28.71
10^{-1}	8.9	8.3	7.7	7.3	6.9	22.47

Table 4: The PAPR efficiency and the PTS-PAPR values at *Ns* = 512 for different sub blocks points at *M* = 1, 2, 4, 8, and 16

Probability of PAPR > PAPR$_0$	**PAPR Value (dB) at *M* = 1**	**PAPR Value (dB) at *M* = 2**	**PAPR Value (dB) at *M* = 4**	**PAPR Value (dB)at *M* = 8**	**PAPR Value (dB) at *M* = 16**	**PAPR Efficiency Improving**
10^{-3}	11.2	10.1	9.4	8.4	8	28.57
10^{-2}	10.4	9.5	8.7	8.2	7.8	25
10^{-1}	9.3	8.7	8.2	7.8	7.4	20.43

Table 5: The PAPR efficiency and the PTS-PAPR values at *Ns* = 1024 for different sub blocks points at *M* =1, 2, 4, 8, and 16

Probability of PAPR > PAPR$_0$	**PAPR Value (dB) at *M* = 1**	**PAPR Value (dB) at *M* = 2**	**PAPR Value (dB) at *M* = 4**	**PAPR Value (dB)at *M* = 8**	**PAPR Value (dB) at *M* = 16**	**PAPR Efficiency Improving**
10^{-3}	11.4	10.4	9.65	8.8	8.4	26.32
10^{-2}	10.6	9.7	9.2	8.6	8.2	22.64
10^{-1}	9.7	9	8.6	8.3	7.9	18.56

the OFDMA signals to enhance the system capability and reliability. It is obvious that OFDMA system performance by PAPR-PTS has clear acheivement and strong capability by approximately 3 dB to 3.7 dB PAPR reduction efficiency at $M = 1$ and $M = 16$ sub-blocks partitions for various QAM subcarriers at Ns, 64, 128, 256, 512, and 1024 as are illustrated in Figure 12 to Figure 16. Also, the relationship between efficiency and probability of PTS-OFDMA is another great improvement that indicating two things decreasing the BER and increasing the system efficiency as is shown in Figure 17. For instance, at the probability of 10^{-3} and Ns = 1024 , the PAPR efficiency of PTS-OFDMA is improved by approximately 27% more than the original one. Truly, the system efficiency is increased while the Ns is increased at desired probability.

9 Conclusions

OFDMA which is consists of multiple OFDM symbols is a excellent technique that increases speed, range, reliability and spectral efficiency for wireless systems for each user. Also, OFDM is used FFT, M-QAM techniques which are supported the OFDMA signals to get high performance and throughout by the wireless networks space. The results using MATLAB proves that the ability of the OFDMA technique has capacity and validity. In fact, OFDMA technology can

bring wonderful results that continually boosts the mobile communications systems for the Fourth and Fifth Generations (4G and 5G) standards. Moreover, the OFDMA system will be more efficient and flexible when the PAPR-PTS reduction technique is used. Consequently, The PAPR reduction efficiency is improved by more than 3 dB for various *M*-QAM subcarriers in OFDMA system. Furthermore, the relationship between efficiency and probability of PTS-OFDMA is another great improvement that indicates two things decreasing the BER and increasing the system efficiency. It is clear that OFDMA with PAPR-PTS reduction technique can accommodate many users with widely varying applications, data rates, and quality of service requirements.

References

[1] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*: Pearson Education, 2007.

[2] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*: John Wiley & Sons, 2010.

[3] M. Doelz, E. Heald, and D. Martin, "Binary Data Transmission Techniques for Linear Systems," *Proceedings of the IRE*, vol. 45, pp. 656-661, 1957.

[4] H. Seung Hee and L. Jae Hong, "An Overview of Peak-to-Average Power Ratio Reduction Techniques for Multicarrier Transmission," *IEEE Wireless Communications*, vol. 12, pp. 56-65, 2005.

[5] G. Hill, "Peak Power Reduction in Orthogonal Frequency Division Multiplexing Transmitters," Victoria University, 2011.

[6] A. D. S. Jayalath, M. U. S. o. C. Science, and S. Engineering, *OFDM for Wireless Broadband Communications: (Peak Power Reduction, Spectrum and Coding)*: Monash University, 2002.

[7] T. Jiang and Y. Wu, "An Overview: Peak-to-Average Power Ratio Reduction Techniques for OFDM Signals," *IEEE Transactions on Broadcasting*, vol. 54, pp. 257-268, 2008.

[8] M. Morelli, C. C. J. Kuo, and M. O. Pun, "Synchronization Techniques for Orthogonal Frequency Division Multiple Access (OFDMA): A Tutorial Review," *Proceedings of the IEEE*, vol. 95, pp. 1394-1427, 2007.

[9] R. Morrison, L. Cimini, and S. K. Wilson, "On The Use of a Cyclic Extension in OFDM," in *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, 2001, pp. 664-668.

[10] S. H. Muller and J. B. Huber, "OFDM with Reduced Peak-to-Average Power Ratio by Optimum Combination of Partial Transmit Sequences," *Electronics Letters*, vol. 33, pp. 368-369, 1997.

[11] R. v. Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*: Artech House, Inc., 2000.

[12] A. Peled and A. Ruiz, "Frequency Domain Data Transmission Using Reduced Computational Complexity Algorithms," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'80.*, 1980, pp. 964-967.

[13] M. A. N. Shukur, K. Pahwa, and E. A. Singhal, "Implementing SC-FDMA & OFDMA in MATLAB," *International Journal of Computing & Corporate Research ISSN (Online)*.

[14] S. B. Weinstein, "The History of Orthogonal Frequency-Division Multiplexing [History of Communications]," *IEEE Communications Magazine*, vol. 47, 2009.

[15] Y. Wu and W. Y. Zou, "Orthogonal Frequency DivisionMultiplexing: A Multi-Carrier Modulation Scheme," *IEEE Transactions on Consumer Electronics*, vol. 41, pp. 392-399, 1995.

[16] H. Yin and S. Alamouti, "OFDMA: A Broadband Wireless Access Technology," in *Sarnoff Symposium, 2006 IEEE*, 2006, pp. 1-4.

[17] W. Y. Zou and Y. Wu, "COFDM: An Overview," *IEEE transactions on broadcasting*, vol. 41, pp. 1-8, 1995.

Majed Mohammad Albogame received the Bachelor's degree in Electrical and Electronics Engineering from King Saud University (KSU), Riyadh, Saudi Arabia. In 2013, he graduated with dual Master degrees in Electrical Engineering and Computer Science from the University of Bridgeport, Bridgeport, CT, USA. Currently, he is pursuing his PhD degree in Electrical Engineering at The University of Toledo, Toledo, OH, USA. His research interests include Communication Systems, Data Communications, Digital Signal Processing, Data Compression for Multimedia Applications, Mobile and Wireless Network, Information Theory and Modern Coding in Digital Technologies.

Junghwan Kim (M'88, SM'09) received the B.S. degree in Electronics Engineering from Seoul National University, Seoul, Korea, in 1975, and the M.S. and Ph.D. degrees in Electrical Engineering from Virginia Polytechnic Institute and State University, Blacksburg, VA, USA in 1985 and 1988 respectively. In September 1988, he joined the faculty of the University of Toledo, Ohio, USA where he is a Professor in the Electrical Engineering and Computer

Science Department and a director of digital communication research laboratory. His current research interests are modeling and performance analysis of satellite/mobile communications systems and networks under interferences and jamming, digital broadcasting, modulation and channel coding. He also has interests in the localization of mobile sensor network and physical-layer based encryption. He is a member of IEEE Communication and Broadcasting Societies. He serves as an Associate Editor of the IEEE Transactions on Broadcasting.

Mohammed Y. Niamat received the Bachelor's degree in electrical engineering from the Aligarh Muslim University, Aligarh, India, the Master's degree in electrical engineering from the University of Saskatchewan, Saskatchewan, Canada, and the Ph.D. degree from the University of Toledo, OH, USA, in 1989. During 1996–1997, he was a Visiting Associate Professor with the Center for Reliable Computing, Stanford University. He has supervised over 50 graduate students. He is currently the focus Group Leader for the High-Performance Computing Research Group of Electrical Engineering and Computer Science department at the University of Toledo. His current research interests include Hardware Security, Testing of Digital/VLSI Circuits, Field Programmable Gate Arrays (FPGAs), Security for Smart Grids, Quantum-dot Cellular Automata (QCA).

Applying Task Models from Human Computer Interaction to Support and Improve Usage Based Statistical Testing for Web Applications*

Gity Karami †
Southern Methodist University, Dallas, TX, 75275, USA

Jeff Tian‡
Southern Methodist University, Dallas, TX, 75275 USA
Northwest University, Xi'an, Shaanxi, CHINA

Abstract

Markov Operational Profile (Markov OP) is a usage model that helps us to perform usage based statistical testing (UBST) and to improve quality of web applications. After maintenance and evolution, the original Markov OP may not reflect actual usage of the updated application accurately. At this point, the updated web application is not under deployment to allow us to construct a new Markov OP. Activity diagrams from software development processes and task models from Human-Computer Interaction (HCI), which describe an application in term of activities or tasks, share some common characteristics with Markov OP. In our previous research, we developed a method to update the original Markov OP using the activity diagrams. In this paper, we develop a new method to update the original Markov OP using a particular variation of task models called concurrent task tree (CTT). We also quantify impact of this method on the accuracy of the updated Markov OP. We have applied this method in a case study to demonstrate its applicability and effectiveness.

Key Words: Task models, activity diagrams, human computer interaction (HCI), usage based statistical testing (UBST), and Markov operational profile (Markov OP).

1 Introduction

A task model is a logical description of an interactive task which is performed by target users through an user interface [1]. Task models are widely used in the human-computer interaction (HCI) domain. HCI is a field of study focusing on the interaction between humans and computers. In addition, task modeling is recognized as a fundamental technique to support user-centered design (UCD). UCD is a user interface design process to develop an interactive application with a high degree of usability [8]. Task models are also employed at different phases of user interface development (UID) and to assess usability of the interactive application [7].

Task models share some common characteristics with Markov OP, which is an usage model to quantify actual usage of software or a web application by target users [12, 18]. Markov OP can be used to capture actual usage of web applications to support usage based statistical testing for effective web quality and reliability assurance [14]. If the structure, functionality, environment, or usage of the web application is changed over maintenance, accuracy of the existing Markov OP may deteriorate. Such decreased accuracy may also negatively affect the effectiveness of testing and quality assurance activities [23]. At this time, the updated web application has not been deployed yet, so that actual usage of the web application could not be collected to construct a new Markov OP. On the other hand, task models which describe the web application in terms of tasks can be used to predict actual usage of the updated web application.

Task models are not usually applied in a software engineering field. In this paper, we develop a new method to apply task models to update the existing Markov OP to maintain its accuracy over maintenance and evolution. We quantify its impact on the accuracy of the updated Markov OP. We applied this method in a case study to demonstrate its applicability and effectiveness. An alternative method was developed recently to update the existing Markov OP using the activity diagrams which describe the recently updated applications in terms of activities [15]. In this paper, we also compare these two methods on the accuracy of their updated Markov OP.

2 Related Work

In this section, we review related work and discuss the problem of deteriorating accuracy of Markov OP over maintenance and evolution.

2.1 Task Models

Task models are interface models to convey the interface development process and to produce a user interface [1]. An interface model describes all aspects of a user interface based on some interface modeling languages. The task models usually

*This works is supported in part by National Science Foundation (NSF) Grant 1126747 and NSF Net-Centric I/UCRC.

†Department of Computer Science and Engineering. Email:gkarami@smu.edu

‡Department of Computer Science and Engineering - School of Informatics. Email:Tian@smu.edu

involve goals, actions, and domain objects. Goals specify when a desired state is met; sequences of actions identify procedures to achieve a goal; and domain objects describe mandatory and required elements of the user interface to complete each task. There are two types of task models: descriptive and prescriptive task models [20]. Descriptive task models describe how the tasks are performed currently. They are developed by domain experts and psychologists using interviews or observations. On the other hand, prescriptive task models describe how an updated application should support the tasks.

Task models are useful for many purposes. They play an important role in the human-computer interaction (HCI) field, because they represent the logical activities that should support users in reaching their goals. HCI is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. Task modeling is also a well-developed technique supporting user-centered user interface design (UCD). UCD is the process of designing an application with a high degree of usability from the perspective of how it will be understood and used by a user [8]. In addition, task models are also employed at various stages of user interface development (UID) [17]. UID is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience.

Task models can also be used to evaluate usability of an interactive system [2, 7]. They have been used to predict the user's performance in reaching their goals or to support analysis of user behavior to identify usability problems. Although task models are widely used in these areas, they are rarely used in software engineering fields [13].

There are several methods for task modeling, including HTA (hierarchical task analysis), GOMS (goals, operators, methods, selection rules), KLM (keystroke level model), UAN (user action notation), and CTT (concurrent task tree), as briefly described below:

- HTA is a widely used task modeling method where a high-level task is decomposed into a hierarchy of sub tasks [1, 9]. HTA involves decomposing goals into sub goals. The order and structure of these goals and sub goals is represented visually as a hierarchical graph structure or in a tabular textual format. HTA is a systematic method of describing how work is organized in order to meet the overall objective of the task. It involves identifying in a top down fashion the overall goal of the task, then the various sub-tasks and the conditions under which they should be carried out to achieve that goal. In this way, complex planning tasks can be represented as a hierarchy of operations, different things that people must do within a system and plans, the conditions which are necessary to undertake these operations.
- GOMS provides a hierarchical description to reach goals in terms of operators [3, 11]. GOMS describes a user's cognitive structure on four components, including a set of goals, a set of operators, a set of methods for achieving the goals, and a set of selection rules for choosing among competing methods for a specific goal. One limitation of GOMS approaches is that it considers error-free behavior and only sequential tasks. The latter limitation is partially overcome by critical path method GOMS (CPM-GOMS). However, operators for representing flexible temporal relationships are not provided in CPM-GOMS. When distributed applications are considered, the time requested by the application to respond to the user interactions is difficult to predict, because it can depend on unpredictable external factors.
- KLM predicts how long it will take an expert user to accomplish a routine task without errors using an interactive computer system [16, 20]. KLM consists of six operators: the first four are physical motor operators followed by one mental operator and one system response operator. 1) keystroke or button press, 2) pointing to a target on a display with a mouse, 3) homing the hands on the keyboard or other device, 4) manually drawing straight-line segments, and 5) mentally preparing for executing physical actions, and 6) response time of the system. To estimate execution of the task, we should add required times for keystrokes, pointing, homing, drawing, mental operators, and response time.
- UAN describes the dynamic behavior of graphical user interface [8], where the interface is represented as a quasi-hierarchical structure of asynchronous tasks. The sequencing within each task is independent of that in the others. It is a textual notation where the interface is represented as a quasi-hierarchical structure of asynchronous tasks, the sequencing within each task being independent of that in the others. The UAN notation is suitable for specifying tasks, which are the components of the specification. It also provides good support for specifying low level actions sequencing. One of the possible limitations is that it can lead to large specification sometimes with many details not always useful for the designer. Another limitation is that the order which has to be followed to interpret the tables of the basic tasks is rigid and inadequate.
- CTT is one of the most widely used notations for task modeling, specifically tailored for user interface model-based design [19, 22]. CTTs are based in a graphical notation that supports the hierarchical structured of tasks, which can be interrelated through a powerful set of operators that describe the temporal relationships between subtasks. CTT provides a set of operators to describe the temporal relationships among tasks, such as enabling, concurrency, disabling, interruption, and choice. Four types of tasks are supported in the CTT notation, including abstract tasks, interaction tasks, user tasks, and application tasks. CTT focus on activities, hierarchical structure, graphical syntax, concurrent notation, task allocation, and objects.

2.2 Markov OP and Its Accuracy

Markov OP is an effective usage model which reflects actual usage of large applications, such as web applications [12, 23]. Markov OP is an effective way for usage based statistical testing (UBST), reliability assessment, and usability evaluation. The constructed Markov OP can also be used for traditional coverage based testing (CBT) with no extra cost [14]. States, state transitions, and transition probabilities are basic elements of a Markov OP.

To construct a Markov OP, data should be collected from appropriate information sources. An access log kept in the web server reflects actual usage of a web application, so it is an appropriate and accurate information source. Requested URL and Referring URL fields of the access log can be used to identify all visited web pages and hyperlinks of the web application [12]. Markov OP can be constructed by assigning each web page or a group of web pages to a unique state in a Markov OP, assigning each hyperlink or a group of hyperlinks to a unique state transition, and calculating transition probabilities using these fields in the web access log.

Accuracy of the constructed Markov OP may deteriorate after maintenance activities. The maintenance activities may change the structure, functionality, and usage of the web application [4]. Since some components of the web application are added, removed, or modified after these activities, the actual usage of the updated web application may change. If environment or user population are changed over the software life cycle, the actual usage of the web application may also change. Therefore, the existing Markov OP constructed before the maintenance activities or environmental change would become less accurate for the updated web application. However, the updated web application is not under deployment at this time, so that actual usage of the updated web application could not be collected to construct a new Markov OP. On the other hand, existing information sources, such as activity diagrams commonly used in the normal software development process can be used to update the existing Markov OP.

Activity diagrams describe a system in terms of activities [5, 6]. The activities in the activity diagrams are shown as states that represent the execution of a set of operations. Activity diagrams share some common characteristics with Markov OP [15]. Activity diagrams consist of a set of activities and transitions; while Markov OP consists of states, state transitions, and transition probabilities to quantify user behavior.

Recently, a novel method was developed to maintain accuracy of the existing Markov OP using updates derived from the activity diagrams [15]. The existing Markov OP constructed before maintenance reflects the initial user behavior. Since the user behavior is not expected to change drastically, the existing Markov OP is used as a starting point in this method to predict actual usage of the updated web application. By comparing the initial Markov OP and the activity diagrams, the structure of the existing Markov OP is updated. The transition probabilities are updated using the partial data estimation method, the analogy estimation method, or the redistribution estimation method. If partial new data through partial deployment of the updated web application or its internal inspection, testing, and evaluation activities can be collected, the transition probabilities of the state transitions are updated using the partial data estimation method. If a state transition in an existing Markov OP which has similar structure or semantics exist, the analogy estimation method is used. Otherwise, the redistribution estimation method is employed.

However, if the activity diagrams are not available, task models can be used instead to maintain accuracy of the existing Markov OP. In addition, different information sources may lead to different levels of accuracy for the updated Markov OP. In the following, we develop a new method to update the existing Markov OP using task models.

3 New Method and Validation

In this section, we first discuss possible task models that can be used to update the existing Markov OP. Then, we develop a new method to maintain accuracy of Markov OP using updates derived from a specific type of task model called concurrent task tree (CTT). Finally, we quantify impact of this method on the accuracy of the updated Markov OP and compare it with the existing Markov OP as well as the updated Markov OP based on our the previous method using activity diagrams.

3.1 Selecting Specific Task Models

Task models share some common characteristics with a Markov OP. Each state in the Markov OP is assigned to a web page or a group of web pages. Corresponding tasks in the task model may be assigned to the same web page or group of web pages. In addition, each state transition in the existing Markov OP is assigned to a hyperlink or a group of hyperlinks. Corresponding transition in the task model may be assigned to the same hyperlink or group of hyperlinks. Therefore, there is a possibility to map a sequence of states in a Markov OP to a corresponding sequence of tasks in a task model.

As we described in Section 2, there are several methods for task modeling, including HTA (hierarchical task analysis), GOMS (goals, operators, methods, selection rules), KLM (keystroke level model), UAN (user action notation), and CTT (concurrent task tree). Lower level task models, such as KLM or keystroke level model- GOMS (KLM-GOMS) are not appropriate models to map the Markov OP. As stated earlier, each state of the Markov OP represents a page or a cluster of pages of the web application. Each of these pages may be related to several tasks in the KLM or KLM-GOMS and some pages may share some common tasks. Therefore, it may be difficult to map the Markov OP to a low level task model. However, there is a good chance to map a higher level task model, such as CTT, HTA, GOMS, and UAN to the Markov OP.

In this paper, we select CTT to maintain accuracy of the existing Markov OP, because CTT and Markov OP both support similar levels of information, details, and structure. In addition, they both support hierarchical structure, graphical syntax, and

concurrent notation. We plan to explore the application of HTA, UAN, and GOMS for similar purposes in the future.

3.2 New Method

Before discussing our new method to maintain accuracy of Markov OP using CTT, we introduce the following notations:

- $\langle S,T,P\rangle$: a given Markov OP. We also use $\langle S^e,T^e,P^e\rangle$, $\langle S^u,T^u,P^u\rangle$, and $\langle S^r,T^r,P^r\rangle$ to denote an existing Markov OP, an updated Markov OP, and a reference Markov OP respectively.
- $S=\{s_i|i=1,2,...,N\}$; s_i: a state.
- $T=\{t_{ij}|i,j=1,2,...,N\}$; t_{ij} : a transition from s_i to state s_j; n_i : the number of possible transitions from s_i.
- $P=\{p_{ij}|i,j=1,2,...,N\}$; p_{ij}: the transition probability from state s_i to state s_j.
- $|S|$: cardinality of set S; $|S|=N$

The structure of the updated Markov OP is first identified in our method. The existing Markov OP is mapped to a CTT to find out how maintenance and evolution affect actual usage of the updated web application. Each state in the existing Markov OP is represented as one web page or a group of web pages. A sub task or a group of sub tasks in the CTT is associated to one web page or a group of web pages. By mapping the existing Markov OP to the CTT, we can identify all expected states and state transitions in the updated Markov OP. The following are scenarios that may happen when we map the existing Markov OP to the CTT.

