Computer Science & Information Technology 89

David C. Wyld
Jan Zizka (Eds)

Computer Science & Information Technology

10th International Conference on Networks & Communications
(NeCoM 2018) June 23~24, 2018, Copenhagen, Denmark

AIRCC Publishing Corporation

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Jan Zizka,
Mendel University in Brno, Czech Republic
E-mail: zizka.jan@gmail.com

ISSN: 2231 - 5403
ISBN: 978-1-921987-87-8
DOI : 10.5121/csit.2018.80901 - 10.5121/csit.2018.80908

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The 10th International Conference on Networks & Communications (NeCoM 2018) was held in Dubai, UAE during June 23~24, 2018. The 7th International Conference on Software Engineering and Applications (SEAS 2018), The 7th International Conference on Control, Modelling, Computing and Applications (CMCA 2018) and The 4th International Conference on Signal Processing (SP 2018) was collocated with The 10th International Conference on Networks & Communications (NeCoM 2018). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The NeCoM-2018, SEAS-2018, SP-2018, CMCA-2018 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, NeCoM-2018, SEAS-2018, SP-2018, CMCA-2018 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the NeCoM-2018, SEAS-2018, SP-2018, CMCA-2018.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld
Jan Zizka

Organization

General Chair

David C. Wyld	Southeastern Louisisna University, USA
Jan Zizka	Mendel University in Brno, Czech Republic

Program Committee Members

Abdellah Hadjadj	Avenue de l'Université, France
Abdelmajid Hajami	Faculty of Sciences and Technologies, Morroco
Alberto Taboada-Crispi	Universidad Central "Marta Abreu" de Las Villas, Cuba
Alex Afanasyev	Florida International University, USA
Ali Yazici	Atilim University, Turkey
Amir Salarpour	Bu-Ali Sina University, Iran
Baozhong Tian	University of Wisconsin, USA
Chaman Lal Sabharwal	Missouri University of Science and Technology, USA
Chang-Wook Han	Dong-Eui University, Korea
Cherif Adnen	University of Tunis Manar, Tunisia
Chouakri Sid Ahmed	University of Sidi Bel Abbes, Algeria
Christian Mancas	Ovidius University, Romania
Dac-Nhuong Le	Haiphong University, Vietnam
Daniel E. Asuquo	University of Uyo, Nigeria
Dmitry A. Zaitsev	International Humanitarian University, Ukraine
Doru Florin Chiper	Technical University Gh. Asachi Iasi, Romania
Fairouz Tchier	King Saud University, Saudi Arabia
Fernando Bobillo	University of Zaragoza, Spain
Franco Frattolillo	University of Sannio, Italy
Habib Rasi	Shiraz University of Technology, Iran
Hamid Ali Abed AL-Asadi	Basra University, Iraq
Hamri Mohamed Mehdi	University of djilali liabes, Algeria
Hanan Salam	University of Pierre and Marie Curie, France
Hani Bani-Salameh	Hashemite University, Jordan
Hyunsung Kim	Kyungil University, Korea
Ian Tan	Multimedia University, Malaysia
I-Ching Hsu	National Formosa University, Taiwan
Ilona Bluemke	Warsaw University of Technology, Poland
Jun Peng	University of Texas, USA
Karzan Wakil	University of Human Development, Iraq
Kecia Marques	CEFET-MG, Brazil
Kemal Avci	Izmir Democrasy University, Turkey
Keneilwe Zuva	University of Botswana, Botswana
Khaled Ahmed Abood Omer	University of Najran, KSA
Kosai Raoof	Le Mans Universite, France
Luiz Carlos P. Albini	Federal University of Paraná, Brazil

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Networks & Communications Community (NCC)

Soft Computing Community (SCC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

10th International Conference on Networks & Communications (NeCoM 2018)

7th International Conference on Software Engineering and Applications (SEAS 2018)

4th International Conference on Signal Processing (SP 2018)

7th International Conference on Control, Modelling, Computing and Applications (CMCA 2018)