- Removing a state or a state transition: If designers consider a web component as an extra component, or users no longer need the web component to reach their goals, the web component should be removed from the web application over maintenance. In this scenario, CTT which is used in UID or support UCD will not include any task corresponding to this web component. If a web component is removed from the web application over maintenance, the web component is no longer accessible to target users. Therefore, we need to remove the corresponding states and state transitions from the existing Markov OP.
- Adding a state or a state transition: If designers find out a web component is missing in the web application, or users need the missing web component to reach their goals, the web component should be added to the updated web application. In this scenario, CTT will include a task or a group of tasks corresponding to the web component. If a web component is added to the web application over maintenance, the new web component will be accessible to the users. Therefore, we need to add corresponding states and state transitions to the updated Markov OP.
- Changing functions associating with a state: If designers decide to change the function of a web component in the web application, or if users need the web component with a different function to reach their goals, the function of the web component in the updated web application should be changed. In this scenario, the CTT includes a different task or a different group of tasks corresponding to the web component. However, even if the function associating to a web component is changed over maintenance, it may only affect the usage probability but not the structure of the updated Markov OP. Therefore, we do not need to change the structure to the updated Markov OP.

After determining the structure of the updated Markov OP in our method, the transition probabilities of the updated Markov OP are updated using one of the following three methods:

- Partial data estimation method: In UCD, part of the updated web application may be under partial deployment for usability testing to study how it will be understood and used by a user. New data can be collected through such activities and other iterative UID activities, such as expert reviews of conceptual and detailed designs, user surveys, etc. [21]. Transition probabilities are estimated from such new data. In the partial data estimation method, the estimated transition probabilities are substituted for those in the corresponding subset of the updated Markov OP.
- Analogy estimation method: If the updated web application is not under partial deployment and partial data estimation method can not be used to update the transition probabilities, an expert can apply the analogy estimation method. In this method, a subset of the updated Markov OP is substituted by a similar subset in the existing Markov OP by the expert. Expertise gained in usability evaluation through heuristic reviews or cognitive walkthrough can be employed to make such expert decisions.
- Redistribution estimation method: If partial data or analogy estimation methods can not be applied due to lack of partial data from UCD/UID activities or unavailability of matching analogies, the redistribution estimation method can be used. In this situation, transition probabilities of the updated Markov OP is proportionally redistributed after addition, deletion, or functional migration of certain states and/or transitions.

  If t^e_{ij} was removed in updating the existing Markov OP, transition probabilities to its siblings could be redistributed proportionally using the following equation:

$$p^u_{ik}= p^e_{ik} * (1+\frac{p^e_{ij}}{1-p^e_{ij}}),\ 1\le k\le n'_i$$

  If t^e_{ij} was added in updating the existing Markov OP, transition probability p^u_{ij} for the added state transition should be estimated first. Then, probabilities to its siblings are redistributed proportionally using the following equation:

$$p^u_{ik}= p^e_{ik} * (1-p^u_{ij}),\ 1\le k\le n'_i,\ k\ne j$$

3.3 Evaluation and Validation

Prior to validating our new method, we need to construct a new Markov OP for the updated web application based on actual measurements after its deployment. We assume the new Markov OP is 100% accurate and use it as a reference Markov OP. By comparing the existing Markov OP with the reference Markov OP, we can quantify inaccuracy level of the existing Markov OP. When we compare the existing Markov OP with the reference Markov OP, we may find:

- Subset of accurate states in the existing Markov OP S_1^e: includes common states which are present in both the reference Markov OP and the existing Markov OP.
- Subset of inaccurate states in the existing Markov OP S_0^e: includes missing states, extra states, and incorrect states. Missing states are absent in the existing Markov OP, while they are present in the reference Markov OP. Extra states are present in the existing Markov OP, while they are absent in the reference Markov OP. Incorrect states in the existing Markov OP support different functionality rather than their corresponding states in the reference Markov OP.

To quantify the inaccuracy level of the existing Markov OP, we need to calculate state inaccuracy of the existing Markov OP σ^e using the following equation:

$$\sigma^e = \frac{|S_0^e|}{|S^r \bigcup S^e|}$$

Let $S^r \bigcup S^e = S$, we then have $\sigma^e = \frac{|S_0^e|}{|S|}$.

We also need to calculate probability inaccuracy of the existing Markov OP. We first compare the probability of every transition in the existing Markov OP p_{ij}^e with the probability of corresponding transition in the reference Markov OP p_{ij}^r and calculate their absolute difference $|p_{ij}^e - p_{ij}^r|$. If a corresponding transition is absent, we treat its transition probability as 0 in this calculation. Then, we can calculate probability inaccuracy of the existing Markov OP ε^e using the following equation:

$$\varepsilon^e = \frac{\sum_{ij} |p_{ij}^e - p_{ij}^r|}{K}$$

Where $K = |P^e \bigcup P^r|$, i.e. the size of the union of transitions actually in the models.

Similarly, we can quantify inaccuracy level for the updated Markov OP estimated by our new method by comparing the updated Markov OP with the reference Markov OP using the same method above. To validate our new method, we need to demonstrate that the updated Markov OP is more accurate than the existing Markov OP constructed before maintenance. Therefore, the following is the validation question for our new method:

> VQ: *The updated Markov OP should be more accurate than the existing Markov OP.*

To answer the validation question quantitatively, we need to demonstrate that state inaccuracy of the updated Markov OP estimated by our new method σ^u is lower than state inaccuracy of the existing Markov OP σ^e. We also need to demonstrate that probability inaccuracy of the updated Markov OP estimated by our new method ε^u is lower than probability inaccuracy of the existing Markov OP ε^e. Therefore, the following expressions should be true to validate our method:

$$\sigma^u < \sigma^e \text{ and } \varepsilon^u < \varepsilon^e$$

In addition, we quantitatively compare impact of this new method with our previous method for inaccuracy of the updated Markov OP. Our previous method updates the existing Markov OP using activity diagrams [15]. We first compare the state inaccuracy of the updated Markov OP estimated by this new method σ^u with the state inaccuracy of the updated Markov OP estimated by our previous method σ^a to determine which method leads us to lower state inaccuracy. In addition, we compare the probability inaccuracy of the updated Markov OP estimated by this new method ε^u with the probability inaccuracy of the updated Markov OP estimated by our previous method ε^a to determine which method leads us to lower probability inaccuracy.

4 Case Study

In the previous section, we introduced a new method to update the existing Markov OP using CTT. We applied this method on a student payments (SP) website to provide an initial validation of this new method. We also compared these results with our previous case study on the same website while updating the existing Markov OP using updates derived from activity diagrams [15].

4.1 Data and Baseline

We collected the access logs of the SP website from two consecutive months in 2015, one before and one after some maintenance activities. The number of hits in the access logs of the pre-maintenance SP website was 1484 and post-maintenance SP website was 1722. Before applying the methods, we constructed the existing Markov OP for the SP website. We first used "Requested URL" field in the access logs to extract all visited web pages. We assigned one visited web page or a group of visited web pages to a unique state in the existing Markov OP. Then, we used "Requested URL" and "Referring URL" fields to extract all incoming navigation links for each web page. We associated one navigation link or a group of navigation links to a unique state transition in the existing Markov OP. Finally, we calculated the transition probability of each state transition using these fields. Figure 1 shows the existing Markov OP for SP.

In our previous research, we updated the existing Markov OP using activity diagrams [15]. We validated that the updated Markov OP has a higher level of accuracy than the existing Markov OP. Figure 2 shows the updated Markov OP for the updated SP website derived from the activity diagram.

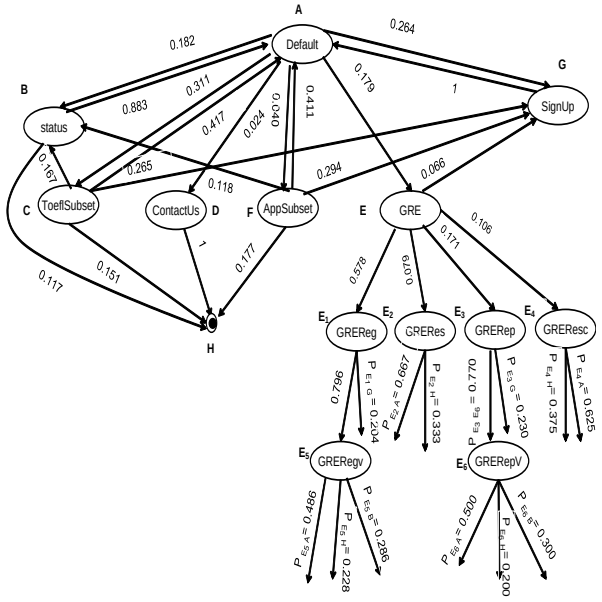

Figure 1: Existing Markov OP for SP

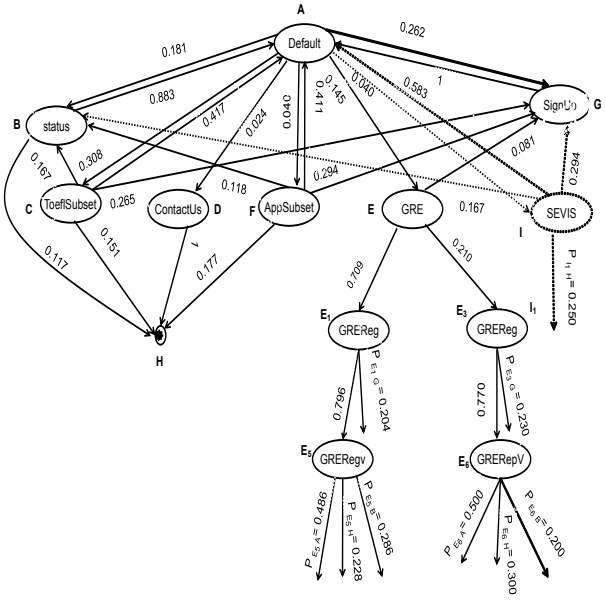

Figure 2: Updated Markov OP derived from the activity diagram for SP

In this paper, we applied our new method to update the existing Markov OP using CTT. After updating the existing Markov OP using our new Method, we compare the updated Markov OP with the existing Markov OP to validate our new method. In addition, the updated Markov OP using activity diagrams above will be used as an additional baseline for comparison. We compare these two updated Markov OPs using different methods to determine which method leads us to a more accurate Markov OP.

4.2 Updating the Structure

To update the existing Markov OP, we first updated its structure. We compared the existing Markov OP with the CTT to find out how maintenance and evolution lead to changes in actual usage of the web application, so that we could update the structure of the existing Markov OP. Figure 3 shows the CTT for the updated SP website. Figure 4 shows the updated Markov OP for the updated SP website estimated using the CTT.

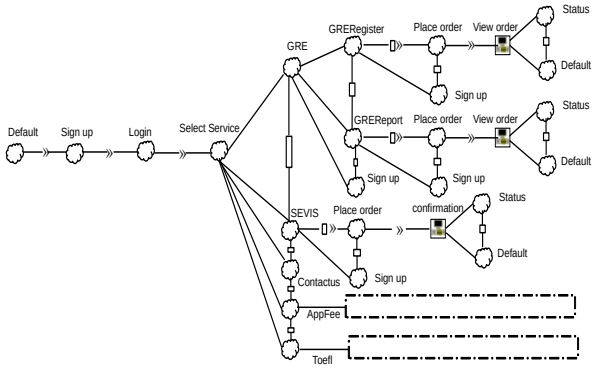

Figure 3: CTT for SP

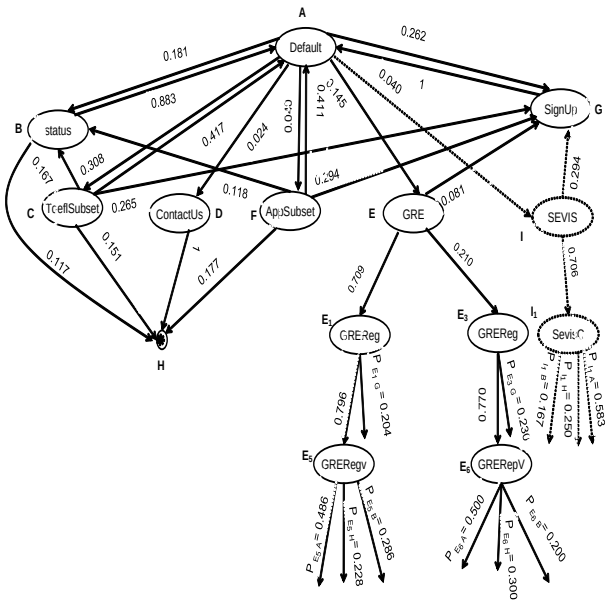

Figure 4: Updated Markov OP derived from the CTT for SP

When we used CTT to update the structure of the existing Markov OP, we added the "SEVIS" state and its incoming and outgoing state transitions to the existing Markov OP, because there is no state in the existing Markov OP corresponding to the "SEVIS" sub task in the task model, and all of the incoming and outgoing state transitions to and from "SEVIS" are missing in the existing Markov OP. Furthermore, we added the "SEVISC" state and its incoming and outgoing state transitions, because we do not have any state in the existing Markov OP corresponding to the "SEVISC" sub task in the task model, and all of the

incoming and outgoing state transitions to and from "SEVISC" are also missing in the existing Markov OP.

In addition, we removed "GREResc" and "GRERes" states, and all their incoming and outgoing state transitions from the existing Markov OP, as "GRERes" state and "GREResc" state in the existing Markov OP can not be mapped to any sub task in the task model. The corresponding tasks for "GREResc" and "GRERes" are no longer supported in the new version of SP.

4.3 Updating the Transition Probabilities

After updating the structure of the existing Markov OP using the CTT, we updated transition probabilities of the existing Markov OP. Since state transitions from "GRE" to "GRERes" and from "GRE" to "GREResc" are removed in Figure 4, we updated the probabilities of the sibling transitions which started from the "GRE" state in the updated Markov OP. As no new data was available at that time, and no analogy could be established, we applied the proportional redistribution estimation method to estimate the transition probabilities. This change was also propagated up to its parent.

In Figure 4, The "SEVIS" state and the "SEVISC" state and their incoming or outgoing state transitions are not present in the existing Markov OP. We applied the analogy estimation method to estimate probability for the incoming transition to the "SEVIS" state and outgoing transition from the "SEVISC" state. We utilized "AppSubset" and its transition probabilities in the existing Markov OP to estimate these probabilities. We also applied the partial data estimation method to estimate probability from the "SEVIS" state to the "SEVISC" state.

For all the remaining state transitions in Figure 4, the transition probabilities remained the same as their corresponding parts in the existing Markov OP. However, if new data become available for some of these state transitions, the corresponding transition probabilities can be updated accordingly by using our partial data estimation method.

4.4 Validation

Before validating our new method, we constructed a new Markov OP based on actual usage of the updated web application by target users using the access log recorded after its deployment. Figure 5 shows the new Markov OP constructed based on actual usage of SP after maintenance. We assumed the new Markov OP is 100% accurate and used it as our reference Markov OP.

To validate our new method, we quantified inaccuracy levels of the existing Markov OP and the updated Markov OP using CTT by comparing them with the reference Markov OP. When we compared the existing Markov OP with the reference Markov OP, we identified accurate and inaccurate states as following:

$$S_1^e = \{A,B,C,D,E,F,G,H,E_1,E_3,E_5,E_6\}\ ,\ S_0^e = \{I,I_1,E_2,E_4\}$$

We calculated state inaccuracy of the existing Markov

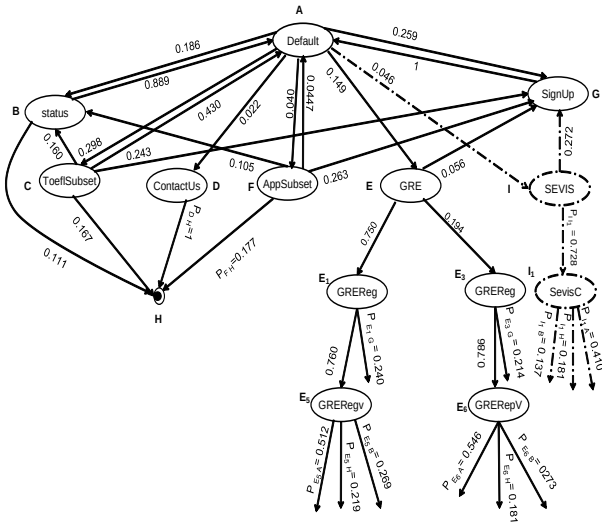

Figure 5: Reference Markov OP for SP

OP σ^e using the following equation:

$$\sigma^e = \frac{|S_0^e|}{|S^r \bigcup S_0^e|} = \frac{|S_0^e|}{|S|} = \frac{4}{12} = 0.333$$

We also calculated probability inaccuracy of the existing Markov OP. We compared the probability of every transition in the existing Markov OP p_{ij}^e with the probability of corresponding transition in the reference Markov OP p_{ij}^r and calculate their absolute difference $|p_{ij}^e - p_{ij}^r|$. Table 1 shows transition probabilities in the existing Markov OP and the reference Markov OP and their differences. We calculated probability inaccuracy of the existing Markov OP ε^e using the following equation:

$$\varepsilon^e = \frac{\sum_{ij} |p_{ij}^e - p_{ij}^r|}{K} = \frac{5.293}{43} = 0.123$$

Similarly we compared the updated Markov OP using CTT with the reference Markov OP, we found accurate and inaccurate states as following:

$$S_1^u = \{A,B,C,D,E,F,G,H,E_1,E_3,E_5,E_6,I,I_1\},\ S_0^u = \emptyset$$

We calculated state inaccuracy of the updated Markov OP using CTT σ^u using the following equations:

$$\sigma^u = \frac{|S_0^u|}{|S^r \bigcup S_0^u|} = \frac{|S_0^u|}{|S|} = \frac{0}{14} = 0$$

We also calculated probability inaccuracy of the updated Markov OP using CTT. Table 2 shows transition probabilities in the updated Markov OP using CTT and the reference Markov OP and their differences. Then, we calculated transition probability accuracy ε^u using the following equation:

Table 1: Probability inaccuracy of the existing Markov OP

	p^e_{ij}	p^r_{ij}	$\lvert p^e_{ij} - p^r_{ij}\rvert$
p_{AB}	0.182	0.186	0.004
p_{AC}	0.311	0.298	0.013
p_{AD}	0.024	0.022	0.002
p_{AE}	0.179	0.149	0.030
p_{AF}	0.040	0.040	0
p_{AG}	0.264	0.259	0.005
p_{AI}	0	0.046	0.046
p_{BA}	0.883	0.889	0.006
p_{BH}	0.117	0.111	0.006
p_{CA}	0.417	0.430	0.013
p_{CB}	0.167	0.160	0.007
p_{CG}	0.265	0.243	0.022
p_{CH}	0.151	0.167	0.016
p_{DH}	1	1	0
p_{GA}	1	1	0
p_{FA}	0.411	0.474	0.063
p_{FB}	0.118	0.105	0.013
p_{FG}	0.294	0.263	0.031
p_{FH}	0.177	0.158	0.019
p_{EG}	0.066	0.056	0.01
p_{EE_1}	0.578	0.750	0.172
p_{EE_2}	0.171	0	0.171
p_{EE_3}	0.079	0.194	0.115
p_{EE_4}	0.106	0	0.106
p_{E_1G}	0.204	0.240	0.036
$p_{E_1E_5}$	0.796	0.760	0.036
p_{E_2A}	0.667	0	0.667
p_{E_2H}	0.333	0	0.333
p_{E_3G}	0.230	0.214	0.016
$p_{E_3E_6}$	0.770	0.786	0.016
p_{E_4A}	0.625	0	0.625
p_{E_4H}	0.375	0	0.375
p_{E_5A}	0.486	0.512	0.026
p_{E_5B}	0.228	0.269	0.041
p_{E_5H}	0.286	0.219	0.067
p_{E_6A}	0.500	0.546	0.046
p_{E_6B}	0.300	0.273	0.027
p_{E_6H}	0.200	0.181	0.019
p_{IG}	0	0.272	0.272
p_{II_1}	0	0.728	0.728
p_{I_1A}	0	0.410	0.410
p_{I_1B}	0	0.137	0.137
p_{I_1H}	0	0.181	0.181
$\sum_{i,j} \lvert p^e_{ij} - p^r_{ij}\rvert$			5.293
ε^e			0.123

$$\varepsilon^u = \frac{\sum_{ij} |p^u_{ij} - p^r_{ij}|}{K} = \frac{0.954}{36} = 0.026$$

By analyzing the results, we found $\sigma^e = 0.333$, $\sigma^u = 0$, $\varepsilon^e = 0.123$, $\varepsilon^u = 0.026$; $\sigma^u < \sigma^e$ and $\varepsilon^u < \varepsilon^e$

Therefore, our VQ is validated for our new method, i.e, *the updated Markov OP is more accurate than the existing Markov OP.*

Table 2: Probability inaccuracy of the updated Markov OP using CTT

	p^u_{ij}	p^r_{ij}	$\lvert p^u_{ij} - p^r_{ij}\rvert$
p_{AB}	0.181	0.186	0.005
p_{AC}	0.308	0.298	0.010
p_{AD}	0.024	0.022	0.002
p_{AE}	0.145	0.149	0.004
p_{AF}	0.040	0.040	0
p_{AG}	0.262	0.259	0.003
p_{AI}	0.040	0.046	0.006
p_{BA}	0.883	0.889	0.006
p_{BH}	0.117	0.111	0.006
p_{CA}	0.417	0.430	0.013
p_{CB}	0.167	0.160	0.007
p_{CG}	0.265	0.243	0.022
p_{CH}	0.151	0.167	0.016
p_{DH}	1	1	0
p_{GA}	1	1	0
p_{FA}	0.411	0.474	0.063
p_{FB}	0.118	0.105	0.013
p_{FG}	0.294	0.263	0.031
p_{FH}	0.177	0.158	0.019
p_{EG}	0.081	0.056	0.025
p_{EE_1}	0.709	0.750	0.041
p_{EE_3}	0.210	0.194	0.016
p_{E_1G}	0.204	0.240	0.036
$p_{E_1E_5}$	0.796	0.760	0.036
p_{E_3G}	0.230	0.214	0.016
$p_{E_3E_6}$	0.770	0.786	0.016
p_{E_5A}	0.486	0.512	0.026
p_{E_5B}	0.228	0.269	0.041
p_{E_5H}	0.286	0.219	0.067
p_{E_6A}	0.500	0.546	0.046
p_{E_6B}	0.300	0.273	0.027
p_{E_6H}	0.200	0.181	0.019
p_{IG}	0.294	0.272	0.022
p_{II_1}	0.706	0.728	0.022
p_{I_1A}	0.583	0.410	0.173
p_{I_1B}	0.167	0.137	0.03
p_{I_1H}	0.250	0.181	0.069
$\sum_{i,j} \lvert p^e_{ij} - p^r_{ij}\rvert$			0.954
ε^u			0.026

4.5 Comparing Methods

After validating our method, we quantitatively compare the accuracy of the updated Markov OP using CTT and the updated Markov OP using the activity diagram. When we compared the updated Markov OP using the activity diagram with the reference Markov OP, we found accurate and inaccurate states

as following:

$$S_1^a = \{A,B,C,D,E,F,G,H,E_1,E_3,E_5,E_6,I,I_1\}, S_0^a = \{I_1\}$$

We calculated state inaccuracy of the updated Markov OP using the activity diagram σ^a using the following equations:

$$\sigma^a = \frac{|S_0^a|}{|S^r \bigcup S_0^a|} = \frac{|S_0^a|}{|S|} = \frac{1}{14} = 0.071$$

We also calculated probability inaccuracy of the updated Markov OP using the activity diagram. Table 3 shows transition probabilities in the updated Markov OP using the activity diagram and the reference Markov OP and their differences. Then, we calculated transition probability accuracy ε^u using the following equation:

$$\varepsilon^a = \frac{\sum_{ij} |p_{ij}^a - p_{ij}^r|}{K} = \frac{3.116}{40} = 0.079$$

By analyzing the results, we observed that the updated Markov OP using activity diagram and the updated Markov OP using CTT are both more accurate than the existing Markov OP. We also found that the updated Markov OP using CTT has a higher level of accuracy ($\sigma^u = 0$ and $\varepsilon^u = 0.026$) than the updated Markov OP using activity diagram ($\sigma^e = 0.333$ and $\varepsilon^e = 0.123$). Therefore, in this case study, CTT leads us to maintain the accuracy of the existing Markov OP better than the activity diagram. However, a more comprehensive empirical study is needed to determine the relative accuracy of the Markov OPs updated using these two different methods.

5 Discussion

After maintenance and evolution, the existing Markov OP will become less accurate. At this point, the updated web application is under testing, but it has not been deployed yet. Therefore, we are not able to collect actual usage of the updated web application to construct a more accurate Markov OP to support UBST at the moment. On the other hand, task models are widely used in different areas, including HCI, UCD, UID, and usability evaluation. Task models are rarely used in software engineering fields. In this research, we developed a new method to apply task models to maintain the accuracy of the existing Markov OP.

Our method to maintain accuracy of Markov OP is practical, as the existing Markov OP and CTT are used with little extra cost. We could predict usage of the updated web application based on the existing Markov OP as a starting point, as user behavior is not expected to change drastically after updating the web application. In our method, we update the structure of the existing Markov OP by comparing it with CTT. Then, we update the transition probabilities of the existing Markov OP using a partial data estimation method using new data for UCD/UID activities, analogy estimation method using expertise gained from those activities, or redistribution estimation method using consistent formulas.

Table 3: Probability inaccuracy of the updated Markov OP using the activity diagram

	p_{ij}^u	p_{ij}^r	$\lvert p_{ij}^u - p_{ij}^r \rvert$
p_{AB}	0.181	0.186	0.005
p_{AC}	0.308	0.298	0.010
p_{AD}	0.024	0.022	0.002
p_{AE}	0.145	0.149	0.004
p_{AF}	0.040	0.040	0
p_{AG}	0.262	0.259	0.003
p_{AI}	0.040	0.046	0.006
p_{BA}	0.883	0.889	0.006
p_{BH}	0.117	0.111	0.006
p_{CA}	0.417	0.430	0.013
p_{CB}	0.167	0.160	0.007
p_{CG}	0.265	0.243	0.022
p_{CH}	0.151	0.167	0.016
p_{DH}	1	1	0
p_{GA}	1	1	0
p_{FA}	0.411	0.474	0.063
p_{FB}	0.118	0.105	0.013
p_{FG}	0.294	0.263	0.031
p_{FH}	0.177	0.158	0.019
p_{EG}	0.081	0.056	0.025
p_{EE_1}	0.709	0.750	0.041
p_{EE_3}	0.210	0.194	0.016
p_{E_1G}	0.204	0.240	0.036
$p_{E_1E_5}$	0.796	0.760	0.036
p_{E_3G}	0.230	0.214	0.016
$p_{E_3E_6}$	0.770	0.786	0.016
p_{E_5A}	0.486	0.512	0.026
p_{E_5B}	0.228	0.269	0.041
p_{E_5H}	0.286	0.219	0.067
p_{E_6A}	0.500	0.546	0.046
p_{E_6B}	0.300	0.273	0.027
p_{E_6H}	0.200	0.181	0.019
p_{IG}	0.294	0.272	0.022
p_{II_1}	0	0.728	0.728
p_{I_1A}	0	0.410	0.410
p_{I_1B}	0	0.137	0.137
p_{I_1H}	0	0.181	0.181
p_{IA}	0.583	0	0.583
p_{IB}	0.167	0	0.167
p_{IH}	0.250	0	0.250
$\sum_{i,j} \lvert p_{ij}^e - p_{ij}^r \rvert$			3.116
ε^a			0.079

We applied our new method in a case study to provide an initial validation of its applicability and effectiveness. We quantified that the state inaccuracy of the existing Markov OP is $\sigma^e = 0.333$; while the state inaccuracy of the updated Markov OP using CTT is $\sigma^u = 0$. In addition, we quantified that the probability inaccuracy of the existing Markov OP is $\varepsilon^e = 0.123$; while the probability inaccuracy of the updated Markov OP using CTT is $\varepsilon^u = 0.026$. Therefore, our case study

demonstrated that the updated Markov OP using CTT has a higher level of accuracy than the existing Markov OP.

If CTT is not available to update the existing Markov OP, we may utilize alternative information sources, such as activity diagrams, or other types of task models. In our previous research, we utilized activity diagrams to maintain the accuracy of the existing Markov OP [15]. We also quantitatively compared the impact of both methods for the accuracy of the updated Markov OP. We quantified the state inaccuracy of the updated Markov OP using activity diagrams is $\sigma^a = 0.071$. In addition, we quantified the probability inaccuracy of the updated Markov OP using activity diagrams is $\varepsilon^a = 0.079$. Therefore, our case study demonstrated that the updated Markov OP using CTT has a higher level of accuracy with $\sigma^u = 0$ and $\varepsilon^u = 0.026$ than the updated Markov OP using the activity diagram. However, a more comprehensive empirical study is needed to determine the relative accuracy of the Markov OPs updated using these two different methods.

There are several methods for task modeling, including HTA (hierarchical task analysis), GOMS (goals, operators, methods, selection rules), KLM (keystroke level model), UAN (user action notation), and CTT (concurrent task tree). We plan to apply other task models to update the existing Markov OP. There is also a possibility to apply a combination of task models. Lower level task models, such as KLM or keystroke level model- GOMS (KLM-GOMS) can be used when we want to predict actual usage of very low level components in the updated web application; while higher level task model, such as CTT, HTA, GOMS, and UAN can be used to predict actual usage of the other components.

Our method to maintain accuracy of Markov OP may yield many benefits. If we utilize the updated Markov OP using our method to test the updated web application, it will have multiple impacts with little extra cost, including better test coverage, higher test efficiency, and better quality assurance. We examined the impact of accurate usage models on reliability, test coverage, and test efficiency. We found supporting evidence that accurate Markov OP improves reliability, test coverage, and test efficiency [14].

Our method has some limitations, including scalability and accuracy of the task models. There is also a possibility that we can not map the task models to the existing Markov OP. We plan to address the difficult issue of mapping such complex CTT to the Markov OPs. We also plan to apply our method in some large scale case studies, such as the telecom web site used in [10], and to further validate our method in such realistic industrial settings. Future comprehensive studies are also planned to further validate our method and compare it to alternative methods.

6 Conclusion

Markov OP is a type of usage model that can support usage based statistical testing (UBST) and help ensure and maximize quality of web applications. We can construct Markov OP based on actual usage of the web application. However, accuracy of Markov OP could deteriorate over maintenance and evolution. The updated web application is not under deployment yet, so we can not collect actual usage data of the updated web application to construct a more accurate Markov OP for UBST at this point. However, we can utilize existing information sources, like task models, to update the existing Markov OP.

Task models, which describe the web application in terms of tasks, share common characteristics with Markov OP. Task models are usually used outside software engineering, including human-computer interaction (HCI), user-centered design (UCD), user interface development (UID), and usability evaluation. In this paper, we developed a novel method to apply existing task models to maintain accuracy of Markov OP over maintenance and evolution. We utilized a particular type of task models called concurrent task tree (CTT) to update the existing Markov OP. We applied our new method in a case study to provide an initial validation of its applicability and effectiveness. We quantified the state and probability inaccuracy for the existing Markov OP and the updated Markov OP using our new method. By analyzing the results we found that the updated Markov OP using CTT has a higher level of accuracy than the existing Markov OP. In previous research, we updated the existing Markov OP based on updates derived from the activity diagram. In this paper, we quantitatively compare impact of both methods for the accuracy of the updated Markov OP. By analyzing the results we found that the updated Markov OP using CTT has a higher level of accuracy than the updated Markov OP using the activity diagram in this case study. To summarize, the main contributions of this paper are:

1. Applying task models from human computer interaction to solve important software engineering problems.
2. A novel method to maintain the accuracy of Markov OP for web applications.
3. A practical way to update the existing Markov OP using existing task models when activity diagrams are not available.
4. A case study to provide an initial validation of the applicability and effectiveness of our method.
5. Quantitatively comparing impact of this method and our previous method for the accuracy of the updated Markov OP.

Our method has some limitations, including availability and scalability of CTT, and the difficult issue of mapping complex CTT to the Markov OPs. To overcome some of these limitations, we plan to apply other types of task models to update the existing Markov OP when CTT is not available or difficult to use. In addition, we plan to apply our method in some large scale case studies as well as more comprehensive empirical studies to further validate our method and compare it to alternative methods. After overcoming these limitations, our research will have a significant positive impact on the web application reliability and user experience.

References

[1] John Annett. "Hierarchical Task Analysis". In *Handbook of Human Factors and Ergonomics Methods*. CRC Press, 2006.

[2] Matthew L. Bolton, Radu I. Siminiceanu, and Ellen J. Bass. "A Systematic Approach to Model Checking Human Automation Interaction Using Task Analytic Models". *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, 41(5):961-976, August 2011.

[3] Georgios Christou1, Frank E. Ritter, and Robert J. K. Jacob. "Codein: A New Notation for GOMS to Handle Evaluations of Reality-Based Interaction Style Interfaces". *IJHCI*, 28(3):1-13, March 2012.

[4] Sergio Cozzetti B. de Souza, Nicolas Anquetil, and Kthia M. de Oliveira. "A Study of the Documentation Essential to Software Maintenance". In *Proc. of the 23rd Annual International Conference on Design of Communication*, pp. 68-75, 2005.

[5] Rik Eshuis. "Symbolic Model Checking of UML Activity Diagrams". *IEEE Trans. on Software Engineering*, 15(1):1-38, 2006.

[6] Xin Fan, Jian Shu, LinLan Liu, and QiJun Liang. "Test Case Generation from UML Subactivity and Activity Diagram". *Second International Symposium on Electronic Commerce and Security*, 2:244-248, 2009.

[7] Ruili Geng and Jeff Tian. "Improving Web Navigation Usability by Comparing Actual and Anticipated Usage". *IEEE Trans. on Human-Machine Systems*, 45(1):84-94, October 2015.

[8] Rex Hartson and Pardha Pyla. "The UX Book: Process and Guidelines for Ensuring a Quality User Experience". Morgan Kaufmann, 2012.

[9] Erik Hollnagel. *"Handbook of Cognitive Task Design"*. CRC Press, 2003.

[10] Wafa Jaffal and Jeff Tian. "Practical Risk-Based Technique to Improve Reliability for Incremental Web Application Development". In *Proc. of 27th Int. Conf. on Computer Applications in Industry and Engineering (CAINE-2014)*, October 2014.

[11] Bonnie E. John and David E. Kieras. "Using GOMS for User Interface Design and Evaluation: Which Technique?". *ACM Trans. on Computer-Human Interaction*, 3(4):287-319, 1996.

[12] Chaitanya Kallepalli and Jeff Tian. "Measuring and Modeling Usage and Reliability for Statistical Web Testing". *IEEE Trans. on Software Engineering*, 27(11):1023-1036, november 2001.

[13] Gity Karami and Jeff Tian. "Using Task Models to Maintain Accuracy of Web Usage Models". In *30th International Conference on Computer Applications in Industry and Engineering (CAINE 2017)*, October 2017.

[14] Gity Karami and Jeff Tian. "Improving Web Application Reliability and Testing Using Accurate Usage Models". *In: Lee R.(eds) Software Engineering Research, Management, and application (SERA 2017), Studies in Computational Intelligence (SCI)*, 75-92, June 2018.

[15] Gity Karami and Jeff Tian. "Maintaining Accurate Web Usage Models Using Updates from Activity Diagrams". *Information and Software Technology (IST)*, 96:68-77, April 2018.

[16] Christoph Lth and Burkhart Wolff. "Functional Design and Implementation of Graphical User Interfaces for Theorem Provers". *Journal of Functional Programming*, 9(2):1023-1036, March 1999.

[17] Daniel D. McCracken. *"User-Centered Website Development: A Human-Computer Interaction Approach"*. Prentice Hall, 2003.

[18] John D. Musa. *"Software Reliability Engineering"*. McGraw-Hill, 1998.

[19] Fabio Paterno. *"Model-Based Design and Evaluation of Interactive Applications"*. Springer Verlag, 1999.

[20] Frank E. Ritter, Gordon D. Baxter, and Elizabeth F. Churchill. "Foundations for Designing User-Centered Systems: What System Designers Need to Know about People". Springer Publishing Company, 2014.

[21] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobsand Niklas Elmqvist, and Nicholas Diakopoulos". *"Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 2016.

[22] Jos L Silva, Jos Creissac Campos, and Ana C R Paiva. "Model-based User Interface Testing with Spec Explorer and ConcurTaskTrees". *Electronic Notes in Theoretical Computer Science*, 208(14):77-93, April 2008.

[23] James A. Whittaker and Michael G. Thomason. "A Markov Chain Model for Statistical Software Testing". *IEEE Trans. on Software Engineering*, 20(10):812-824, October 1994.

Gity Karami is a postdoc fellow and adjunct faculty at Lyle School of Engineering, Southern Methodist University. She received her Ph.D. degree in Computer Science from Southern Methodist University in 2017. She received her first M.Sc. degree in Information Technology from Shiraz University and her second M.Sc. degree in Software Engineering from Southern Methodist University in 2008 and 2016 respectively. Dr. Karami has extensive research experience in software engineering, with a primary focus on assuring and improving the quality, dependability, reliability, usability, and safety of various large software systems. her research currently focuses on big data and machine learning techniques.

Jeff (Jianhui) Tian received a B.S. degree in Electrical Engineering from Xi'an Jiaotong University in 1982, an M.S. degree in Engineering Science from Harvard University in 1986, and a Ph.D. degree in Computer Science from the University of Maryland in 1992. He worked for the IBM Software Solutions Toronto Laboratory between 1992 and 1995 as a software quality and process analyst. Since 1995, he has been with Southern Methodist University, Dallas, Texas, now as Professor of Computer Science and Engineering. From 2012 to 2017, he was also a Shaanxi 100 Professor in the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. Since 2018, he has also been a visiting professor in the School of Informatics, Northwest University, Xi'an, China. His current research interests include software quality, reliability, usability, testing, measurement, and applications in commercial, net-centric, web-based, service and cloud computing software and systems. He is a member of IEEE and ACM.

An Empirical Algorithm for Turn Detection from Vehicle Data[1]

Jennifer Knull[*], Steven Beauchemin[*], and Michael Bauer[*]
The University of Western Ontario, London, Ontario, N6A5B7, CANADA

Abstract

Advanced Driver Assistance Systems have been shown to play an important role in accident prevention. Future driver assistance systems will not only rely on information about the vehicle and environment, but also on the state of the driver. This requires the development of individual *driver models* which can help predict how a person may behave in certain situations. A key in building such models is the ability to detect and analyze common driving maneuvers, such as turns. Algorithms are needed which can detect and characterize individual driving maneuvers. In this paper, we introduce an algorithm for detecting turns from vehicle instrument data and GPS coordinates. We evaluate the algorithm on data collected from 16 drivers of an instrumented vehicle. The drivers followed (approximately) the same route and the result was a data set involving 278 turns. The method achieves an accuracy of 97%. The algorithm can be used for post processing of driving data as well as real-time analysis during driving.

Key Words: Driving assistance systems, vehicular data, driver models.

1 Introduction

Driving is a risky activity but one that is also essential to modern society. According to the National Highway Traffic Safety Administration (NHTSA), there were 29,989 fatal motor vehicle accidents in the United States alone in 2014. Increasingly, automobile manufacturers and researchers are looking at technologies that can help reduce the burden of driving on humans and can improve safety. The forefront of this research today includes autonomous vehicles, which take full control of the vehicle and decision-making, and Advanced Driver Assistance Systems, or ADAS, which can notify the driver of potential dangers and may even perform emergency maneuvers in dangerous situations. ADASs have been shown to play a crucial role in accident prevention and in providing driver feedback. They continue to be a focus of research in finding new approaches to augment the driving experience with the end goal of providing the driver with the best support possible.

Future intelligent ADASs will rely on information about the state of drivers, including their patterns of behavior, driving habits, alertness, etc. in varying situations as well as information about the vehicle and the environment. Understanding the state of drivers or their behaviors requires the development of methods that enable the creation of *driver models.* Such models will be elements in an intelligent ADAS that can be used to predict how a person may react in certain situations, determine if the individual is not performing normally (ill or tired), is not reacting adequately (distracted), etc. Models of drivers will ultimately need to be built dynamically based on specific individual behaviors and characteristics, as no two people drive and react identically. Such models need to characterize how individuals perform various driving maneuvers, how they react in different circumstances, how easily they are distracted, how different driving conditions affect their behavior, and so on. Central in building such models is the ability to detect and analyze common driving maneuvers, such as making turns or changing lanes, on an individual-by-individual basis. Thus, there is a need for algorithms that can detect and characterize individual driving maneuvers.

In this contribution, we present an algorithm for detecting turns from vehicular data. The algorithms can be used for post processing of driving data, (as is the case here), as well as real-time analysis during driving. Our overall aim is to build automated methods that can use such algorithms to eventually determine characteristics of individual drivers during turns in order to build models of drivers.

This contribution is organized as follows: Section 2 provides an overview of related work on driver models and driver behavior. Section 3 describes the data that we have collected for use in this study. Section 4 presents the algorithm for turn detection followed by an analysis of the effectiveness of the algorithms in Section 5. Section 6 provides a conclusion and directions for future work.