HOLMES : HOLISTIC MICE-ELEPHANTS STOCHASTIC SCHEDULING IN DATA CENTER NETWORKS

Tingqiu Tim Yuan, Cong Xu, Tao Huang, Jian Li

Huawei Technologies

ABSTRACT

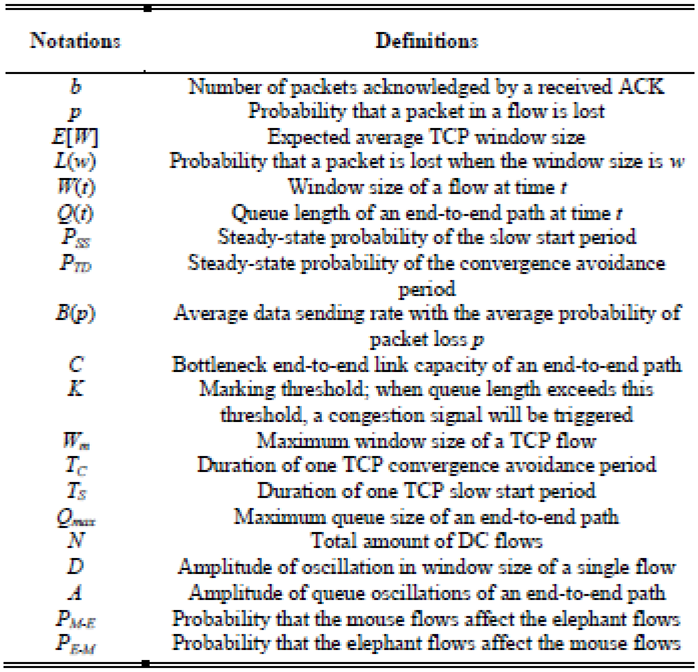
Dependability of Scheduling between latency-sensitive small data flows (a.k.a. mice) and throughput-oriented large ones (a.k.a. elephants) has become ever challenging with the proliferation of cloud based applications. In light of this mounting problem, this work proposes a novel flow control scheme, HOLMES (HOListic Mice-Elephants Stochastic), which offers a holistic view of global congestion awareness as well as a stochastic scheduler of mixed mice-elephants data flows in Data Center Networks (DCNs). Firstly, we theoretically prove the necessity for partitioning DCN paths into sub-networks using a stochastic model. Secondly, the HOLMES architecture is proposed, which adaptively partitions the available DCN paths into low-latency and high-throughput sub-networks via a global congestion-aware scheduling mechanism. Based on the stochastic power-of-two-choices policy, the HOLMES scheduling mechanism acquires only a subset of the global congestion information, while achieves close to optimal load balance on each end-to-end DCN path. We also formally prove the stability of HOLMES flow scheduling algorithm. Thirdly, extensive simulation validates the effectiveness and dependability of HOLMES with select DCN topologies. The proposal has been in test in an industry production environment.

1. INTRODUCTION

The wide adoption of diverse cloud-based applications and services exacerbates the challenges in design and operation of Data Center Networks (DCNs). Consider a typical mix of traffic patterns in cloud applications: mouse vs. elephant data flows. Mouse data flows (mice) are emails, web pages, data requests or any other short-lived data flows. Elephant data flows (elephants), on the other hand, are persistent data flows such as VM migrations, data migrations, MapReduce and other application flows that impact network bandwidth for minutes or hours if not longer. In multi-tenant mode, long-lasting elephant and short-lived mouse flows share on DCN paths [15]. According to the results shown in [7], the sizes of the numerous short-lived flows are usually less than 10KB, the majority of which tend to require very low latency. On the other hand, the long-lasting heavy DC flows are typically much larger than 100KB; although the number of these large flows is extremely small compared to that of the small flows. These elephant flows account for more than 80% of bandwidth in DCNs. The competing performance requirements of the two types of flows make it challenging to schedule them on shared paths. That is, the throughput-oriented large flows demand for high bandwidth, while the latency-sensitive small flows prefer minimal queuing delay. Note that, in hyper-scale data centers there are also many flows that are both high bandwidth and require low per packet latency. Such flows can be handled as an extension to the strategy of scheduling mice-elephants flows.

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP - 2018
pp. 01–21, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80901

Table 1 : Summary of Key Notations and Definitions



Notations	Definitions
b	Number of packets acknowledged by a received ACK
p	Probability that a packet in a flow is lost
$E[W]$	Expected average TCP window size
$L(w)$	Probability that a packet is lost when the window size is w
$W(t)$	Window size of a flow at time t
$Q(t)$	Queue length of an end-to-end path at time t
P_{SS}	Steady-state probability of the slow start period
P_{TD}	Steady-state probability of the convergence avoidance period
$B(p)$	Average data sending rate with the average probability of packet loss p
C	Bottleneck end-to-end link capacity of an end-to-end path
K	Marking threshold; when queue length exceeds this threshold, a congestion signal will be triggered
W_m	Maximum window size of a TCP flow
T_C	Duration of one TCP convergence avoidance period
T_S	Duration of one TCP slow start period
Q_{max}	Maximum queue size of an end-to-end path
N	Total amount of DC flows
D	Amplitude of oscillation in window size of a single flow
A	Amplitude of queue oscillations of an end-to-end path
$P_{M\text{-}E}$	Probability that the mouse flows affect the elephant flows
$P_{E\text{-}M}$	Probability that the elephant flows affect the mouse flows

To provide high bisection bandwidth, DCN topologies are often multi-rooted topologies, e.g. Fat-Tree, Leaf-Spine, characterized by a large degree of multipath. There are multiple routes between any two DCN endpoints [1, 2]. However, a critical issue in such network topologies is to design an efficient scheduling mechanism to balance the load among multiple available paths, while satisfying different application requirements defined in the Service Level Agreements (SLAs).