2 Related Work

There has been a wide variety of previous research that can be broadly called “driver modeling”. However, central to the discussion of driving models is the notion of a driving maneuver. In driving, a *maneuver* is a collection of actions or series of actions that a person engages in controlling a vehicle for some specific task. These actions may include such physical

[1]This work was supported by the Natural Science and Engineering Research Council of Canada.
[*]Department of Computer Science. Email: {Jknull, beau, bauer}@uwo.ca.

activity, e.g. turning a steering wheel, or involve visual activity, such as checking a mirror. Driving maneuvers can be defined based on traffic and road infrastructure and can include "following", "turning at intersection", "changing lanes", "reacting to an obstacle", etc. These maneuvers can be differentiated by situational factors, such as the type of road, speed limit, number of lanes, the existence of other vehicles, pedestrians, traffic signs, or traffic lights ahead of the vehicle.

Driving models can be broadly grouped into four categories [25]: 1) vehicle-centric, a focus on vehicle performance and operations, 2) driver-centric, a focus on driving style or driver state/condition, 3) combined driver and vehicle, and 4) traffic-environment. We concentrate on works that focus on the driver or on a driver and vehicle combination.

The research in this area falls into two broad subareas [25]: a) understanding the driver and (the individual) driver behavior (our focus) and b) path and speed planning, and optimized driver/driving behavior. The first is related to modeling how a person drives a vehicle, how they execute maneuvers - this is the focus of our work and we shall restrict our review of related work to this area.

As noted, our specific interest is in modeling driver behavior: namely, how the driver interacts with the surrounding environment through gaze and the driver's use of vehicle controls, such as the brake and accelerator pedals, turn signals, and steering wheel. We are interested in building driver models from data about the vehicle and driver collected from sensors and actuators within the vehicle as well as data about the environment, such as objects (signs, vehicles), road conditions, for instance. Modern vehicles are already equipped with cameras and sensors to provide vehicular information. Radar systems for detecting distance, LIDAR systems [20] for obstacle detection, visual systems [28] for detecting road objects [1], and vehicle navigation systems, such as GPS [19], have been used in ADAS. Driver behavior is typically modeled by observing the manner in which a driver executes certain driving tasks, such as turning, lane changing, or overtaking a vehicle.

Chandler, et al. [5] implemented a vehicle-following model and used data from experiments with real vehicles to set the model's parameters. Ioannou [10] implemented an Automatic Cruise Control (ACC) system and compared it to three different human driver models. The results showed that the ACC system was able to provide safer driving. A hybrid, three-layered ACC system based on fuzzy logic and neural networks has also been implemented by Germann et al. [7]. Sandberg and Wahde [26] and Sanpeng et al. [27] explore lane keeping using steering behavior. Munigety and Mathew [23] reviewed lane-centric driving models, such as lane changing and ramp merging behaviors, in the context of mixed traffic conditions characterized by the presence of different vehicle types and weak lane discipline. Liu, et al. [18] focused on recognizing turns, lane changes and U-turns using gyroscopic data from a smartphone mounted in a vehicle. Other researchers have also studied lane changes [15] and turning at intersections [4].

Hallac, et al. [8] used CAN-bus data, such as speed, pedal pressure, and steering wheel angle, to develop a recognition system to identify individual drivers by their characteristics within a single turn. A number of methods have also been implemented for assessing the driver's sleepiness. Most of them utilize some features from the driver's face through video cameras and use eye and eyelid movements [3, 11, 24, 27], head movements [3, 11, 12] or some combination to infer sleepiness.

Other researchers have focused specifically on cognitive driver models which are based on human psychological behaviors. These models pay attention to behavioral and attentional features that a driver shows in performing different maneuvers. The psychological factors that can be incorporated into cognitive driver behavior modeling include distraction, reaction time, body strength (stamina, strength of muscles, etc.), vision, impairment (stress, fatigue, alcohol) and so on [9]. Metari et al. [22] analyzed the cephalo-ocular behavior of drivers in vehicle and road events, such as overtaking and crossing an intersection. This work was specifically concerned with finding the relationship between vision behavior of older drivers and their actions in road events. They posited that the cephalo-ocular information can be related to the driving maneuvers a driver decides to make. Other researchers have also tested the impact of eye movements on control of actions in everyday events, such as driving [16, 17].

Details of driving maneuvers have been considered essential in the development of driver models. Much of this work, however, has relied on the researcher to first identify such maneuvers, often manually, which are then used in subsequent analyses. To be able to infer models of individual drivers, driving maneuvers must first be determined *in situ* and then used to build the model. A first step is to be able to identify such maneuvers from the data available from the vehicle. This is the central aspect of the work reported here.

3 Dataset

The data was obtained from the RoadLAB in-vehicle laboratory (see Figure 1). Vehicular data was received from the on-board diagnostic system (OBD-II) using the CAN-bus protocol. Details of the vehicular instrumentation is provided by Beauchemin et al. [2]. Vehicular information and GPS data were collected at a sampling rate of 30Hz. Every frame contains current contextual information about the vehicle, geographical position, stereo sequences, and ocular data. The turn detection approaches we employ only use sensory vehicle information and geographic data to extract turns.

The raw data was collected from 16 different drivers between the ages of 20 and 47. All drivers followed the same route during data acquisition (see Figure 2). The route is generally located in the north end and downtown area of London, Ontario, starting and ending in the parking lot behind Middlesex College at Western University. It contains a total of 18 turns, 10 right turns and 8 left turns.

The turn detection approaches process the vehicular data received from the OBD-II. There are the occasional missing sensor readings from the different vehicle components which are recorded as zero (not simply missing), and, hence, remain undetected as actual values can also be zero. The turn detection algorithm treats each value equally, despite whether it may be a

Figure 1: The RoadLAB instrumented vehicle

Figure 2: Aerial view of driving route in London, Ontario (courtesy of google maps)

missing value. Additionally, some frames were affected by noise, with entries making abrupt changes to absurdly large, or even impossible, values which exceeded the normal ranges. When detected, a linear interpolation step was used to correct such values when they were detected in the raw data.

GPS is also known for its signal interference. While this is sporadic, this contingency is dealt for specifically by the turn detection algorithms, which are explained further in Section 4.

The dataset provided by RoadLAB lacks breadth. While it gives an ample amount of detail (just under 100,000 data frames at 30 frames per second for each driver), it only provides data for one route and 16 drivers. For our particular analysis, we reduce the number of frames by selecting every other one, providing sequences where the time between a pair of frames is $\frac{1}{15}$ second. The turn detection approaches ensure that they do not over-fit the results and remain as general as possible.

Lastly, the drivers were supervised as they participated in the study. One operative directed the driver on where to travel, following a predetermined route, while a second operative monitored the equipment. The driver listened to the first operative and turned where they were told. Occasionally, the driver turned on the wrong street or the operative gave them incorrect directions, yielding incorrect turns. For this reason, the route itself cannot be used to validate the turns found by the algorithm. The video sequences were employed instead. The drivers also knew they were being studied. As a result, the data may potentially be biased in some undetermined way. This however does not affect the turn detection algorithm.

4 Turn Detection Method

Each video in the dataset was manually analyzed and every turn along the driving path was recorded according to this definition to form a sequence of true turns. Ideally, this would be the same for all drivers but some deviated from our preplanned route, resulting in slightly different turn sequences. Furthermore, any true turn contained in a time interval where there was missing sensor information was excluded from the sequence.

The goal of a turn detection algorithm is to detect true turns. Any reported turn that is not an actual turn is a false positive and any true turn not reported is a false negative. We ignore turns in parking lots (departure and arrival points in the route) and only consider turns constrained to a road network.

We are concerned with empirical turn detection algorithms that do not require any map or information on specific road networks. The turn detection approach relies only on the vehicle data; the algorithm has no *a priori* knowledge of the driving route. Thus, the method can also be used to detect turns in real-time as the vehicle drives.

The CAN-bus system records steering wheel angle measurements. Steering wheel angle values will be one of the more dominant factors in determining a turn. We will see that this gives reasonable experimental results, but does rely on inductive assumptions, specifically on the necessity and the sufficiency of steering wheel use in order to complete a turn. We assume that turns occur at road crossings at near right angles

with at most two lanes in each direction, such that the steering wheel must be turned a significant amount in order to complete the turn.

The method we describe involves the use of GPS data and vehicular data, which we will refer to as the *Position-Based Method (PBM)* to detect turns. This method is an extension and refinement of the algorithm introduced in [14].

4.1 Concept for Position-Based Method

As a vehicle travels along the Earth, its position can be uniquely represented as a latitude and longitude coordinates using GPS. This coordinate maps to only one location, making the position of any vehicle identifiable and traceable. The idea behind the method is to use latitude, φ, and longitude, λ, to identify a change from one road to another. Since GPS can pinpoint any location, it is assumed that we can analyze the coordinates that represent these locations to detect turns. The RoadLAB dataset provides the latitude and longitude in signed degrees format of up to (and including) four decimal places. This degree of precision can identify individual streets and land parcels according to the qualitative scale and provides enough accuracy for turn identification.

Geographic latitude and longitude can be mapped to a Cartesian coordinate system in which φ represents a numerical value along the *y*-axis and λ represents a value along the *x*-axis. Turns are detected using the coordinate system by specific fluctuations in $\Delta\varphi$ and $\Delta\lambda$.

As a vehicle travels along a path, its position vector can be calculated. This vector signifies the direction of travel using two (λ, φ) endpoints. The algorithm takes two position vectors, $\vec{u}$ and $\vec{v}$, when there is a change in endpoints, and if the angle between $\vec{u}$ and $\vec{v}$ is greater or equal to 90 degrees then there is evidence of a turn. Given the endpoints of the two position vectors representing a vehicle along a road, namely, $\vec{u} = ((\lambda_1,\varphi_1)(\lambda_2,\varphi_2))$ and $\vec{v} = ((\lambda_2,\varphi_2),(\lambda_3,\varphi_3))$, we estimate the slopes as:

$$s_1 = \frac{(\varphi_2 - \varphi_1)}{(\gamma_2 - \gamma_1)}, \text{ and } s_2 = \frac{(\varphi_3 - \varphi_2)}{(\gamma_3 - \gamma_2)}$$

The slopes, s_1 and s_2, are used to describe how much of a change there is in direction between one position vector and another. Instead of calculating the angle between $\vec{u}$ and $\vec{v}$, we can determine if there is a shift between quadrants in the coordinate system. Each quadrant is separated by 90 degrees, so if s_1 and s_2 define two adjacent quadrants, then a turn has occurred. There are four cases between s_1 and s_2 when it comes to determining quadrants:

1. s_1 and s_2 are both positive.
2. s_1 and s_2 are both negative.
3. s_1 is positive and s_2 is negative.
4. s_1 is negative and s_2 is positive.

Cases 1 and 2 suggest that the slopes are not in adjacent quadrants; cases 3 and 4 suggest that the slopes are in adjacent quadrants, in which case a turn is detected. Thus, if $s_1 \times s_2$ is negative then this suggests a turn has occurred, otherwise it suggests that no turn has occurred.

If there is no change in direction, there is a potential division by zero. However, since we are only interested in the sign of $s_1 \times s_2$ and not the actual values, we can compute two measures for classification as follows:

$$m_1 = (\varphi_2 - \varphi_1)(\gamma_2 - \gamma_1) = \Delta_{12},$$

and $$m_2 = (\varphi_3 - \varphi_2)(\gamma_3 - \gamma_2) = \Delta_{23}.$$

This narrows the classification to two cases:

1. $\boldsymbol{m_1 \times m_2}$ is negative, or
2. $\boldsymbol{m_1 \times m_2}$ is positive.

We have been assuming that turn maneuvers take place at 90 degrees or more, but it is possible for a turn to occur at less than 90 degrees and remain in the same quadrant. To account for this possibility we rotate the positional vectors by 45 degrees and calculate new m_1 and m_2 using the rotated values. Note that for this work we have used 45 degrees, but the rotation can be more or less, as this choice of value determines how turns less than 90 degrees are treated.

We are now able to identify additional turns based on the quadrants. However, originally identified turns may not be detectable after the rotation. Thus we make use of two sets of calculations, as described in the algorithms in the next Section – one set from our original calculations and one set from the calculations after rotations.

4.2 Algorithms

The turn detection method consists of three stages, amalgamated to produce the final set of turns for each driver. The three stages are:

1. Turn Detection (with and without rotation).
2. Merge Turns.
3. Remove Duplicates.

The basic turn detection algorithm, *detectTurns*, is presented in Table 1. The algorithm for detecting rotated turns, *detectRotatedTurns*, is essentially identical with two changes: rotated coordinates are computed to obtain m_1 and m_2 in lines 9 and 12 of the *detectTurns* algorithm. The algorithms use several thresholds to account for noise and to adjust their sensitivity:

1. θ_{noise}: a noise threshold. Geographic coordinates provide a sound basis for detecting turns. However, GPS is susceptible to noise from the signals received from the satellite network, which can cause inaccuracies in the data. Signal interference can be accounted for by comparing the values to a noise threshold. This threshold is used to avoid any false positives that may cause a turn to be detected when there is missing or noisy data. The latitude and longitude values were examined and

normally they do not increase or decrease by more than 0.0001. Thus, this threshold was set to 0.0002 to ensure that the vehicle is travelling incrementally when a change is detected.

2. θ_{bend}: a threshold for a turn. Before a possible turn is accepted (line 13 in Table 1), the absolute average steering wheel position between the start frame and end frame of the potential turn is computed and checked against this threshold. This is needed because the latitude and longitude coordinates are precise enough to also detect (some, but not all) bends in the road. The resulting list of detected turns may contain false positives. Wheel position is already used to classify a left or right turn, but it can also be used to differentiate a turn from a bend. Conceptually, the average wheel position will be close to 0 for a bend, since the steering wheel will generally remain straight. The average wheel position for a turn will deviate much further from 0 because there is more variation in wheel position. A threshold for turn angle is set to 40[2] for *detectTurns* and 100 for *detectRotatedTurns*. These thresholds were established experimentally.

Table 1: Algorithm for detecting turns from a frame sequence

Algorithm: *detectTurns(frameSeq)*

Input: *frameSeq: sequence of frames*
Output: S_f *: set of turns {<direction, sFrame, eFrame, sLat, eLat, sLong, eLong>}*

1. Get first frame f_1, latitude, φ_1, and longitude, λ_1.
2. *change* ← *False*
3. **for** *f* **in** *frameSeq* **do**
4. Get first frame f_1, latitude, φ_1, and longitude, λ_1.
5. $\Delta\varphi \leftarrow (\varphi_2 - \varphi_1),\ \Delta\lambda \leftarrow (\lambda_2 - \lambda_1)$
6. **if** $(\Delta\varphi \neq 0$ **or** $\Delta\lambda \neq 0)$ **and** $\Delta\varphi < \theta_{noise}$ **and** $\Delta\lambda < \theta_{noise}$ **then**
7. **if not** *change* **then**
8. *change* ← *True*
9. $m_1 \leftarrow \Delta\varphi\Delta\lambda,\ f_{start} \leftarrow f,\ \varphi_1 \leftarrow \varphi_2,\ \lambda_1 \leftarrow \lambda_2$
10. **else**
11. $f_{end} \leftarrow f$
12. $m_2 \leftarrow \Delta\varphi\Delta\lambda$
13. **if** $m_1 \times m_2 \leq 0$ **and** $wheelTurn(f_{start}, f_{end}) < \theta_{bend}$ **then**
14. $direction \leftarrow wheelDirection(f_{start}, f_{end})$
15. $turn \leftarrow <direction, f_{start}, f_{end}, \varphi_1, \lambda_1, \varphi_2, \lambda_2>$
16. $S_f.append(turn)$
17. $change \leftarrow False$
18. **else**
19. $f_{start} \leftarrow f_{end},\ m_1 \leftarrow m_2,\ \varphi_1 \leftarrow \varphi_2,\ \lambda_1 \leftarrow \lambda_2$
20. **else**
21. $\varphi_1 \leftarrow \varphi_2,\ \lambda_1 \leftarrow \lambda_2$

[2] This is in terms of "steering wheel units" as reported by data from the CAN-bus.

The function *wheelTurn()* computes the average steering wheel position over the frames specified by starting and ending frames. The function *wheelDirection()* determines the direction of the turn: if the average is negative, then the steering wheel position was, on average, to the left of the center position which signifies a left turn while if the average was positive, then, on average, the steering wheel was to the right of the center and indicates a right turn.

If $\boldsymbol{m_1} \times \boldsymbol{m_2} > 0$ then the program does not detect a turn. In this case, the vehicle continues along a straight path. However, this could also indicate the start of a new turn. The program does not ignore the possibility of a new turn, so it sets m_1 to m_2 and the variable *change* remains *true*. The start of an identified turn is defined as the first change in latitude or longitude values. The end of the turn is defined as the second change in latitude or longitude values.

The basic turn detection algorithms, *detectTurns* and *detectRotatedTurns*, both detect turns but will detect the same turns and possibly different turns (or possibly false positives). The results, therefore, need to be combined and duplicates removed. This algorithm, *consolidateTurns*, is presented in Table 2. The merging part of the algorithm (lines 1-2) combines the turns from both sets and sorts them by frame numbers (note that any identical turns, defined by identical tuples, can be removed in this step). The result is a list of turns ordered by occurrence.

Table 2: Algorithm for merging turn sequences and removing duplicates

Algorithm: $consolidateTurns(frameSeq_1, frameSeq_2)$

Input: $frameSeq_1, frameSeq_2$*: sequence of frames*
Output: S_f *: set of turns*

 # A turn is defined by a tuple: <direction, sFrame, eFrame, sLat, eLat, sLong, eLong>
 # Merge
1. $S_{temp} \leftarrow frameSeq_1 \cup frameSeq_2$
2. $S' = sortByFrame(S_{temp})$

 # Remove duplicates
3. $S_f \leftarrow \emptyset$
4. $turn_1 \leftarrow S'.getNext()$
5. $turn_2 \leftarrow S'.getNext()$
6. $\Delta start \leftarrow abs(turn_2.sFrame_2 - turn_1.sFrame_1),\ \Delta end \leftarrow abs\ (turn_2.eFrame_2 - turn_1.eFrame_1)$
7. **if** $!(\Delta start < \theta_{thresh})$ **or** $!(\Delta end < \theta_{thresh})$ **then**
8. $S_f.append(turn_1)$
9. $S_f.append(turn_2)$
10. **else**
11. $S_f.append(turn_2)$
12. **for** *turn* **in** *S'* **do**
13. $turn_1 \leftarrow turn_2$
14. $turn_2 \leftarrow turn$
15. $\Delta start \leftarrow abs(turn_2.sFrame_2 - turn_1.sFrame_1),\ \Delta end \leftarrow abs\ (turn_2.eFrame_2 - turn_1.eFrame_1)$
16. **if** $!(\Delta start < \theta_{thresh})$ **or** $!(\Delta end < \theta_{thresh})$ **then**
17. $S_f.append(turn_2)$

The removal of duplicates in *consolidateTurns* (Table 2) extracts distinct turn events from the combined set of turns. It first tests whether the first two turns are the same turn or two separate turns. The start frame and end frame of both turns are evaluated to determine the closeness between the turns. A frame threshold, θ_{threh}, measures this closeness. If the difference between start frames and end frames is less than this threshold, then both turns occur within a certain amount of time and so are considered to be the same turn otherwise they are considered to be two separate turn events. Based on experiments, a frame difference of 170 (about 11.3 seconds), was determined to provide the best results.

If the first two turn events represent different turns, then both turn events are added to S_f. If they are duplicates of the same turn, then the second turn is added to S_f. We assume that there is no advantage to choosing one turn over the other. The algorithm proceeds to read the rest of the turn events from *S'* one turn at a time processing them in pairs in a similar fashion. Since the turns in *S'* are sorted by occurrence, it is suitable to read and compare turns in sequential pairs. If there are duplicate turns, then they will occur one after the other in *S'* and only one will be added to S_f.

5 Experimental Results

The algorithms presented in the previous Section were then used to analyze the turn events for the 16 drivers the RoadLAB data set contains. Some turns where there was missing or very noisy data were excluded. The results are presented in Table 3 for each driver. *Actual* indicates the actual number of good turns (i.e., not noisy or missing) for each driver. *Detected*, *FP and FN* indicate the number of turns detected by the algorithm, the number of False Positives and number of False Negatives, respectively, for each driver. Totals are provided in the last column.

The accuracy of the algorithm is then measured by:

$$Accuracy = \frac{(n - FP - FN)}{N}$$

where *Accuracy* is the computed accuracy, *n* the number of actual turns, *FP* is the number of false positives, and *FN* the number of false negatives across all drivers. The overall accuracy for the algorithm based on this measure was 97.8%.

There is the question whether turns that were missed due to missing data should be included in the measurements or not. From one perspective, these missing turns could indicate that the turn detection algorithm should be further refined in order to try to accommodate the possibility of missing turn data. However, it is arguable that it is not the limitations of the algorithm but of the data collection process itself. In either case, Table 4 summarizes the results with missed turns due to missing data included. The accuracy of the algorithm then becomes 95.77%.

6 Conclusions and Future Directions

A turn detection algorithm based on GPS coordinates and vehicle has been presented along with experimental evaluation of the approach. The algorithm performed well, achieving a 97% accuracy of detection, and even a 95% accuracy overall when missing turns were considered.

As noted several times throughout the paper, the analysis of the accuracy of the algorithm is based on a small number of drivers and turns on a single driving route. The algorithm needs to be tested against a larger set of turn data with greater variation. The algorithm also made use of a number of thresholds. While the values of these thresholds were determined from the analysis of the data, they do need to be refined on more turn data and drivers. We are also very interested in finding how the algorithm, suitably adapted, operates in a real-time mode to detect turns but again this will require further studies.

While turns are common and important driving maneuvers, there are other maneuvers that also need to be detected and characterized. In developing models of drivers, it will be necessary to analyze these and other maneuvers in order to characterize the driving behavior of different drivers. This might be achieved by developing parameterized models for each driver and for each type of maneuver. It might also be possible to develop broader classes of driving behavior associated with each type of maneuver and to characterize the classes. Then it would only be necessary to determine which class a driver belonged to for each maneuver. We are currently examining this in the context of turns.

Table 3: Results of the turn detection algorithm

Driver	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	***Total***
Actual	**18**	**18**	**18**	**18**	**16**	**16**	**17**	**18**	**17**	**18**	**18**	**18**	**16**	**16**	**18**	**18**	**278**
Detected	**18**	**18**	**20**	**18**	**16**	**16**	**17**	**18**	**17**	**17**	**18**	**18**	**15**	**16**	**18**	**18**	**278**
FPs	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**3**
FNs	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**2**	**0**	**0**	**0**	**3**

Table 4: Results of the turn detection algorithm with missing turns included

Driver	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	***Total***
Actual	**18**	**18**	**18**	**18**	**18**	**17**	**17**	**18**	**18**	**18**	**18**	**18**	**18**	**16**	**18**	**18**	**284**
Detected	**18**	**18**	**20**	**18**	**16**	**16**	**17**	**18**	**17**	**17**	**18**	**18**	**15**	**16**	**18**	**18**	**278**
FPs	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**3**
FNs	**0**	**0**	**0**	**0**	**2**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**4**	**0**	**0**	**0**	**9**

References

[1] R. Abou-Jaoude, "ACC Radar Sensor Technology, Test Requirements, and Test Solutions", *IEEE Trans. on Intelligent Transportation Systems*, 4(3):115-122, 2003.

[2] S. Beauchemin, M. Bauer, T. Kowsari, and J. Cho, "Portable and Scalable Vision-Based Vehicular Instrumentation of the Analysis of Driver Intentionality", *IEEE Transactions on Instrumentation and Measurement*, 61(2):391-401, 2012.

[3] L. Bergasa, J. Nuevo, M. Sotelo, R. Barea, and M. Lopez, "Real-Time System for Monitoring Driver Vigilance", *IEEE Trans. on Intelligent Transportation Systems*, 7(1):63-77, 2006.

[4] H. Berndt and K. Dietmayer, "Driver Intention Inference with Vehicle Onboard Sensors", IEEE Int. Conf. on Vehicular Electronics and Safety (ICVE), pp. 102-107, 2009.

[5] R. Chandler, R. Herman, and E. Montroll, "Traffic Dynamics: Studies in Car Following", *Operations Research*, 6(2):165-184, 1958.

[6] S. Cheng and M. Trivedi, "Turn-Intent Analysis using Body Pose for Intelligent Driver Assistance", *IEEE Pervasive Computing*, 5(4):28-37, 2006.

[7] S. Germann and R. Isermann, "Nonlinear Distance and Cruise Control for Passenger Cars", *IEEE Proceedings of the American Control Conference*, pp. 3081-3085, 1995.

[8] D. Hallac, A. Sharang, R. Stahlmann, A. Lamprecht, M. Huber, Martin Roehder, R. Sosic, and J. Leskovec, "Driver Identification using Automobile Sensor Data from a Single Turn", *IEEE Proceedings of the International Conference on Intelligent Transportation Systems (ITSC)*, pp. 953-958, 2016.

[9] S. Hamdar, *Driver Behavior Modeling*, Springer, pp 537-558, 2012.

[10] P. Ioannou and C.-C. Chien, "Autonomous Intelligent Cruise Control", *IEEE Transactions on Vehicular Technology*, 42(4):657-672, 1993.

[11] Q. Ji, Z. Zhu, and P. Lan, "Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue" *IEEE Trans. on Vehicular Technology*, 53(4):1052-1068, 2004.

[12] S. Jin, S-Y Park, and J-J Lee, "Driver Fatigue Detection using a Genetic Algorithm", *Artificial Life and Robotics*, 11(1):87-90, 2007.

[13] D. King-Hele, "Erasmus Darwin's Improved Design for Steering Carriages and Cars", *Notes and Records of the Royal Society of London*, 56(1):41-62, 2002.

[14] J. Knull, C. Ardelean, S. Beauchemin, M. Bauer, "Automatic Identification of Turning Maneuvers from Vehicle Data", Computer Applications in Industry and Engineering (CAINE), pp. 127-134, October 2017.

[15] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, "A Driver Behavior Recognition Method Based on a Driver Model Framework", *SAE Transactions*, 109(6):469-476, 2000.

[16] M. Land, "Eye Movements and the Control of Actions in Everyday Life", *Profress and Retinal and Eye Research*, 25(3):296-324, 2006.

[17] Y.-C. Lee, J. Lee, and L. Ng Boyle, "Visual Attention in Driving: The Effects of Cognitive Load and Visual Disruption", Human Factors: *The Journal of the Human Factors and Ergonomics Society*, 49(4):721-733, 2007.

[18] X. Liu, H. Mei, H. Lu, H. Kuang, and X. Ma, "A Vehicle Steering Recognition System Based on Low-Cost Smartphone Sensors", *Sensors*, 17(3):29 pages, 2017.

[19] J. Lowenau, P. Venhovens, and J. Bernasch, "Advanced Vehicle Navigation in the BMW Real Time Light Simulation", *Journal of Navigation*, 53(1):30-41, 2000.

[20] M. Lu, K. Wevers, and R. van der Heijden, "Technical Feasibility of Advanced Driver Assistance Systems (ADAS) for Road Traffic Safety", *Transportation Planning and Technology*, 28(3):167-187, 2005.

[21] J. McCall, D. Wipf, M. Trivedi, and B. Rao, "Lane Change Intent Analysis using Robust Operators and Sparse Bayesian Learning", *IEEE Trans. on Intelligent Transportation Systems*, 8(3):431-440, 2007.

[22] S. Metari, F. Prel, T. Moszkowicz, D. Laurendeau, N. Teasdale, S. Beauchemin, and M. Simoneau, "A Computer Vision Framework for the Analysis and Interpretation of the Cephaloocular Behavior of Drivers", *Machine Vision and Applications*, 24(1):159-173, 2013.

[23] C. Munigety and T. Mathew, "Towards Behavioral Modeling of Drivers in Mixed Traffic Conditions", *Transportation in Developing Economies*, 2(6):20 pages, 2016.

[24] Y. Noguchi, R. Nopsuwanchai, M.Ohsuga, and Y. Kamakura, "Classification of Blink Waveforms towards the Assessment of Driver's Arousal Level - an Approach for HMM Based Classification from Blinking Video Sequence", Engineering Psychology and Cognitive Ergonomics, pp. 779-786, 2007.

[25] M. Plochl and J. Edelmann, "Driver Models in Automobile Dynamics Application", *Vehicle System Dynamics*, 45(7-8):688-741, 2007.

[26] D. Sandberg and M. Wahde, "Particle Swarm Optimization of Feed forward Neural Networks for the Detection of Drowsy Driving", *Int. Journal Comp. Neural Networks*, 38:788-793, 2008.

[27] D. Sanpeng, X. Xiaoli, Y. Xuecui, and M. Dehua, "Research on the Driver Fatigue Monitoring Method Based on the Dempster-Shafer Theory", *IEEE Proceedings of the Control and Decision Conference (CCDC)*, pp. 4176-4179, 2010.

[28] L. Vlacic, M. Parent, and F. Harashima, *Intelligent Vehicle Technologies: Theory and Applications*, Butterworth-Heinemann, 2001.

Jennifer Knull received her B.Sc. (Honors) and her M.Sc. degree from the University of Western Ontario, Canada. The focus of her Master's thesis was on the characterization of driver behavior for turn maneuvers. Her research interests include augmented driving, data processing and analytics, and database systems.

Steven S. Beauchemin is an Associate Professor of Computer Science at Western University London, Ontario, Canada. Dr. Beauchemin's research interests revolve around sensing techniques for the automotive industry. Dr. Beauchemin leads a team of graduate researchers and students in the use of sensing and tracking technologies for advanced driving assistance systems.

Michael A. Bauer is a Professor of Computer Science at Western University London, Ontario, Canada. He was the Chair of the Computer Science Department at Western University London from 1991 to 1996 and from 2002 to 2007. From 1996 to 2001, he was the Associate Vice-President of Information Technology. He served on NSERC's (National Science and Engineering Research Council) Computer Science Grant Review Committee from 2005 to 2008 and was the chair of the committee in 2008–2009. He was the Principal Investigator for the CFI project that initially funded the creation of SHARCNET (www.sharcnet.ca) - a multi-university high performance computing grid. He is currently the Scientific Director for SHARCNET. His research interests lie in the fields of distributed systems, mobile and vehicular applications and communication, high performance and parallel computing. He has published over 220 refereed articles. He has served on the organizing and program committee of numerous conferences and has refereed for a variety of international journals. He has supervised over 70 graduate students. Prof. Bauer is a member of the IEEE and the Association for Computing Machinery and has served on various committees of both organizations.

Using Orbital Network for Scalable Multi-Core

Nagi Mekhiel*
Ryerson University, Toronto, Ontario M5B 2K3, CANADA

Abstract

The increase in speed and number of processors in a chip achieved by continuous improvements in technology is causing major obstacles to the design of a suitable NoC, with enough bandwidth for all processors. The time spent in waiting for the messages between processors with the bandwidth of available from NoC limits the performance gain of multi-core.

We propose a new Orbital Network on a Chip that maps messages to time so that processors need complex NoC to communicate. A shared orbit is used to broadcast all messages between processors that are mapped to time, such that each processor is assigned a specific time slot in which it can place in the shared orbit the content of its message to the desired processor. Messages of all processors spin in orbits and rotate in a fixed orbital cycle time where each processor can snoop in the orbit and receive and send messages. We present systems with single orbit, adaptable, out of order and multi level for Orbital network. Results show that adaptable, out of order and multi level systems are scalable to a large number of processors.

Key Words: Network on chip; scalability of multi-core; memory organization; communication in parallel computers; scheduling of messages.

1 Introduction

The traditional shared bus multiprocessor system cannot be made scalable to a large number of processors due to the over head of arbitration and the waiting time to acquire the bus which is proportional to the square of number of processors according to [9]. The bus structure cannot support multiple processors at the same time, therefore processors must wait for each other to acquire the bus one at a time [9].

Network on Chip (NoC) [5], [1,2,3,6,7,8] helps in making the connection between processors less severe and can dynamically route traffic through switches to avoid congestion. The links of the NoC are shared by many signals. A high level of parallelism is achieved, because all links in the NoC can operate simultaneously on different data packets. Therefore, as the complexity of systems keeps growing, a NoC provides enhanced performance (such as throughput) and better scalability in comparison with previous communication architectures as shared buses.

Although NoC has many important advantages, it has performance limitations due to the interconnects essentially requiring multi-hop communication between any non-neighbor nodes. This causes high latency and increased power consumption. The large number of nodes is the main reason that limits the performance of NoC with wired communication. In the future, the number of processors and memories in a single multi-core system will increase to hundreds or even thousands [10]. Using wired line as the communication channel may cause a high latency, high power consumption and low throughput. The longer the path, more interference from other crossing paths will increase latency and power. In addition, the throughput will decrease. If there are hundreds or thousands of nodes integrated on a single chip, the problem will become very severe limiting the scalability of future multi-core.

We need a new method of communication that can offer a large number of multiprocessor systems messages at low cost and predictable time. This can be achieved with a simple system that its complexity does not increase with the increase number of processors and effectively deals with contention, size of network used, and latency.

2 The Concept of Orbital Data

Mapping data to time "TAM" rather than space has been proposed in [11] and its applications and performance evaluation was presented in [12]. Each data element is accessed in a specific time and not in space. There is no address lines needed to map it in space. TAM need not to arbitrate to access one location at a time because the data becomes available in specific time whether it is needed or not. Data spins continuously in orbits and becomes available to the outside in specific time.

A single orbit that uses a fast single link is available for all processors to read messages placed on it from all processors all the time. Each processor can place a message on this link at a specific time. Mapping messages to time rather than using complicated NoC with multidimensional links and switches and buffers, makes communication simpler and faster.

Figure 1 shows the concept of single level orbit that is shared by multiple processors to post their messages mapped to time.

*Department of Electrical and Computer Engineering. Email: nmekhiel@ee.ryerson.ca

Processor P1 places its message at time slot1 , and followed by others in a sequential order to the last processor PN that places its message at time slot N. The orbit rotates the content of the messages and spins them allowing each processor to change or post new messages in the orbital cycle. All processors snoop on the shared orbit to receive messages from any other processors in the assigned time slot. Processors write messages while it is posting it during the time slot assigned to each message. Conventional NoC has to route messages in space to reach the destination processor.

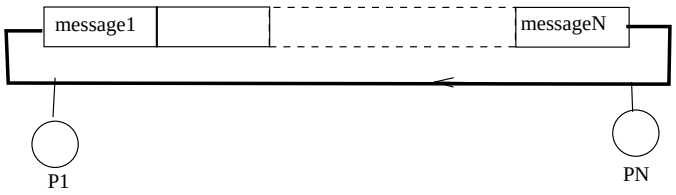

Figure 1: Concept of orbital data

3 Multi-Level Orbital Data

To deal with a large number of processors, a multi-level orbital messages is proposed [11,12]. Each orbit spins messages with a cycle time that depends on the number of messages in that orbit. A processor could receive messages from any processor by waiting for the time to be in a higher orbit if it does not share the same orbit. Multi-level orbits allows messages to be accessed from different orbits at different cycle time. Cycle time is the time it takes to post all messages from all processors in that orbit in sequential order until it is posted again in a cyclic fashion.

Figure 2 shows the concept of multi-level Orbits. The whole messages from all processors spin on ORBIT0, which has the longest cycle time. Each message occupies a specific time slot in each orbit and the contents of lower level orbits each appear in a specific time at the next higher orbit. Other lower level orbits are rotating their contents at the same time in a cyclic fashion each posting its messages in the higher orbit at a different time in sequential order. ORBIT1 has a portion of the messages that are in ORBIT0 therefore it spins at a faster orbital cycle time. ORBIT2 has the smallest number of messages and spins at the fastest cycle time. When the contents of messages in ORBIT2 needs to be in ORBIT1, both are aligned and become a section that belongs to ORBIT1 without a need to copy or transfer messages. This is because the messages in ORBIT1 are divided into an integer number of messages from multiple ORBIT2 messages. Therefore messages could be available in different orbits at the same time, creating parallelism in communication that is different from conventional messages between single sender to single receiver.

Figure 3 shows the mapping of the different messages in time for the different orbit levels. At level2, there are J number of orbits each spins M messages in a cyclic fashion at the same time in parallel with orbit cycle time=M which is the

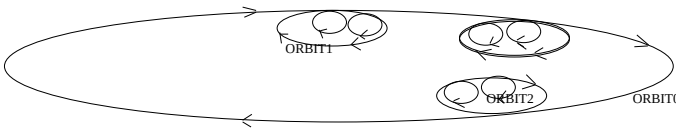

Figure 2: Concept of Multi-Level Orbits

fastest orbit level. Message X1 of a level2 orbit appears in a higher level orbit1 periodically after N=JxM cycles. There are K number of parallel level1 orbits each with cycle time N=MxJ that are spinning their messages in different parallel level1 orbits. Messages from X1 to XN of a level1 orbit spin at N clock cycles in level1 orbit and after KxN cycles, X1 of level2 are available at level1 and at the same time available at level0 (because it is one location no need to copy or transfer messages).

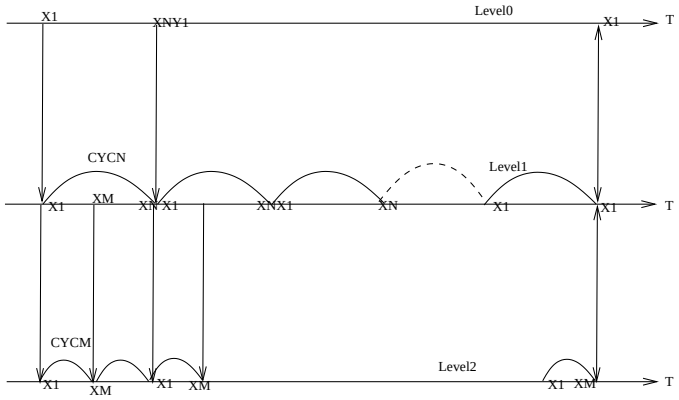

Figure 3: Time mapping of messages in multi-level orbits

4 Implementation of Basic Orbital Network

Messages are mapped to memory each at a specific assigned location that is accessed sequentially one after the other in time and implemented in DRAM technology or any other technology. It also could be implemented using a different organization than the known array structure to optimize the cost of implementation. We assume using the known DRAM technology to implement orbital messages and the contents of DRAM is accessed sequentially in a serial fashion without the need for an address or decoders. Because the time to access each location is known ahead of time, the activation time overhead is hidden and overlapped among the different DRAM banks.

The access time for each location will be only the time of accessing the sense amplifier. There is no address lines, and the DRAM will use the address lines to transfer more data which will double DRAM bandwidth.

Figure 4, shows the organization of orbital data. It consists of the following components:

DRAM Array: Organized as an array of N rows by M columns. Each cell is located at the intersection of a specific

row and a specific column. The storage cell consists of one capacitor and one transistor as in any DRAM structure.

Row Shift Register: Has N outputs, each is responsible to activate the corresponding row similar to the row decoder outputs in DRAM. The Row shift register consists of N D-type flip flop.

Column Shift Register: Has M outputs, each corresponds to a column selection output that allows the flow of data similar to the column decoder outputs in DRAM. It also uses a D-type flip flop.

Sense Amplifiers: Used to access the data from the input output DO/DI signal as in DRAM. The direction of data flow is controlled by a /WE signal as in any DRAM.

Time Control: Uses the state machine to provide the following signals:

Start-act-ROW: This initialize first D-flip-flop to be active for the row shift register.

Row CLK: Clock for the row shift register. The period of this clock defines the time for the row to stay active. Each row clock period, the row shift register activates the next row by shifting the active output of a flip-flop to the next.

Start-shift-COL: This initializes the first D-Flip-Flop to be active for the column shift register.

COL CLK: Clock for the column shift register. The period of this clock defines the length of time for the column to stay active. Each clock period, the column shift register activates the next column by shifting the active output of a flip-flop to the next.

/WE: Write enable signal, when it is active, the latches of sense amplifiers are updated with the value of data input in DI. On memory read, this signal becomes not active and the data from the sense amplifier are transferred to the output signal, DO.

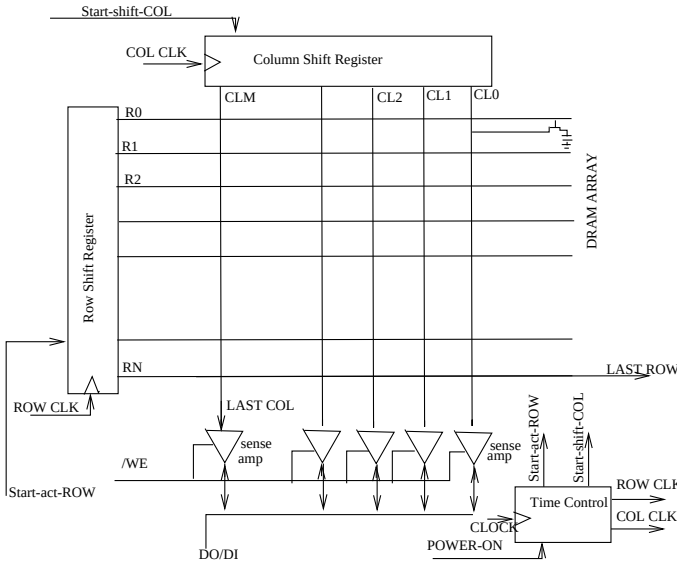

Figure 4: Orbital network organization

4.1 Orbital Network with Adaptable Access Patterns

In the basic orbital network system that has fixed access patterns, the access of memory starts from the first location of the first chip and ends at the last location of the last chip. The number of locations to access in a memory cycle is always fixed. The time to complete a memory cycle includes waiting for all memory locations to be accessed as mentioned above. The processor must wait until the location it needs to arrives.

Figure 5 shows a orbital network system that is adaptable to the demands of a processor and can change its access patterns. The adaptable system uses a control circuit to determine which chips to access. The control circuit uses a start address to select the first chip and a second address to select the last chip. The address lines is log base 2 of the number of chips.