The de facto DCN flow scheduling scheme Equal Cost Multi Path (ECMP [3]) cannot meet such dynamic performance requirements in data centers. Hash collisions in ECMP can cause congestions, degrading throughput [4-6] as well as tail latency [7-9] of DCN flows. To balance the load between DCN switches and paths, stateful schedulers have been proposed, e.g. Conga [10], Hedera [4], etc. They monitor the congestion state of each path and direct flows to less congested paths, hence more robust to asymmetry network without control plane reconfigurations [11, 12]. Since maintaining global congestion information at scale is challenging, local congestion-aware or stochastic scheduling schemes are proposed, e.g. Expeditus [12], Drill [13], Hula [14], etc. Using simple or local information collection, these mechanisms are more efficient and applicable for complex DCN architectures, e.g. 3-tier Clos topologies. However, the majority of these scheduling mechanisms focus on balancing the loads of DCN according to the congestion information, without any consideration of cloud applications or data center traffic patterns.

Existing solutions to scheduling the mice-elephants hybrid DCN traffic fall into two categories. The first category deploys unified schedulers for both mice and elephants on shared paths, in spite of the competing performance requirements of the two. Based on the analysis of DC traffic patterns, these studies design novel scheduling algorithms or congestion signals [16, 17] and strike at the right balance between throughput and latency on shared DCN paths. The main challenge though is the interference between the elephant and mouse flows. The second category deploys network partition schemes that transfer the two types of flows over separate paths. [18] By isolating elephant and mouse flows, network partition solutions avoid the aforementioned interference. This is particular attractive as hardware and system cost continues to drop. Nonetheless, new policies are required to adaptively partition the DCN paths, given dynamic DC traffic patterns and varied DC architectures.

This paper focuses on the second category, the network partition solutions. Using a stochastic performance model, we first theoretically prove that the interference between mice and elephants are inevitable under the unified scheduler, indicating that network partition is a more appropriate solution for handling such hybrid DC traffic. We then propose a novel scheduling scheme, HOLMES, for such hybrid traffic in datacenters. HOLMES architecture partitions a DCN into high-throughput and low-latency sub-networks, decouples the two competing scenarios and eliminates the interference in the hybrid traffic. HOLMES further deploys a stochastic and global congestion-aware load balancing algorithm that optimizes the performance of each sub-network. The stability of the HOLMES flow scheduling algorithm is also proved and validated in this paper.

The main contributions of this work are:

- We prove with a closed form the necessity of applying network partition solutions to mice-elephants DC traffic, showing that unified flow scheduling schemes or congestion control mechanisms are not optimal.
- We design HOLMES, an architecture that partitions all the DCN paths into sub-networks, isolating the paths of elephant and mouse flows. As a result, HOLMES eliminates the interference between the two. Moreover, an AI module of the HOLMES architecture enables machine learning techniques and offers a more comprehensive analysis of the DCN status for better scheduling.
- We propose a novel stochastic load-balancing algorithm for HOLEMS that is global congestion aware. Using the end-to-end congestion signal, the algorithm is agile to link or node failures in DCN. Moreover, it uses only limited global congestion information to improve the scheduling efficiency and sustain stability in all possible scenarios.
- We use an analytical model to evaluate the adaptability of HOLMES. We then confirm the finding with detailed simulations under varied DCN architectures and scheduling policies. These experimental results can further guide the hardware implementation of HOLMES.

The rest of this paper is organized as follows. Section II presents an overview of HOLMES. Section III and IV discuss in detail the HOLMES architecture and its load balancing mechanisms, respectively. Section V evaluates HOLMES with extensive simulation experiments. Section VI concludes the paper.

2. HOLMES OVERVIEW

HOLMES aims to satisfy the competing performance requirements of elephants and mice by partitioning a DCN into high-throughput and low-latency sub-networks. HOLMES proposes a stochastic and global congestion-aware mechanism to schedule the elephant and mouse flows in these separate sub-networks. This section presents an overview HOLMES.