To explain the operation, let us assume that the number of chips is 16 where each DRAM chip has 64 million locations, and the processor needs to access a memory section that starts from location 193 million to location 520 million. The starting address will be chip number 4 (193 divided by 64). The last chip address will be chip number 9 (520 divided by 64). Only chips 4, 5, 6, 7, 8 and 9 are accessed. The CONTROL circuit will make Start4 output signal active through decoding the start chip address from the processor. This is a pulse and only becomes active at the start of the memory access operation. It becomes active again when the LAST ROW signal of the last chip to be accessed (chip 9) becomes active. The following conditions make Start4 active:-

- The Decode of Start Address 4. This is only active for a time that is enough to start memory access similar to POWER-ON time.
- The Decode of End Address 9 AND LAST-ROW-of-chip 9 becomes active to allow the cyclic action of memory access.
- The CONTROL circuit decodes the END-chip address 9, and activates all the End signals of the chips from chip 4 to chip 9. Furthermore, End4, End5, End6, End7, End8 and End9 becomes active.

  The conditions to make any End signal active is:-

  – The chip number is equal or greater than the Start Address 4 AND LESS THAN OR EQUAL TO the End Address 9. The design of such a circuit is simple and could use a combination of a comparator, decoder and encoder.

4.2 Orbital Network with 'Out-of-Order' Access Patterns

The adaptable TAM system, explained above, has the following limitations:-

- Accessing of different locations must be in order, starting from first location of first accessed chip to the last location of the last accessed chip.
- The first accessed chip and length of memory locations to be accessed are fixed during the memory access cycle.

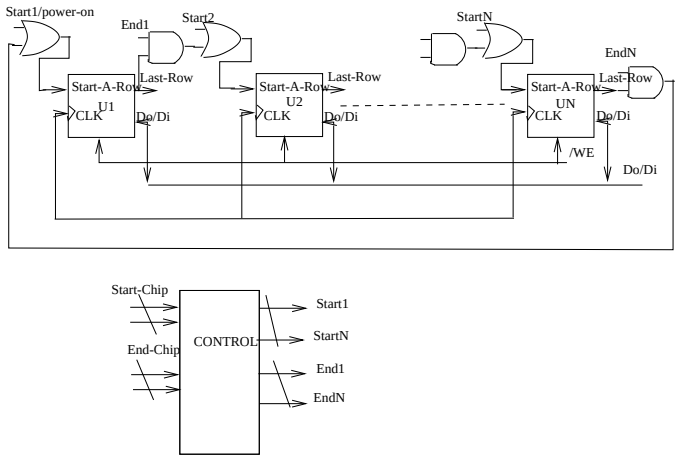

Figure 5: Orbital network system with adaptable access patterns

This means that during accessing the memory, we cannot change the order of accesses. In the above system the accesses are to chip 4, followed by chip 5, .. the last is chip 9. We cannot have accesses to chip 4, followed by chip 7, followed by chip 5.

To deal with 'Out-of-Order' access patterns and to solve the memory fragmentation, the following Orbital Network system organization is used. Figure 6 shows a block diagram of 'Out-of-Order' ACCESS organization. We assume that each chip consists of multiple banks, and each bank contains several memory rows. For example, a DRAM that has 1024 ROWS could be divided to 32 banks where each bank contains 32 rows. The number of locations in each bank depends on the number of columns in each row.

Each memory bank works as a basic message cell and provides accesses to locations in a serial fashion. The access of the last row in the bank generates a LAST-ROW signal indicating the last row access in the bank. The Start-A-Row signals the start of the serial access, and will activate the first row in the bank as explained above.

The DECODER and COUNTER generate the Start signals for any bank based on a select bank address and the mode signal. If the requested address is in Unit 17, and the mode is set for 'Out-of-Order' operation, the unit decodes the address and makes the Enable signal En17 active. En17 is connected to an AND gate that waits until the LAST ROW signal of the current accessed chip becomes active, indicating the end of bank access.

The value of the bank select address determines which bank to be accessed next in any order, thus allowing for 'Out-of-Order' access patterns.

To support 'In-Order' access patterns, this unit has a counter that counts from 0 to the number of banks. The counter output is connected to the decoder. If the mode is set for IN ORDER ACCESS, then the decoder decodes the output of the counter.

Message expansion with 'Out-of-Order' access patterns is simple and uses a controller that selects the start of the next chip based on a chip select address similar to bank select address. The controller waits for the LAST ROW signal from the current accessed chip and decodes the chip select address, then it generates the start activation of the row for the selected chip. The start signals of the rest of the chips are forced to be not active and these chips cannot start any access.

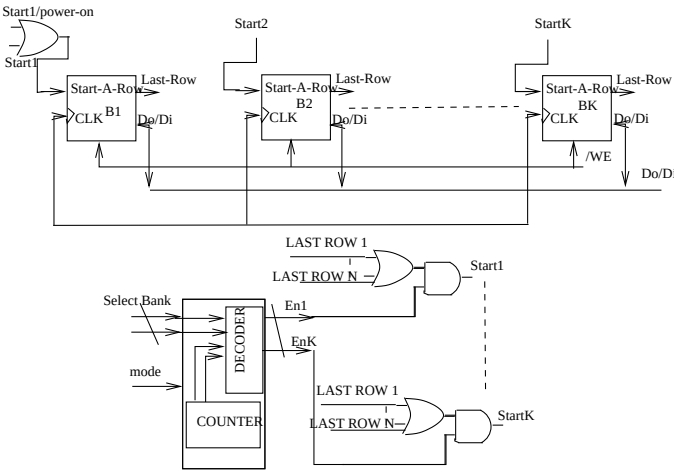

Figure 6: Orbital network system with out-of-order access patterns

4.3 Parallel Orbital Network with Multiplexer/De-Multiplexer

Orbital network organization that uses multiplexers can allow data in each message that maps to a memory bank to spin around in cyclic form and creates a parallel data accesses between each bank.

Figure 7 shows the orbital network organization using a multiplexer and demultiplexer. The select bank address selects the specific bank to be accessed among the multiple banks. It allows output data DO of the specific bank to pass through the multiplexer to the memory bus. It also allows the input data, DI from the memory bus to be passed through the demultiplexer to be written to the selected bank.

This organization supports Out-of-Order access patterns as the select address determines the accessed bank in any order. It also supports 'In-Order' Access patterns by using a counter and a mode signal to access banks 'In-Order' through the multiplexer/demultiplexer.

4.4 Multi-Core using Multi-Level Orbital Network

Figure 8 shows multi -level orbital network for multi-core. Each group of processors are connected to one multiplexer/demultiplexer to access sections of memory based on the selected bank address. Other groups of processors are similarly connected to form a parallel processor. This organization is different from the other known parallel systems because it provides a second level of parallelism achieved by parallel orbital network organization. Each parallel processor

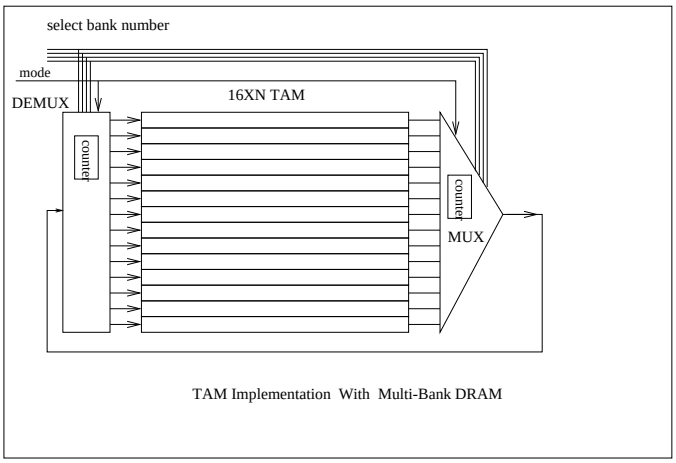

Figure 7: Orbital network implementation with multiplexer/demultiplexer

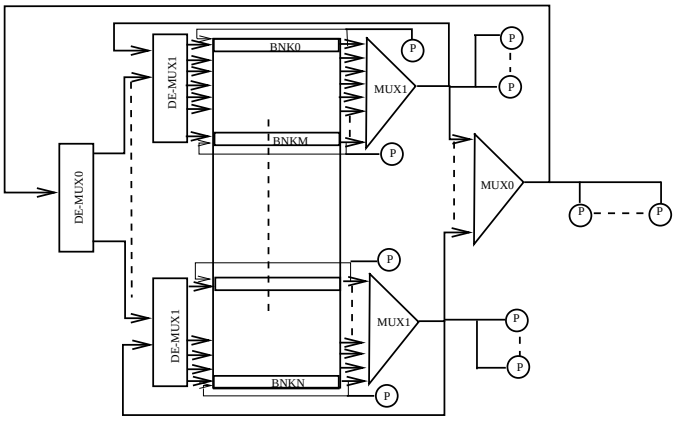

Figure 8: Multi-core using multi-level orbital network

group shares one portion of memory at a specific time without the need to exclude the other groups of processors. First level parallelism is obtained among the processor group sharing one multiplexer/demultiplexer as explained above in basic orbital data. The second level of parallelism is achieved among each processor group by using the higher level orbit. First level parallelism is obtained among the processor groups sharing multiple multiplexer/demultiplexer MUX1/DE-MUX1. MUX1/DE-MUX1 is used to route messages from each lower level orbit to a higher orbit. To route messages to the highest level, the system uses MUX0/DEMUX0.

5 Performance Evaluation of Multi Core with Different Orbital Network Systems

5.1 Application Equation Solver

We assume P parallel processors to solve NXN Matrix for finite element analysis application[12].

The cost of computation = NXN to get the average of 4 neighboring elements plus the element itself. The system uses parallel computer with messages as explained in [4]. The cost of communication = 2N size of two boundary rows if it uses strip decomposition and for block decomposition is given by:-

$$Time of communication for block decomposition = (\frac{4N}{P})^{1/2} \quad (1)$$

5.2 Performance of Single Processor

The single processor need not use messages and must compute the average value of each element in sequential order, therefore the time cost of computation is given by:-

$$T of Single processor = N \times N \quad (2)$$

5.3 Multiple Parallel Processors Using Single Orbital Network

The system with single orbit as explained in Fig1 to Fig4 has the following features:-

For 100 processor on a chip can use a shared orbit with a single bus, with a clock cycle time as fast as the processor clock as there is no arbitration, no switches, no buffers and is assumed to be 3 GHz, with an orbit width of 8 bytes. If each message size is 1 KB, the time assigned for each processor slot to place it in orbit is 1 KB at a bandwidth of 24 GB/S is about 40 ns. The total orbital cycle time for all messages from all processors will be 100 times the 40 ns or about 4 microseconds.

For the application equation solver, the computation time is NXN divided by P, and communication time cost per process is two rows of 2N.

The single orbit must send all messages for all processors in sequential order, therefore it is equal to 2NP.

$$T of single Orbit = \frac{N \times N}{P} + 2NP \quad (3)$$

5.4 Multi Core Using Single Adaptable Orbital Network

The system uses the implementation given in Fig 5.

An adaptable single orbit can be implemented by monitoring the waiting time for each processor to receive messages and reorders the assignments of each processor time slot such that the time from message sent to message needed is minimized. For example if processor P45 sends P4 a message at time slot 45, then processor P45 is reassigned to a time slot close to P4 time slot so that they can communicate in less waiting time.

An elastic and adaptable single orbit is also possible by limiting the number of processors assigned to time slots in the orbit to those that are currently active, therefore making the orbital cycle time much faster as it consists of less number of rotating messages. If only 10 processors are active then the cycle time will be .4 Microseconds for all messages from all processors.

To explain the need for adaptable orbital network, the application concurrency changes in time [4]. The system can reduce the number of processors needed for communication from maximum number P to optimum Pa that minimize the time cost of single orbit given above in equation [3], with differentiation.

$$Pa = (\frac{N}{2})^{1/2} \tag{4}$$

$$TofAdaptableOrbit = \frac{N \times N}{Pa} + 2NPa \tag{5}$$

5.5 Multi Core Using Single Out of Order Orbital Network

To minimize the cost of communication a block decomposition is assumed in which each processor is assigned to a block as explained in [4]:-

$$Tofblockcommunication = \frac{4N}{(P)^{1/2}} \tag{6}$$

The total time could be calculated assuming there is P number of messages in a single orbit, however the system needs to use out of order data access as given in Fig 6 because the data are not assigned to same row as in the basic single orbit.

$$TofOutofOrderOrbit = \frac{N \times N}{P} + 4N(P)^{1/2} \tag{7}$$

5.6 Multi Core Using Multi Level Orbital Network

A large scale multi-core that has 1000 processors for two level orbits, would have 32 processors sharing each lower level orbit as shown in Fig 8. There should be 32 multiplexer/demultiplexer needed to route these 32 orbits to the higher orbit. One multiplexer/demultiplexer is needed to route all messages from 1000 processors to the highest orbit. Total number of multiplexers/demultiplexers in this 1000 processors system is only 33. Cycle time of fastest orbit is 32x40ns= 1280 ns or 1.28 microseconds. Next level orbit cycle time is 32x1.28= 41 Microseconds for all 1000 processors. A faster orbital cycle time is possible using the adaptable or elastic features mentioned above for single level orbit.

A processor in any lower level orbit using adaptable orbital network will be able to post its message to any other lower level processor in the same orbit in latency less than orbit cycle time of 40 ns, and to any other processor of the 1000 processor by allowing the select lines of multiplexer to pass to outer orbit the message of the sender processor then allowing the select lines of the demultiplexer to select the receiver orbit in latency that is the same as 40 ns the cycle time of lower orbit.

This organization has the advantages of easy synchronization among different processors, flexibility in assigning processors to the different time slots in each orbit based on their need, and less bus loading for distributing processors among different bus levels.

For a three level orbital network, the first level orbit has a number of processors sharing parallel orbits equal to:-

$$Pnumberofprocessorsineachorbit = (P)^{1/3} \tag{8}$$

The number of parallel orbits in first level is:-

$$Nnumberoforbits = (P)^{2/3} \tag{9}$$

The total time to execute the equation solver in this three level orbital system is:-

$$TofMultipleOrbits = \frac{N \times N}{P} + (2NP)^{1/3} \tag{10}$$

5.7 Results of different Orbital Network Systems

Figure 9 shows the following:-

- The performance of a single processor is the slowest and has a execution time = 1000,000 steps.
- The performance of multi core with single orbit improves the execution time by 9 times for 10 processors, however when number of processors increases, the cost of communication increases linearly to limit performance gain to 5 for 100 processors, and 3 for 200 processors. This simple single orbital system is suitable for a small number of parallel processors.
- The adaptable orbit gives an improvement of about 11 times compared to single processors and this gain is constant for the large number of processors.
- The out of order performs better than the adaptable with performance gain of about 21 when using 50 processors but decreases for larger number of processors and gain is 16 for 200 processors.
- The multi level orbital system performs the best and its scalability is not affected by the large number of processors. For 50 processors , the gain is 50, and for 100 processors, the gain is 100 and for 200 processors, the gain is 200.

  This system is ideal for large scale parallel computing with multiplexers/demultiplexers organization. The gain of this system justifies the added cost and complexity of this organization.

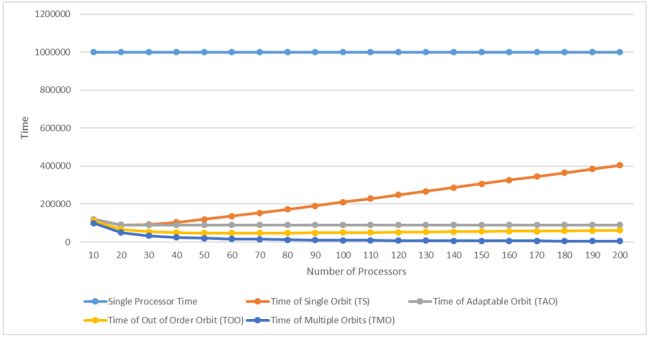

Figure 9: Results of single processor and multi core with different orbital network systems

6 Conclusions

Orbital network provides a much simpler and faster communication that is suitable for large scale multi-core. It reduces the complexity and power consumption by eliminating multidimensional network buffers and arbitration.

Large scale multi-core can use a multi-level orbital network to send and receive messages among thousands of processors at a much reduced latency with no contention and no need to use NoC with multiple links or routing algorithms that cannot scale with large number of processors.

References

[1] D. Bertozzi and L. Benini, "Xpipes: A Networkon- Chip Architecture for Gigascale Systems-on- Chip", Circuits and Systems Magazine, IEEE 4(2):18-31, 2004.

[2] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Cost Considerations in Network on Chip", *Integration - the VLSI Journal*, 38:19-42, 2004.

[3] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS Architecture and Design Process for Network on Chip", *Journal of Systems Architecture*, 50:105-108 February 2004.

[4] David E. Culler, Jaswinder Pal Singh and Anoop Gupta, *Parallel Computer Architecture: A Software/Hardware Approach* Morgan Kaufmann Publishers, San Francisco, California ISBN 1-55860-343-3.

[5] G. Dimitrakopoulos, A. Psarras, I Seitanidis, *Introduction to Network-on-Chip Design. In: Microarchitecture of Network-on-Chip Routers.* Springer, New York, NY 2015

[6] W. J.Dally, B. Towles, Route packets, not wires: on-chip inteconnection networks. In Proceedings of the 38th Annual Design Automation Conference (Las Vegas, Nevada, United States). DAC '01. ACM, New York, NY, 684-689. DOI= http://doi.acm.org/10.1145/378239.379048, 2001

[7] K. Goossens, J. Dielissen, and A. Radulescu, "A Ethereal Network on Chip: Concepts, Architectures, and Implementations", IEEE Design and Test of Computers, pp 414-421, 2005

[8] P. Guerrier and A. Greiner, " A Generic Architecture for On-Chip Packet-Switched Interconnections ", *Proc. Design, Automation and Test in Europe (DATE)* pp. 250-256, 2000.

[9] J. Hennessy, D.A. Patterson, *Computer Architecture: A Quantitative Approach* Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1996.

[10] ITRS. *International Technology Roadmap for Semiconductors*, 1 st edition, 2005.

[11] N. Mekhiel, , Data Processing with Time-Based Memory Access, US Patent 8914612B2, Dec 16 2014.

[12] Nagi Mekhiel, Introducing TAM: "Time Based Access Memory, *IEEE Access Journal, Special Section on Security and Reliability Aware System Design for Mobile Computing Devices*, 4:1061-1073, March 2016.

Nagi Mekhiel SM IEEE since'88, received the B.Sc. degree in electrical engineering (communication)from Assiut University, Egypt, in 1973, the M.A.Sc. degree in electrical engineering from the University of Toronto, in 1981, and the Ph.D. degree in computer engineering from McMaster University, Hamilton, ON, Canada, in 1995. He is a Professor with the Department of Electrical and Computer Engineering, Ryerson University, Toronto since 1990. His research interests are computer architecture, parallel processing, high performance memory systems, advanced processors, VLSI, and performance evaluation of computer systems. He holds many U.S. and World patents in memory and multiprocessors. He is conducting research to solve the fundamental problems facing computer industry, including scalability of parallel processors, and processor/memory speed gap.

Use of Ethereum Blockchain for Authentication, Access Control, and Data Sharing in Untrusted Environments

Syed Hasnain*, Abhinav Kalra*, Peter Bodorik*
Dalhousie University, Halifax, Nova Scotia, CANADA

Dawn Jutla†
Saint Mary's University, Halifax, Nova Scotia, CANADA

Sandeep Kuri*
Dalhousie University, Halifax, Nova Scotia, CANADA

Abstract

An enormous growth in data centric partnerships and collaborations has given rise to trust and transparency issues within information sharing ecosystems. Blockchain offers a decentralized computing environment and an append-only, immutable, and replicated data store that is kept in consensus by all participating systems. Ethereum is a platform based on the blockchain technology that also allows code, which can be executed and can interact with the internally stored data, to reside on the blockchain. We investigate the use of Ethereum for a data sharing solution that includes access control and authentication. We examine architectural alternatives and build a proof of concept for the selected alternative in order to explore how to adopt and adapt the inherent trust and transparency of blockchain into the data sharing solution that includes access control and authentication.

Key Words: Blockchain, ethereum, trust, security, privacy, data sharing, access control, authentication.

1 Introduction

As we move towards a more inter-connected world, we observe many companies collaborating with each other to share information to stay competitive in their space. Sharing data is governed by various real-world contracts put in place by both parties. However, as companies share information with more than one partner, who is frequently at the same time a competitor, privacy and trust are two important issues with which all companies must deal when sharing data. Many systems have been implemented and are used by data owners to provide various methods for data sharing. From FTP to integrated solutions, companies may need to have multiple file sharing solutions for an increasing number of partners with whom they need to share data with varying privacy requirements depending on the levels of trust. Data collaboration provides many benefits, including data driven decision making, better co-ordination amongst companies, and an overall improved oversight of structure of the data resources amongst partners (Verhulst & Sangokova, [14]). As companies have varying levels of trust when sharing data with other companies and thus require different data sharing solutions, the resource requirements for the different data sharing solutions impacts smaller companies the most as they frequently cannot afford the multiple data sharing solutions usually requiring infrastructure with security mechanisms.