A. Necessity of Network Partition

In order to lay out a solid foundation of HOLMES, we first theoretically prove the limitations of the unified scheduling schemes when handling hybrid DC traffic. The analytical model in turn promotes the need for deploying network partition solutions in the architecture design of flow scheduling and congestion control schemes.

The main challenge that any unified scheme needs to address is the interference between the elephant and mouse flows. A typical scenario that mouse flows affect elephant flows is that a burst of mouse flows triggers a congestion signal, e.g. Quantized Congestion Notification (QCN), which in turn reduces the transmission rates of elephant flows in the next long period of time. Measurement results show that the short-lived flows in large-scale clusters exhibit significant traffic bursts. Theophilus Benson et al further explain that the traffic bursts in DCNs are caused by the huge amount of aggregated edge links as well as the concurrency of parallelized computing schemes, e.g. shuffle period in a MapReduce process. Experimental results also show that a momentary traffic burst can lead to short-lived congestion in a DCN. On the other hand, the effect of elephant flows on mouse flows is that elephant flows occupy the vast majority of the network buffers; hence the mouse flows will result in long queuing delay and flow completion time (FCT), a.k.a. the bufferbloat problem.

We calculate the probability that the above two scenarios would occur with the unified flow scheduling solutions and how they would affect network performance. Some key notations and definitions are illustrated in Table I. We then prove that fundamental limitations exist in the unified scheduling schemes when handling hybrid DC traffic. (Details of the proof is omitted in this paper due to page limit, however, we will post it after publication.) Therefore, network partition is a better solution that eliminates the interference by separating the mouse and elephant flows.

B. HOLMES Architecture with Network Partition

W. Wang et al [18] and W. Cheng have applied the network partition solutions to separate the large and small flows in DCN paths, and to avoid the interference between the two.

Similarly, HOLMES deploys the network partition solution, which partitions all the end-to-end DCN paths into two logical sub-networks to isolate the transmission of elephant and mouse flows. A centralized controller is deployed to map each link to the high throughput or the low latency sub-network. The traffic transmitted between each pair of edge switches may contain both elephant and mouse flows. Therefore, HOLMES architecture needs to ensure that at least one of the multiple paths between each two edge switches belongs to the high throughput sub-network, while at least another one belongs to the low latency sub-network too. The scales of the two subnetworks can be determined either statically or dynamically according to the architecture of the DC as well as the traffic patterns.

In addition to network partition, network statistics and analysis functions are provided by the HOLMES AI module. We discuss the detailed architecture in Section III.

C. Global Congestion-aware Load Balancing Algorithm based on Power of Two Choices Model

Another important component in HOLMES is its load balancing algorithm. Since local congestion-aware load balancing algorithms have been proved to react slowly to link failures [10, 14] and are prone to form congestion trees, we deploy global congestion-aware load balance in HOLMES. Inspired by the Power-of-Two-Choices model [43, 44], we further design a stochastic load-sensitive flow scheduling algorithm, which reduces the overhead of congestion information maintenance and improves the scheduling efficiency.

M. Mitzenmacher and S. Ghorbani et al have shown that tagging a single unit of memory with random sampling and identify the shortest output queue from the previous time slots guarantees stability [13, 43]. We extend a distributed implementation of this approach: When a packet arrives at an input port of an edge TOR switch, that TOR switch resolves the source and destination addresses of the packet.

It then assigns the packet to the least loaded of d randomly chosen end-to-end paths from all N (d << N) paths. The choice of the previous time slot is saved for the scheduling of the next time slot.

Section IV will discuss the detailed process and analyze the stability of this load balancing algorithm.

3. HOLMES ARCHITECTURE

Fig. 1 shows the HOLEMS architecture in three layers: AI layer, control layer and infrastructure layer. The HOLMES AI layer contains a cognitive computing cluster to implement the software AI module. The AI module collects the DCN state information from the DCN monitors, and applies the machine learning algorithms to generate some analysis results, e.g. networking tendency predictions, network outlier locations, etc. These learning results generated by the AI module provide a more comprehensive view of DCN behaviors. They will be then used for network partition and flow scheduling operations.

The HOLMES control layer is responsible for generating the network partition, congestion control and local balancing policies, based on the monitoring information as well as the learning results generated by the AI module. The policies generated in the SDN are decomposed into a series of finegrained partition and scheduling orders, which are transmitted to the DCN switches and end hosts for execution. Without deploying the HOLMES AI module, the functions provided by the control layer are the same as the traditional SDN controllers.

The HOLMES infrastructure layer executes the specific network partition as well as the flow scheduling operations. It is responsible for storing, forwarding and processing data packets. The detailed operation orders are transmitted and configured on the DCN switches. The DCN switches first map each link to the high throughput network or the low latency subnetwork, according to the network partition policies. When the elephant and mouse flows arrive at a DCN switch, their packets are scheduled to the pre-partitioned paths separately. This process is managed by the HOLMES control layer.

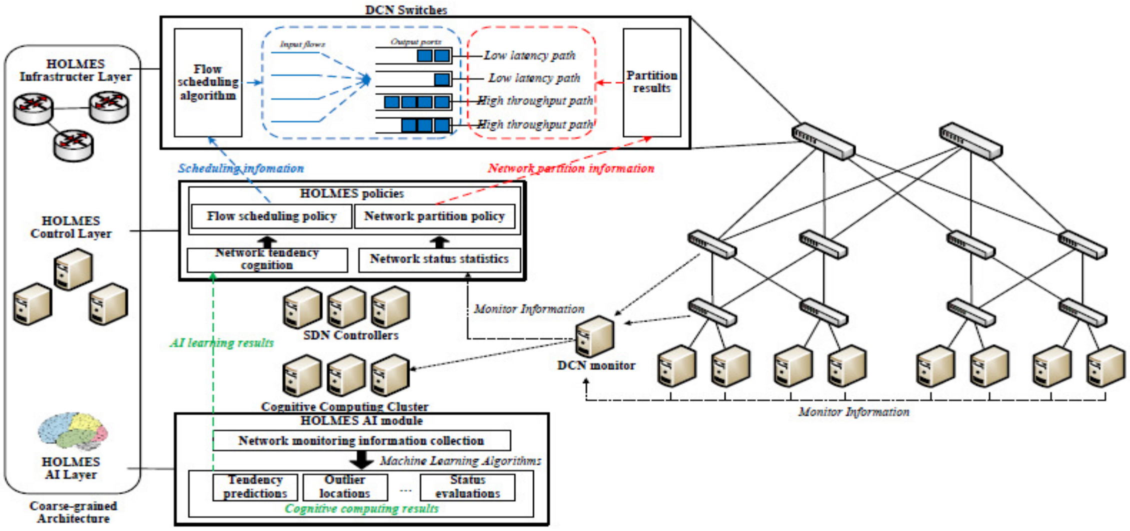


Fig. 1. HOLMES architecture: Based on the real-time monitor information, HOLMES AI module first analyzes the status or tendency of the DCN using machine learning models, and generates the learning results. Next, according to the analysis results, HOLMES control layer designs a network partition policy and a corresponding flow scheduling policy, and the policies are generated in the SDN controllers. Finally,

the network partition as well as the flow scheduling operations will be executed on the DCN switches or hosts, under the guidance of the SDN controllers.

A. HOLMES Network Partition

Both static and dynamic network partition mechanisms are enabled by HOLMES. When the scale of the DCN is small, HOLMES provides static network partition policies to DCN switches to avoid the frequent scale changes of the two subnetworks. It also makes the execution of the network partition policy more stable. On the other hand, when the scale of the DCN is large, HOLMES will deploy the dynamic network partition solution that dynamically adapts the scales of the two types of sub-networks to the traffic load variants. Moreover, the dynamic network partition solution utilizes the network resources more efficiently, especially when the arriving rate of DC traffic is slow.

Both static and dynamic network partition mechanisms have been studied [18]. One can either integrate these mechanisms in HOLMES or implement the appropriate network partition mechanism according to the scale of the DCN as well as the traffic patterns.

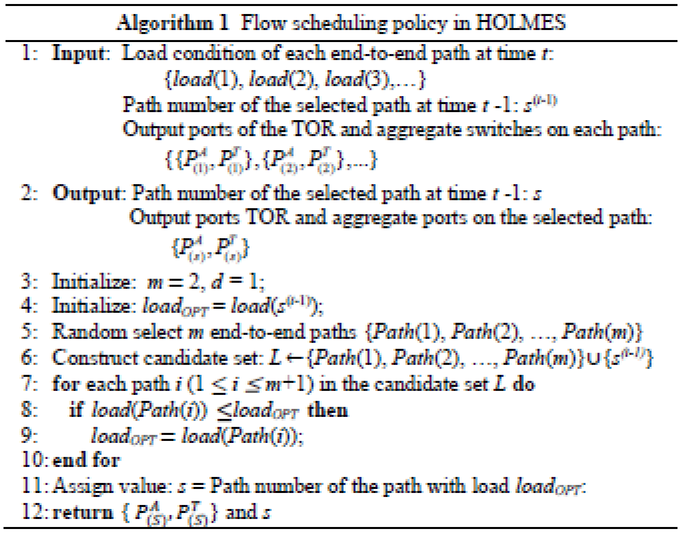
Both static and dynamic network partition mechanisms are enabled by HOLMES. When the scale of the DCN is small, HOLMES provides static network partition policies to DCN switches to avoid the frequent scale changes of the two subnetworks. It also makes the execution of the network partition policy more stable. On the other hand, when the scale of the DCN is large, HOLMES will deploy the dynamic network partition solution that dynamically adapts the scales of the two types of sub-networks to the traffic load variants. Moreover, the dynamic network partition solution utilizes the network resources more efficiently, especially when the arriving rate of DC traffic is slow.