We have seen Blockchain technology emerge as the new trust and transparency paradigm in the 21st century. Blockchain can be perceived as a distributed database or a ledger, keeping track in a chronological order, of all performed operations recorded in the form of ledger transactions. Blockchain systems, or the distributed ledgers, ensure that the data is present as a shared database that is always reconciled via consensus. Bitcoin is the first concrete implementation of a verifiable open ledger system (Rashid, [10]). Subsequent projects, such as Ethereum (Ethereum, [6]) and Hyperledger (Hyperledger, [15]), aim to expand upon the transactional nature of distributed ledgers by incorporating *smart contracts* with business logic that can be executed within the environment of the distributed ledger. Although the technology has been used to build primarily standalone applications, it can also be used to supplement existing systems. Specifically, it can be used to develop data sharing solutions to provide a new transparency layer to the trust environment of systems of collaborators with different levels of trust. This work examines the use of blockchain technology in forming a solution for data sharing in an environment in which there is either a lack of trust, or different levels of trust, amongst the collaborating parties who are small to medium-sized enterprises that cannot afford expensive solutions. We examine general approaches to providing data sharing solutions, and then provide a blockchain-based data-sharing solution that includes authentication and access control.

* Faculty of Computer Science. Email: (Syed.Hasnain, Abhinav.Kalra, Peter.Bodorik, Sandeep.Kuri)@dal.ca.
† Department of Finance, Information Systems, and Management Science, Sobey School of Business. Email: Dawn.Jutla@gmail.com.

The second section provides background on the need for data sharing and then on authentication, access control, the blockchain, and Ethereum, which is the chosen platform for our investigation. The third section examines and compares the architectural alternatives available for data sharing solutions as they apply to the environment in which data-sharing parties do not trust each other. Section 4 describes and compares alternative designs for the use of the blockchain technology for our project in order to choose the appropriate one for our environment. The design parameters include the use of blockchain for auditing, data structures, and smart contract code. We then describe how the chosen design alternative is applied in using Ethereum for data sharing that includes authentication and access control. In section 5 we describe implementation, while in section 6 we discuss execution costs. The final section provides conclusions.

2 Background

2.1 Data Sharing, Authentication, Access Control

As IT technologies evolved, data became an ever more valuable commodity not only for a single organization, but also for sharing amongst business partners to boost the efficiency and effectiveness of operations. According to Rashid et. al [10], a survey showed that 46 percent of all jobs surveyed involved data sharing and collaboration. How information is shared amongst the companies is constrained by the systems they use for data sharing. Furthermore, smaller companies cannot afford more sophisticated information sharing systems. Stefansson [13] investigated the state of information sharing amongst supply chain companies and showed that SME sized companies lack many forms of data sharing mechanisms. Another challenge is the standardization of data sharing formats and trust.

While databases have generally been the preferred solution for sharing data, there are various other ways and platforms to move data amongst the different partners. File Transfer Protocol (FTP) is a way of transferring files of any type across two ends. We can also see Web Services as another way of transferring data amongst different entities. Both technologies are capable of handling multiple data types and allow themselves to be accessed from within almost any technology eco-system. Apart from this, we also have the generalized data languages for the data exchange formats. JavaScript Object Notation, or JSON, and Extensible Markup Language, or XML, are the two data types utilized by web services, but can be extended to file formats as well.

Peterson et al. [9] envision a blockchain built from the ground up, focusing on the health sector where patient information is stored on the blockchain. Financial industries are also augmenting information sharing features on blockchain. Bank of America, Merrill Lynch, and HSBC are exploring data sharing strategies on blockchain for a Letter of Credit (Peterson, [19]). This allows various parties involved in the workflow to trust the system that can be (almost) fully automated.

When data is being shared amongst companies, clearly there needs to be authentication and access control mechanisms in place to ensure that data is accessed only by authorized users. As authentication and access control are well understood, established, and researched topics, here we are interested in exploring use of blockchains in their provision. Incorporation of the blockchain technology to data access control systems can add another layer of trust by providing a verifiable audit trail of all data access related activity that is timestamped and immutably appended to the blockchain. In Zyskind, Nathan & Pentland [16], the authors have elaborated on a blockchain-based approach for permission management for personal data. *MedRec* incorporates blockchain for access management of electronic medical records (Azaria, [1]).

Although blockchain technologies are new and in nascent use, a small number of startup companies have developed blockchain-based authentication systems for specific applications – examples include:

- *ShoCard* is a blockchain-based distributed application (DApp) to authenticate passport information that is stored on a blockchain (Dillet, [4]).
- *CryptID* is a system for membership cards. It uses multifactor authentication in which an ID number is associated with a QR code, while all user data is stored on a blockchain using Fatcom technology that encrypts all data creating a unique fingerprint (CryptID, [3]).
- *Dloc* is a document authentication system that applies a security chip to a document, while Fatcom blockchain is used for data storage (Smartrac, [12]).
- *Blocksign* provides a way to securely sign a document online. The signature includes the signers name and email address. After the user uploads the document it is hashed and stored on the Blockchain. The document's Blockchain address is associated with user's Blocksign account thus linking the signer's identity to the document (BlockSign, [2]).

2.2 Blockchain and Ethereum

Blockchain platforms, such as Ethereum, offer a number of key features. Firstly, the state of the system is reconciled across all nodes via consensus, which ensures that the system transitions from one valid state to another valid state. Secondly, being a continuously appended ledger, all updates to the system are immutable wherein any attempt to modify an earlier system state, written in blocks as part of the blockchain, will require a tremendous re-computation and is infeasible. It should be noted that the *smart contracts can create, view and edit their data structures, but invocations of the smart contracts operations, including their parameters, and any updates to their data structures are stored on the blockchain as immutable records.* Thirdly, trust and transparency are achievable since all transactions and updates to the blockchain are identified through a transaction hash and are timestamped.

Fourthly, as the ledger is replicated in each of the network nodes to which applications can connect, it offers availability and resiliency to node failures. Ethereum comes equipped with a Turing complete programming language to build complex applications. Ethereum Virtual Machine controls and holds the internal state of the machine while allowing it to execute programs called Smart Contracts.

3 Architectures for Data-sharing

To build a data sharing system that can provide all the required functionality, we shall first review the requirements and then explore alternatives for the system architecture. We then evaluate the alternatives for the purposes of selecting the most suitable one for our proof of concept.

3.1 Requirements

The system is expected to provide a basic form of User Identity management which must allow the administrators to store user information and provide authentication. Access Control Management (ACM) modules need to provide ways of maintaining the various permissions on a set of resources defined in the system. The permissions defined in the ACM modules also require that the resources must be permissioned by the ACM. And lastly, we require a mechanism through which data can be shared amongst the various parties that do not trust each other. We start with exploring the architectural alternatives for sharing the data.

3.2 Architectural Alternatives for Data Sharing

We explore three architectural alternatives: *distributed, centralized, and broker based* – each is discussed in turn.

3.2.1 Distributed Architecture. In a distributed architecture, wherein each individual party has its own hardware/software stack and where communication is achieved through agreed upon protocols, the possible number of channels of communication with different protocols is n^2, where n is the number of parties. Besides communication channels, when there is cooperation in software for various purposes, such as identity management and access control management, each party must keep track of all software protocols for the various purposes and their software stacks.

When dealing with data sharing, each request for data access must be verified individually along with the other party's identity. Thus, the architecture is heavily reliant on security measures between pairs of parties. The security, authentication, and access control compliance become very difficult in an environment in which a party negotiates different contract/protocol for data sharing, identity management, and access control with each of the other parties.

In terms of the benefits of a distributed approach, each party is in full control of its systems as it is autonomous. Furthermore, if any one of the nodes fails, it has minimal effect on the other nodes. Another important point to note here is that if a security failure happens, the other connected systems can lock out the affected party providing more control over the security aspects of this architecture.

3.2.2 Centralized Architecture. In a centralized architecture, all the parties utilize a common hardware and software platform to provide all their data sharing requirements. Advocates of centralized approach point out that these systems are robust enough to handle multiple entity types and that a centralized approach is easier to manage and scale in comparison to a distributed one.

With a single point of interest being the central system, each party must trust fully the central system. As no party can host its own solutions individually, each party uses the same protocol stack to communicate with the central system and hence, in comparison with a distributed system, the central approach greatly simplifies the management function. With a centralized system, all the parties must connect to the unified system, which allows for relatively easy maintenance and for cheaper support of the system. With a single system, each participant trains its employees and sets up resources for only one system.

Data sharing is usually controlled and handled by the central system via some governing authority. All the parties must trust the governing authority and the central system in handling their data. Many cloud solutions provide similar applications deployed on the cloud to make data accessible across multiple global locations.

One of the major disadvantages of the centralized system is the loss of control from the perspective of an individual party. All the parties are now reliant on a centralized solution that handles their data and privacy. If the central system goes down, it affects all the users/parties. Securing the central system becomes critical as all services are reliant on one system only.

3.2.3 Broker Architecture. A broker architecture is an architectural paradigm that focuses on structuring the distributed components as an integrated solution. This allows the decoupled separate components to interact with each other via a coordinating front. The broker is responsible for handling the overall functionality of the core systems which may be decoupled within the umbrella.

With a broker architecture, the individual components within the architecture are decoupled, thus allowing easier upgrades and support for third party modules. The freedom of modules depends upon the type of integration within the eco-system. This also allows the independence of physical locations of these modules and database as well. The broker architecture specifies the protocol to be used for collaboration, in our case data sharing, but does not dictate which software modules are to be used within each party, as long as the modules adhere to the communication protocol with the broker. The broker solution is similar to the centralized solution in that the parties must trust the broker authority since it acts as the controlling authority for collaboration. For data sharing, the oversight is done from the broker's end while there may be some interfaces

exposed from the party's end for tighter integration. The parties can add additional privacy and security levels at their ends.

3.2.4 Selection. A distributed system puts a burden on the individual party to host their own hardware and software that can collaborate with each of the other nodes, where the protocols used may be different, thus increasing the number of hardware and software modules that need to be managed. Furthermore, any new party that would like to participate in such an environment, would face great challenges to its hardware and software infrastructure and applications – it would need to accommodate individual setup for each of the other nodes. The above issues more than outweigh the benefits of full control and autonomy for each system.

Management of protocols within a central architecture is much easier in comparison to the distributed one, but central system, by its nature poses additional challenges in terms of availability, reliability, resiliency, privacy, and, perhaps most of all, trust in one single authority. If the central system were blockchain based, in which the ledger and the smart contracts are replicated, then the blockchain properties could be inherited by the centralized approach and, thus, the issues of availability, reliability, and resiliency would be alleviated. However, in such a case, we no longer have centralized systems, but something that resembles a replicated broker.

We choose the broker architecture for our data sharing solution approach because we can gain the benefits of the centralized solution and reap some of the benefits of the distributed architecture by judicious use of blockchain technology in the broker design and implementation. More specifically:

a. As each party communicates with the broker, each party needs to support only one protocol.
b. As the broker provides for authentication, identity management, and access control, the effect on the party's software infrastructure and applications is relatively small.
c. Although the parties must trust the broker in performing authentication and authorization, the trust is formed by:

    i. the broker code being implemented through the blockchain smart contracts that are located on the blockchain, which means that they are immutable and transparent, and by
    ii. providing trusted auditing functionality, based on the blockchain technology, that can be used by a party to ascertain that authorization is correct.

d. Although a broker is used, a single point of broker failure is avoided through use of the blockchain technology and smart contracts in which the data and code on the blockchain are replicated.
e. Individual parties do not have to build all modules from ground up and can utilize available modules on the broker end.

4 Design

There are several aspects that influence the choice of an appropriate design for authentication, access control mechanism, and data sharing that incorporate blockchain. These include defining the role of blockchain within the system, the data related to the operations that will be stored on blockchain, privacy of data stored on the blockchain and the functionality associated with smart contracts. The generally available approaches are reviewed below and then we describe the design of each of the data sharing, authentication, and access control components.

4.1 Alternatives for Using Blockchain

We review general approaches that can be adopted in using the blockchain technology in developing distributed applications. In the first one, the blockchain is used as an audit trail; in the second, blockchain smart contracts are used for mediation of requests, in our case for data sharing, authentication, and access control, while data structures are stored in an external storage, that is off blockchain. In the third one, the blockchain is used for both mediation/request-handling and holding the data structures that also contain hashes of the audit trail records. In the last one, full audit trail, data structures, mediation logic are stored on the blockchain.

4.4.1 Blockchain as an audit trail. This is the most trivial approach that incorporates blockchain. Blockchain, being a distributed ledger, wherein all transactions, once executed, become part of appended blocks, are immutable. These blocks, once they are added to the chain cannot be destroyed. In short, the blockchain is used as an immutable log of events capturing operations that were requested, for instance requests for authentication, access to an object, or a request to get a file from another company, and the system response – requests and responses that a user can verify.

4.1.2 Off-Blockchain Storage, On-Blockchain Mediation. This approach is to a considerable extent adapted from the work in Zyskind [16]. Herein, the data structures required by a requested operation reside on an external platform such as a relational table, DHT or any other media, while the requested operation code is implemented within smart contracts on blockchain. Although information/rules governing the requested operations are stored on external data structures, all updates to these data structures are either logged on the blockchain or are stored off blockchain, but their hashes are stored on the blockchain. The advantage of this approach is that the program logic is on the blockchain and hence is transparent and is difficult to hack.

4.1.3 On-Blockchain Data Structures and Mediation Logic. The data structures as well as the logic reside on the blockchain. Smart contracts are defined and deployed on blockchain that enable users to create, view, and update the data structures. The advantages of storing the data structure on the blockchain is avoiding a central point of failure, since the

blockchain state is replicated on all nodes in the chain, and the ability to establish trust and consensus on the state of data in an environment with untrusted parties. Please recall that smart contracts have access to their data structures and can update them; however, any updates are recorded as immutable records on the blockchain. As above, audit trail records can be stored either on the blockchain or their hashes are stored on the blockchain.

The first drawback associated with storing a larger dataset on the blockchain is the storage cost of a larger and possibly expanding dataset which is subsequently replicated on all nodes running the blockchain. Secondly, the computation cost to traverse and update the structures by the smart contracts will be higher.

The advantages however, of using this proposed architecture are manifold. Firstly, all data and the implementation logic reside on blockchain. Smart contracts execute the logic and interact with the data structure directly. No unreliable third-party tools or platform is needed. This promotes trust in users for the architecture. Secondly, due to the append-only nature of the distributed ledger, all changes and updates are immutable and can't be corrupted. Any mischievous attempt to alter the state of the data structures will require changes to blocks already appended in blockchain, a task that is computationally infeasible.

4.1.4 On-Blockchain Data Structures, Mediation Logic, and Audit Trail. The benefit of this approach is that the logic, data structures, and the full audit trail are all on the blockchain and hence gain the benefits of availability due to the replication on the blockchain nodes. However, there is also the cost associated with storing everything on the blockchain that must be considered.

4.2 Design Choice: On-Blockchain Data Structures and Mediation

Due to the primary advantages discussed above, the On-Blockchain Data Structures and Mediation/program appears as the most optimal method for the three major components: authentication, access control, and data sharing. The data structures for authentication and access control should not be large and hence the cost associated with their storage and delays in accessing them should not be prohibitive. As the audit trail size depends on the level of sharing and accumulates over time, it can be significant, and we therefore choose to store hashes of audit trail records on the blockchain while storing the records themselves off the blockchain. The architecture is shown in Figure 1. The broker interface APIs are exposed to the companies participating in the data sharing eco-system. Requests received by the broker result in invocation of the Ethereum's blockchain smart contracts that process the request and provide response. All requests, responses and events are logged in an audit trail in a DB that is accessible to clients via the broker's interface, while the hashes of audit trail records are stored on the blockchain to ensure that no one has tampered with the records, which is achieved by comparing a calculated hash of an audit trail record with the hash of the record as stored on the blockchain.

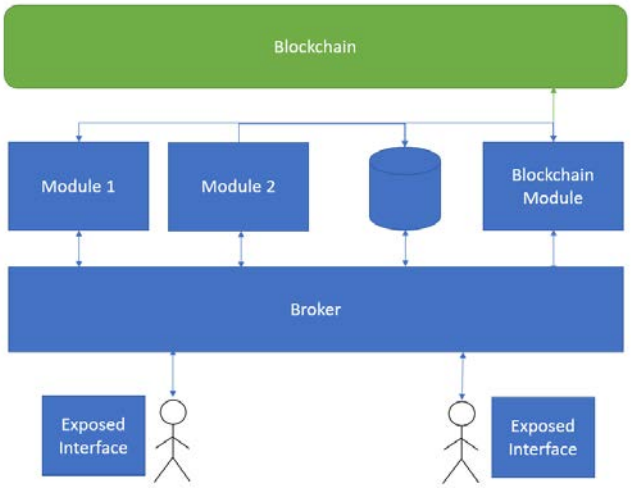

Figure 1: Broker architecture with blockchain

4.3 Data Sharing

We first discuss the workflow and then the broker's smart contracts that implement it.

4.3.1 Workflow

1. The company's user, who wishes to access another company's object, performs a login into the front-end of the broker.
2. The front-end forwards the request to the broker.
3. The broker then confirms the user's identity via the Identity Management module while also confirming if the user has access to the resource by checking with the Access Control Management module.
4. Audit trail records are created of the request and the result of authentication and access control are stored in the audit trail in a DB while the audit trail record hashes are stored on the blockchain.
5. Once both confirmations are in place, that is the user was authenticated and is authorized to access the object, the broker calls the second company's exposed web interface. The broker also provides the confirmation ID, provided by the Access Control Management module so that the request can be tracked on the blockchain as well as in the hosted DB, and a security token to be used in accessing the resource.
6. The second company responds with information to the broker and provides the API information where to obtain the object together with the hash of the object. Audit trail records and their hashes are created and stored by the broker.
7. The broker forwards the security token and the object hash to the first client (requesting the object) together with information on the second company's API to be used for

accessing the object.

8. The first company (requestor) uses the second company's API, while providing it the security token, to obtain the data object. The first company then calculates the object's hash and compares it to the one received from the broker and informs the broker of the completion of the operation.
9. The second company (object provider) informs the broker of the access to the object by the other company.
10. The broker logs all information received from the two companies about the transfer completion in the audit trail and stores the hashes of the audit trail records on the blockchain.

4.3.2 Smart contracts. The smart contracts utilized for data sharing, as opposed to authentication or access control, can be broken down into two categories. The first category is the Central Smart Contract in which all the successful transfers are being stored. A successful transfer may be defined as a complete cycle from end to end of the file transfer process. This does not include the failed transfers and denied requests as they are recorded in the company's contracts. We utilize the Events functionality of the smart contracts to achieve external logging recorded outside the contract. Also, we have put up covenants that the transactions to the blockchain will only be performed by the contract owner (in this case, broker has been defined as the contract owner) while others can only read the data.

The central contract contains a *Request_Response struct* which contains information such as first and second *transferID* along with the *aclconfirmationID* (provided by the Access Control Management Module). The *RequestID* provides a primary feature, along with *aclconfirmationID* from the Access Control Management module, that allows for tracing how the authorization was provided via its internal data structure. We also have the *dochash* string which holds the hash of the document value being transferred. Other components of the smart contract include the event *Request_Response_Listener* that fires when the data is written via the *RequestData* function.

The other category of contracts is the company specific Smart Contracts. Each company contract corresponds to each individual company. This allows for contract information, which is stored on the blockchain, to be shared with the specific company only while a broker also has access to these contracts. This decision was made to allow companies to leverage the *Events* functionality so that they can listen to the activities on their contracts live in real time.

For each individual company, we have a separate contract with the same code and data structures. A contract for a company has *Request* and *Response structures* with the information about the incoming requests and outgoing response, while for the central contract, we have the *Request* and *Response* events. Corresponding functions have also been added to include information in the data structures. These contracts also have the same covenant as the Central Contract where the contract is owned by the broker and the data can only be written by the broker. The Application Binary Interface (ABI) and the address of a company's contract is available to the company, so it can easily review the incoming requests and outgoing responses.

4.4 Authentication

Authentication, in general, is based on one or more basic factors, namely: knowledge, possession, inference, time, and location. In this project we implemented a two-factor authentication in which the user first provides a password and, as a second factor, the system communicates to the user a code that the user then presents to the system as a second factor. The communication channel is usually either an SMS text message or an email message. The type of the communication channel is set up when the user initially registers. For our project, we used email as the communication channel. If the user passes authentication, she is provided with a security token, with a limited life-time, that is used for further authentication of any requests that the user invokes. Thus, to access a resource, the security token is supplied with the request and, before the request is allowed, the security token is validated by the resource server by contacting the authentication server, as is shown in Figure 2. Thus, the security token is used for authentication but not access control. The user first contacts an authentication broker that performs a two-factor authentication on the user and, if it passes, provides the user with a security token. To get a service, the service request must include a security token and, before the service is provided, the authentication server is contacted to validate the security token provided by the user. If the token is validated, the services are provided.

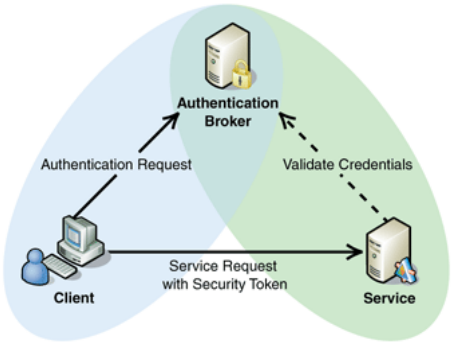

Figure 2: Authentication architecture (Adopted from https://www.codeproject.com/Tips/821772/ Claims-And-Token-Based-Authentication-ASP-NET-Web)

Our authentication server is a smart contract written in *Solidity* with methods to register a client, to authenticate a user, which is composed of two methods, one for each of the factors, one to validate a token, and methods to change a password and reset it in case the password is lost.