Both static and dynamic network partition mechanisms have been studied [18]. One can either integrate these mechanisms in HOLMES or implement the appropriate network partition mechanism according to the scale of the DCN as well as the traffic patterns.

B. Application Scenarios for HOLMES Architecture

Compared with the commonly used SDN architectures, a prominent feature of HOLMES is the implementation of the AI module and its machine learning algorithms. Machine learning methods have been widely used in network management and DC scheduling policy generation operations. Those continuing learning and analysis results provide a comprehensive understanding of network features and behaviors, which benefits the designing of the corresponding network partition and flow scheduling policies. Therefore, the deep analysis and accurate AI prediction provided by the AI module enable the HOLMES architecture to perform more complex and intelligent operations.

One typical application scenario for HOLMES architecture is the deployment of application driven networks (ADN), where a physical network is sliced into logically isolated sub-networks to serve different types of applications. Each network slice in ADN can deploy its own architecture and corresponding protocols, to satisfy the requirements of the applications it serves. The key operations when implementing ADN are: (1) Constructing an application abstraction model to formulate the resource requirements of the applications; (2) mapping the distinct properties of applications to respective network resources. It is shown that the complexity and performance of these operations can be improved when some application features are pre-known [8]. Hence, the HOLMES AI module benefits the analysis of application features as well as the prediction of resource requirements, which further alleviate the complexity of application abstractions and mapping operations. Moreover, the design and implementation of network slicing mechanisms can also be realized by the cooperation of the control layer and the infrastructure layer.



Algorithm 1 Flow scheduling policy in HOLMES

```
Input: Load condition of each end-to-end path at time $t$:
        $\{load(1), load(2), load(3), \ldots\}$
       Path number of the selected path at time $t$ -1: $s^{(t-1)}$
       Output ports of the TOR and aggregate switches on each path:
        $\{\{P^A_{(1)}, P^T_{(1)}\}, \{P^A_{(2)}, P^T_{(2)}\}, \ldots\}$
Output: Path number of the selected path at time $t$ -1: $s$
        Output ports TOR and aggregate ports on the selected path:
         $\{P^A_{(s)}, P^T_{(s)}\}$
Initialize: $m = 2, d = 1$;
Initialize: $load_{OPT} = load(s^{(t-1)})$;
Random select $m$ end-to-end paths $\{Path(1), Path(2), \ldots, Path(m)\}$
Construct candidate set: $L \leftarrow \{Path(1), Path(2), \ldots, Path(m)\} \cup \{s^{(t-1)}\}$
for each path $i$ $(1 \leq i \leq m+1)$ in the candidate set $L$ do
  if $load(Path(i)) \leq load_{OPT}$ then
    $load_{OPT} = load(Path(i))$;
end for
Assign value: $s$ = Path number of the path with load $load_{OPT}$;
return $\{P^A_{(S)}, P^T_{(S)}\}$ and $s$
```

Similarly, HOLMES architecture is also applicable for some other intelligent or complex application scenarios, which demand a deep understanding of network or application features such as Internet of Vehicles (IoV), coflow scheduling and some other network architecture based on network slicing or network partitions. With the immense proliferation of complex and diverse cloud-based applications, we expect such architecture to be the development trend in the future.

C. Out-of-order vs. Efficiency

While elephants contribute to the majority volume of DCN traffic, mice account for 80% of the number of instances. Outof-order scheduling done at host side may sacrifice efficiency. In addition, compatibility with legacy hardware has to be ensured. Therefore, one may still want to consider deploy conventional, sometimes oblivious, ECMP scheduling for inorder scheduling of mice. However, we expect such overhead to diminish, while hardware and system cost continues to drop. As a result, HOLMES is the way to go.