4.5 Access Control

Once the user is authenticated, the next step is to ensure that the user has appropriate access rights to the requested object for the requested operation. Without the loss of generality, we focus on .pdf file objects and operations to reads/retrieval only. Herein, the functionality is constrained to three primary objectives:

Access Control Policy: A user defined access policy, which states which objects the user is enabling to be shared with which other users within the application environment.

Access Privilege: The access privilege is defined as a read permission that a user is setting for a file object to allow another user to obtain access to that object.

Permission Management: The access control module processes the incoming object access request and depending upon request parameters and the policy of an applicable rule, the request will be permitted or denied.

Unless a user sets an '*allow*' permission on a file object, the permission is by default deemed as denied and object is not shared. Also, updates to access permissions within the access policy upon user removal/deletion is assumed to be managed by the parent system and is outside the scope of the current implementation.

4.5.1 Operations. The three major workflows deal with the creation and updates to the access policy by the users, confirmation of access permission by validating the access rule for each object request and updates to logs for audit purposes. The following nomenclature is defined for the terms that we will be using in the following text:

User Owner (UO): The user of the system, identified by a unique identifier, who wishes to share files he/she owns with other users in the system

User Requestor (UR): The user of the system, identified by a unique identifier, who wishes to access an object/file owned by another user in the system

Object (O): Herein defined as a data (.pdf) file residing on the host OS of the user who wishes to share it with other users using our application and identified by a unique identifier

Permission(S): The permission status: allow/deny that the user sets for their files

Policy(P): The access control policy defining all rules set by the users and stored on a data structure on the blockchain. The data structure is based on Authorization Relation based implementation of Access Matrix model (Samarati and Capitani de Vimercati, [11])

Rule(R): A record taking the format (UO, UR, O, S) stored in policy P.

4.5.2 Access Policy Updates. A user can define or update an access rule for a file object that he wishes to share with other users. A new rule is added to the access policy data structure whereas an update triggers modification to the data structure to reflect the change in the rule. The rule stored in the data structure as part of the Policy P can be visualized as a tuple consisting of the following attributes (UO, UR, O, S).

In case of a permission granted to multiple users for an object, there is a record/tuple in the policy P for each accessor to that object while the rest of the attributes are the same. When a rule in the policy is updated, the input provided to the smart contract takes the format: (UO, UR, O, S_{status_old}, S_{status_new}). The policy structure is traversed until the rule is matched and thereafter updated with the new status. The operation is mined and stored on the blockchain as a transaction with a unique transaction hash. This allows us to update the activity log as a verifiable audit trail which can't be disputed.

4.5.3 Check Permission. Before a file can be transferred between users, the access status/permission needs to be confirmed to determine if the owner has allowed that file to be shared with the user requesting access to it. This is achieved by matching the request against the rules in the Policy P. If a matching grant rule is found via parsing the policy P, access is granted or else the request is denied.

An incoming access request takes the form of the tuple (UO, UR, O, S_{read}). The output of this operation returns permit/deny value along with certain other parameters such as rule hash and block timestamp.

4.5.4 Audit Updates. All updates to the policy data structure are appended as transactions to a DB and their hashes are stored on the blockchain. Each transaction returns a distinct transaction hash and is mined and appended in blocks on the blockchain. These distinct transaction hashes serve as irrefutable and immutable proof of the occurrence of every update on the system. These values can be appended to the audit trails and made available to the users for verification.

Further, events are triggered from within the smart contract functions when access policy P is updated and functions return values when an access permission is sought and checked against the rules in policy P. The events are captured by listeners on the front end and appended to the audit logs as well. Similarly, the return values from the permission check methods can also be appended to the audit logs for audit purposes. This log data can be made available to applications outside the blockchain as well to establish trust and transparency.

5 Implementation

The scope of the proof of concept is to build the data sharing solution incorporating the blockchain aspect while using Ethereum smart contracts to achieve the stated objectives. We used *Solidity* to write the smart contracts to implement our operations for all three major components: data sharing, authentication, and access control. The smart contracts for authentication, access control and data sharing, described above, were subsequently deployed at addresses on the Ethereum blockchain to be executed within its environment.

The contract creation and deployment process were aided with the use of *Embark* framework and *EmbarkJS Javascript library* (Embark-framework, [5]). For our implementation, we use both *TestRPC* simulator and *Geth* Ethereum client running a private Ethereum node. In our design, Ethereum smart contracts implement the functionality described above:

Data Structures: Data structures store the necessary information used by the smart contract code. In case of authentication, they store the pairing of user information and authentication required; in case of access control they store rules which make up the access policy P; while in case of data sharing they store information on authenticated requests and their progress and APIs of companies to be used to access the files.

Functions: Functions that implement the necessary logic for each of the components.

Events: Events that are fired post execution of a workflow and propagate updates to the audit logs as well as to external application and higher-level interface. They are used within our implementation to keep track of activities.

5.1 Software Stack

Ethereum based applications require multiple development technologies to build a cohesive system. The user interface is the conventional Web front-end with JavaScript, HTML, and CSS. To connect the web front-end with Ethereum, Web3 API was used as a low-level mediator between blockchain and the application. Smart contracts were written in Solidity language. At the bottom layer sat the Ethereum blockchain with *Geth* running as the blockchain client.

To minimize the existing dependency on multiple software stacks, we have utilized the same stack with some added essentials for application development. The user interface for the broker application utilized the JavaScript/HTML/CSS combination combined with NodeJS for server-side code. The Ethereum based Smart Contracts ran on *Geth* while we utilized *MySQL* database for data storage along with its connectors for *NodeJS*. The web services on the client's end are built using *.Net WebAPI 2* that provides the functionality for file sharing. Figure 3 depicts the software stack while Figure 4 shows a modular view of the implementation.

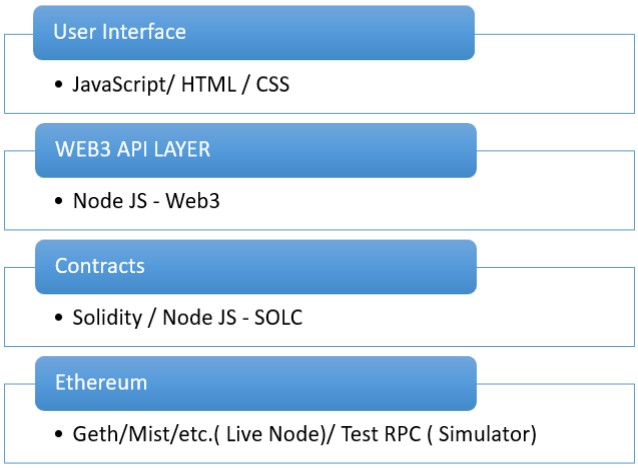

Figure 3: Software stack for Ethereum application development

6 Evaluation and Discussion

We evaluated the implementation by designing a custom user interface and performing the operations defined above. In the backend, we run a live Ethereum node via the *Embark* blockchain simulator for development environment. The *Embark* blockchain simulator runs as a real *Geth* client but with custom settings for development purposes on a private blockchain environment. We updated the configuration so that the mining difficulty is minimal and new blocks are mined almost instantly.

Our test showed that, as expected, the delay for a request to obtain a file object can be broken down into overhead cost,

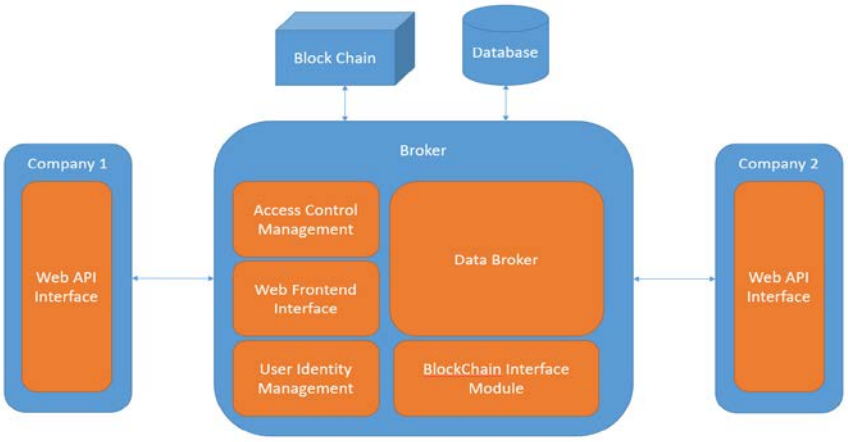

Figure 4: Modular overview of the proof of concept implementation

due to interaction between various components, authentication, access control, and the file transfer cost itself, which is directly proportional to the file size. Furthermore, for larger files, the file transfer cost dominated. This is not surprising, as the delay introduced by the broker and the blockchain is independent of the file size and hence constitutes a fixed size overhead, while the transfer of the file itself is directly proportional to the file size. However, as the file transfer delay depends primarily on the communication network(s) between the two parties, we do not report it here. There are countless references that explore delays over various networks of various types. One of the studies conducted by Krishnamurthy et al. [8] describes the numerous factors that affect the end-to-end performance of data transfer. Apart from the general factors such as network delay, server load and total bytes transferred, there are also more involved factors such as multi-server content, caching and various protocols.

Any contract that is deployed to blockchain has two types of major cost associated with it. The *transaction* cost is the cost associated with executing the contract code on the blockchain as a part of transactions. The transaction cost depends on various variables, such as instructions that are executed and their complexity. The second type of cost, referred to as *execution* cost, is associated with storing various global variables and methods on the blockchain. There is also cost of deploying the contract on the blockchain – but considering that this *deployment* cost is a one-time cost to deploy a contract on the blockchain, we simply report it but do not elaborate upon it. All the costs measured in this section are taken from Browser Solidity (Remix), which is an open source software available online. It allows contracts to be compiled directly in the browser.

Computational costs depend upon the complexity of code being executed on the smart contract and the size of data variables being updated or operated upon (Ethereum costs, [6]), where in Ethereum, it is computed in units of *Gas*. Computational expenses and their corresponding costs in the cryptocurrency, Ether, play crucial roles when it comes to the deployment and use of smart contracts for decentralized applications on the blockchain environments. Consequently, our evaluation places stress on these parameters.

The one-time *gas* execution cost of the contract deployment was about 1700000 units of *gas.* Similarly, setup and configuration costs, such as user registration and preparing access control rules, are also relatively infrequent operations. More important are the execution and transaction costs of methods for actual data sharing that can be executed frequently. Table 1 shows the transaction and execution costs associated with the individual components, that is smart contract methods for the purposes of data sharing, authentication, and access control. Authentication includes login and second factor costs and access control includes the cost of determining whether a request to access an object should be accepted or denied.

We note that the execution cost of one Data Sharing operation is stable as it excludes the authentication and access control costs. The same applies for authentication costs per one operation. However, our current implementation of access control is such that the execution cost of determining, whether a subject has access to the requested object/file, includes scanning rules and hence depends on the number of rules. As the number of rules increases, so does the execution costs. Consequently, in a production environment we would have to re-write the data structures containing the rules to avoid a sequential scan.

7 Conclusions

This work focused on creating a system which leverages the features of the blockchain technology, like immutability and transparency, and adapt them in a data sharing solution focused around trust-less environments. We considered possible architectures when utilizing blockchain for these types of solutions and analyzed their strengths and weaknesses. Our proof of concept utilized the broker architecture allowing it to be relatively independent while providing the essential features required for a data sharing solution. Our smart contracts allowed us to do authentication, access control, and data sharing, while providing transparency to the parties involved. The parties can also gain trust from the contracts and their availability for inspection on the blockchain.

In our implementation, the data structures that store the access control policies and rules reside on Ethereum via smart contracts. The increase in the size of the data also increases the size of the blockchain on all the nodes in the network. Further, due to increasing data structure size, larger amounts of computational currency, *gas,* will be consumed for operations. *Gas* or *Ether* is in public blockchains obtained by the process of mining, which in turn is computationally expensive, or by spending fiat currency to purchase it. Thin clients do not have the sufficient processing power and system capabilities to handle this adequately. Thus, we see that the scalability of the application is bound by the design and the limitations of the underlying blockchain platform.

Any updates to the state of a contract results in the permanent update to the overall state of the Ethereum blockchain which is reconciled via consensus across all nodes of the blockchain. These updates are recorded as transactions which form parts of blocks that are subsequently mined. While updates to blockchain data by legitimate processes, such as contract functions, are available, a malicious or unauthorized attempt to rewrite the state of the blockchain will require re-computation of subsequent blocks. Mining process and consensus via proof-of-work involves considerable computational expense. This makes any malicious attempts to rewrite the distributed ledger state intractable. Further, since the transactions are hashed and timestamped and recorded into the blockchain, a trustworthy history of updates to the blockchain is maintained. This engenders trust as well as verifiability of audit records.

Table 1: *Gas* Execution and transaction costs for the modules

Contract Type	***Operation Type***	***Transaction Cost (Gas units)***	***Execution Cost (Gas units)***
Access Control	Contract Deployment	1699695	434106
Access Control	Access Policy Rule Creation	282826	253899
Access Control	Access Policy Rule traversal and read	31204 (avg)	7215
Authenticate -4 Eye	Deployment	1536270	1126562
Authenticate -4 Eye	Sign up	191510	167358
Authenticate -4 Eye	Password Update	34694	11054
Authenticate - Single	Deployment	1207843	879111
Authenticate- Single	Signup	107407	84727
Authenticate- Single	Password Update	10213	33789
Authenticate – 2 Factor	Deployment	1625752	1197036
Authenticate – 2 Factor	Sign up	148230	125550
Authenticate – 2 Factor	Password Update	33046	9598
Authenticate – 2 Factor	OTP Store	7876	30623
Authenticate – 2 Factor	OTP Compare	3796	26540
Authenticate – Factor	Login	26474	3730
Data Sharing -Central	Deployment	751107	530399
Data Sharing -Central	Process Completion	207904	181320
Data Sharing -Company (Individual)	Deployment	1035253	743409
Data Sharing -Company (Individual)	Request Data Addition	202281	175697
Data Sharing -Company (Individual)	Response Data Addition	133847	109695

References

[1] A. Azaria, A. Ekblaw, T. Vieira and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management", 2016 2nd International Conference on Open and Big Data (OBD). doi:10.1109/obd.2016.11, 2016.

[2] BlockSign, "Blocksign.com", https://blocksign.com/about, Retrieved 16 February 2018.

[3] Cryptid, "CryptID: Secure Your Identity Now", http://cryptid.xyz/, Retrieved 19 February 2018.

[4] R. Dillet, "ShoCard and SITA want to Store your ID Details on the Blockchain to Authenticate Travelers", https://techcrunch.com/2016/05/25/shocard-and-sita-want-to-store-your-id-details-on-the-Blockchain-to-authenticate-travelers/, Retrieved 19 February 2018.

[5] Embark-framework, "Embark-framework", https://github.com/iurimatias/em bark-framework, Retrieved February 19, 2018.

[6] Ethereum, "Ethereum Blockchain App Platform", https://www.ethereum.org, Retrieved February 19, 2018.

[7] Ethereum Accounts, "Account Types, Gas, and Transactions", Ethereum Homestead 0.1 documentation. Ethdocs.org. http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html, Retrieved February 19, 2018

[8] B. Krishnamurthy and C. Wills, "Analyzing Factors that Influence End-to-End Web Performance", *Computer Networks*, 33(1-6):17-32, 2000.

[9] K. Peterson, R. Deeduvanu, P. Kanjamala, and Kelly Boles, "A Blockchain-Based Approach to Health Information Exchange Networks", https://www.

healthit.gov/sites/default/files/12-55-blockchain-based-approach-final.pdf, Retrieved February 19, 2018.
[10] F. Rashid, "Employees Need Tools for Secure Inter-Company Collaboration", http://securitywatch.pcmag.com/security/295791-employees-need-tools-for-secure-inter-company-collaboration, Retrieved February 19, 2018
[11] P. Samarati and S. Capitani de Vimercati, "Access Control: Policies, Models, and Mechanisms", *International Conference on Foundations of Security Analysis and Design*, pp. 137-196, 2001.
[12] Smatrac Launches dLoc, "Smartrac Launches dLoc – a Breakthrough Document Authentication Solution Leveraging Blockchain", from https://www.smartrac-group.com/pr/smartrac-launches-dloc.html, Retrieved 17 February 2018.
[13] G. Stefansson, "Business-to-Business Data Sharing: A Source for Integration of Supply Chains", *International Journal of Production Economics*, 75(1-2):135-146, 2002.
[14] S. Verhulst and D. Sangokoya, "Data Collaboratives: Exchanging Data to Improve People's Lives", https://medium.com/@sverhulst/data-collaboratives-exchanging-data-to-improve-people-s-lives-d0fcfc1bdd9a, Retrieved February 19, 2018.
[15] Welcome to Hyperledger, "Welcome to Hyperledger Fabric", from http://hyperledger-fabric.readthedocs.io/en/release/, Retrieved February 19, 2018
[16] G. Zyskind, O. Nathan and A. Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data", IEEE Security and Privacy Workshops (SPW), San Jose, CA, USA, doi:10.1109/SPW.2015.27, pp. 180-184, 2015.

Syed Shahzeb Hasnain (photo not available) holds a Bachelor of Science degree, in Computer Science, from University of Karachi and a Masters in Applied Computer Science degree from Dalhousie University where he researched sharing of objects using blockchain technology, specifically based on the Ethereum platform. He is currently a Blockchain/Cloud Software Engineer at Peer Ledger Inc. where he is using his blockchain and cloud computing expertise in developing a distributed application, which uses a Hyperledger fabric, to track and trace minerals in a supply chain. Before this role, he was a .Net(software) Consultant for a leading bank in UAE.

Abhinav Kalra (photo not available) holds a Bachelor of Engineering in Information Technology from Sinhgad College of Engineering and a Masters in Applied Computer Science degree from Dalhousie University where he researched access control mechanism based on the Ethereum blockchain technology. He is currently working as a Blockchain/Integration Engineer with Peer Ledger Inc. He has experience with Bitcoin, Ethereum, and Hyperledger blockchain platforms. He previously worked at TCS on projects for clients in investment and retail banking domains, where he managed and coordinated onsite and offshore project teams and liaised with business, operations teams, vendor and product support teams on critical migrations and activities.

Peter Bodorik (photo not available) is a Professor at the Faculty of Computer Science, Dalhousie University where he has held various administrative positions, such as Director of Master of Electronic Commerce, Associate Dean Academic, and Associate Dean Research (Acting). He has published over eighty refereed papers in conference and journal venues with concentration on managing data in distributed systems, in which he focused on efficient mechanisms for transaction management and querying; electronic commerce, with concentration on e-commerce models and bench marking; and privacy with concentration on models and tools to support privacy on the client and server sides. Currently, he is interested in blockchain technologies and recently, with Dr. Jutla, has applied for patents to USPTO on the identity bridge and on provenance of goods and services in responsible supply chains. He received and participated in many grants and awards from NSERC, CFI, and industry. Over eighty graduate students have successfully completed their studies under Dr. Bodorik's (co)supervision at both the masters and PhD levels.

Dawn Jutla (photo not available) received her Master and Ph.D. degrees in Computer Science in the areas of distributed shared memory and multi-view access control respectively from the Technical University of Nova Scotia. Dawn then spent 20+ years doing multi-disciplinary R&D and consulting in computer science and business from the Sobey School of Business where she currently holds the post of the Scotiabank Professor of Technology Entrepreneurship and Innovation. In 2009, she received the World Technology Award for IT Software in the Individual Category for R&D contributions to online privacy. She is founder and CEO of Peer Ledger, a blockchain company.

Sandeep Kuri (photo not available) received a Bachelor of Technology degree in Computer Science from Rajasthan Technical University and a Masters in Applied Computer Science degree from Dalhousie University where he researched authentication mechanisms based on a blockchain technology, specifically based on the Ethereum platform. He is currently working as a Jr. QA Analyst at KINDUCT, a data analytics software provider. He was also working as a consultant for CGI Information Technology and Services.

Instructions For Authors

The International Journal of Computers and Their Applications is published multiple 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. Fred Harris, Jr. Fred.Harris@sce.unr.edu.
2. Illustrations should be high quality (originals unnecessary).
3. Enclose a separate page for (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.

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. Figures should be captioned and referenced.

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.
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). Integrate at the end of the paper.
   - Author Photos (jpeg files are required by the printer)
   - Figures, Tables, Illustrations. These may be integrated into the paper text file or provided separately (jpeg, MS Word, PowerPoint, eps). title of the paper.
3. Specify on the CD/DVD label or in the email the word processor and version used, along with the title of the paper.
4. 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 author will be invoiced for publication charges of $50 **USD** per page (in the final IJCA two-column format) to cover part of the cost of publication. For ISCA members, $100 of publication charges will be waived if requested.

January 2014