4. HOLMES SCHEDULING ALGORITHMS

Once the hybrid DC traffic is separated into different subnetworks, scheduling algorithms affect the performance of each sub-network. In this section, we discuss the aforementioned global congestion-aware scheduling algorithm and prove the stability of its stochastic policy.

A. Necessity of Deploying Stochastic Scheduling Algorithms

Compared with the other state-aware flow scheduling algorithms, stochastic flow scheduling algorithms are more applicable for large-scale data centers according to the following reasons:

1) Simplification of computing complexity

One of the key factors that degrade the performance of the traditional ECMP mechanism is the lack of global congestion information. To overcome this limitation, a group of studies have designed new flow scheduling polices based on a global “macroscopic” view of the entire DCN, e.g. CONGA [10]. However, in large-scale and heavily loaded data centers, the global macroscopic load balancing algorithms introduce unacceptable computing complexity to deal with the massive information, and the control loops in these scenarios are much slower than the duration of the majority of congestion incidents in data centers [13]. Therefore deploying the stochastic algorithms to achieve micro load balancing is a more viable solution. The micro load balancing solutions require only limited congestion information, which simplifies the computing complexity and enables instant reactions to load variations in large-scale data centers.

2) Optimization of storage complexity

In data centers, 8 and 16-way multipathing are common, while there is growing interest in multipathing as high as 32 or even 64. Specifically, with 40 servers in a rack, there will be 40 uplinks. Each flow can use a different subset of the 40 links, leading to 2^{40} possible subsets. Keeping the state of each path in this scenario requires unacceptable storage resources, which is difficult to be implemented. On the contrary, stochastic scheduling algorithms are effective to cope with the optimization of storage complexity, as well as the small number of register reads. Edsall *et al* [26] deploy the stochastic power of-two-choices hashing solution for balancing the loads of DC routers. The storage complexity of such a stochastic solution is logarithmically reduced.

3) Better adaptability for heterogeneous DCNs

A typical flow scheduling method in multi-rooted DCNs is equal traffic splitting based on hashing, as used in the traditional ECMP approach. However, the uniform hashing approach cannot achieve optimal load balance without the assumption of symmetric and fault-free topology [5, 10], which is not generally true in heterogeneous data centers. To provide better adaptability for heterogeneous DCNs, weighted traffic distribution methods have been widely adopted in the global macro load balancing solutions [11]. In order to correct the imbalance caused by the even distribution approach and enable fair resource allocation, the weighted approaches distribute the traffic among available paths in proportion to the available link capacity of each path. The global weighted traffic distribution solutions have shown good adaptability to dynamically changing network topologies. However, these solutions still need real-time state information collection of all the paths, which introduces additional computing and storage complexity.

Stochastic flow scheduling algorithms can reduce the computing and storage overhead of weighted traffic distribution mechanisms, while maintaining the adaptability to heterogeneous DCN topologies. Consider the stochastic Power of-Two-Choices: The algorithm only needs to record the states of the two randomly chosen paths; therefore, the storage and computing complexity are dramatically reduced. Moreover, the algorithm compares the load conditions of these two chosen paths, select the better one, hence performs a weighted operation in another form. Stochastic load balancing solutions have also been proved to be applicable for heterogeneous DCNs [13, 26]. Based on these justifications, we extend stochastic flow scheduling algorithms to our HOLMES mechanism.

B. Flow Scheduling Algorithm in HOLMES

We consider a stochastic scheduling policy, (d, m) policy: The input port chooses d random end-to-end paths out of all the possible paths. It finds the path with the minimum occupancy among all the d samples and m least loaded samples from the previous time slot. It then schedules the input packet to the selected end-to-end path.

Increasing the value of d and m to >>2 and >>1 will degrade the performance, since the large number of random samples makes it more likely to cause the burst of packet arrivals on the same path [13]. As a result, we set m=1 and d=2 in our scheduling model. The detailed flow scheduling procedure is shown in Alg. 1.

Using global congestion information, the algorithm reacts rapidly to the link or node failures. Moreover, the limited information used in the algorithm improves the scheduling efficiency while avoids the traffic bursts on the same switch ports.

C. Stability Analysis of HOLMES's Scheduling Algorithm

We prove the stability of this stochastic global congestion aware scheduling algorithm in a two-tier Clos DCN topology. We abstract an end-to-end path in a Clos network (Fig. 2A) as a serial queuing system consists of a series of queues (Fig. 2B). As a result, the whole Clos DCN topology is abstracted as a queuing network. We then evaluate the performance of a specific leaf-to-spine DCN path using a stochastic queuing network model.

We focus on analyzing the stability of the scheduling process from when a packet arrives at a TOR switch to when the packet reaches the spine switch. The packet experiences two queuing processes, at the TOR and the aggregate switch port, respectively. The entire path from the TOR node to the spine node can also be modeled as a large queue.

Based on the results of [56] and with a similar method shown in [57], we prove that HOLMES's scheduling algorithm is stable for all uniform and nonuniform independent packet arrivals. Some key notations and definitions used in the scheduling model are illustrated in Table II.

We prove that the global (1, 1) policy is stable for all admissible parallel arrival process. We construct a Lyapunov function L as follows:

$$L(t)=\sum_{i=1}^{NN_A}(\tilde{Q}_i(t)-Q^*(t))^2+\sum_{i=1}^{NN_A}Q_i^2(t)$$

To prove the algorithm is stable, we show that there is a negative expected single-step drift in the Lyapnuov function, i.e.,

$$E[L(t+1)-L(t)\,|\,L(t)]\le \varepsilon L(t)+k \quad (\varepsilon,k>0)$$

We divide the Lyapunov function into two sub functions as:

$$L_1(t)=\sum_{i=1}^{NN_A}(\tilde{Q}_i(t)-Q^*(t))^2 \qquad L_2(t)=\sum_{i=1}^{NN_A}Q_i^2(t)$$

Based on the above formulation, we prove that there exists a negative expected single-step drift in the Lyapnuov function in each possible case. Therefore, the global (1, 1) policy is stable. Based on the (d, m) policy, the HOLMES's scheduling algorithm is also stable. (Details of the proof is omitted due to page limit; however, we will post it after publication.)

5. HOLMES PERFORMANCE EVALUATION

We evaluate HOLMES using simulation based on OMNET++. We construct a test-bed simulation platform to simulate the data transmission process in symmetric and asymmetric fat-tree DCNs. Similar to [10], a heavy-tailed distribution is used to generate DC flow of different sizes. The hosts of the DCN run TCP applications. The flow request rate of each TCP connection satisfies Poisson process.

By comparing the performance of the same DCN under different load balancing mechanisms, we validate HOLMES and analyze the influence of the network partition solution and end-to-end scheduling schemes on DCN performance. Moreover, we evaluate the adaptability of HOLMES's load balancing algorithm for heterogeneous architectures and discuss some technical issues in the hardware implementation of HOLMES.

A. Evaluation of HOLMES Network Partition

We evaluate the network partition policy of HOLMES in a scenario that hybrid elephant and mouse flows are scheduled in a same DCN with different scheduling schemes. The DCN topology deployed in this experiment is a Clos network with 2 and 4 leaf and spine switches respectively. We generate elephant and mouse flows to leaf switches with average sizes of 100KB and 1KB, respectively. The queue lengths of the switch ports are used as performance indicators.

Since the scale of the DCN in the simulation is not very large, HOLMES deploys the static policy that partitions the two sub-networks in advance. The buffer sizes of all the switch ports are set to be the same. When the buffer of a switch port is full, all the upcoming input packets to that port will be dropped. We compare HOLMES against two start-of-the-art unified load balancing schemes: CONGA [10] and queue length gradient based scheduling. Similar to the delay gradient based congestion control policy used in [16], we deploy the queue length gradient as the indicator and schedule the arrived packet to the port with the minimum length gradient.

Figs. 3A-3C show the queue length variation of the four ports of a spine switch under the three scheduling schemes. The Xaxis indicates the time period and the Y-axis denotes the queue length. We can see from Fig. 3A that using CONGA, the buffers of all the four ports are full after a period of time, indicating the throughput of the switch has been maximized, which benefits the transmission of the elephant flows. However, when a mouse flow arrives, all the packets in that flow have to wait for a long queuing time since all the output port are of heavy loads. Consequently, the latency of the mouse flow will increase and degrade the overall performance of the hybrid DC flows.

Similarly, as shown in Fig. 3B, the buffers of the four output ports are also almost full after a period of time using the length gradient based policy. The results indicate that the length gradient based load balancing policy still suffers from the interference between the elephant flows and the mouse flows.

Comparing Fig. 3A and Fig. 3B, we find that the load balancing condition of the gradient based scheme is a little worse than CONGA. The reason is that the gradient based scheme schedules the DC flows according to the changing trend but not the current state of the DCN. Finally, Fig. 3C shows that HOLMES has successfully isolated the elephant and mouse flows. Two of the ports have been partitioned to the low latency sub-network and used for transmitting the mouse flows. Fig. 3C shows that the buffers of the two ports are almost empty during the entire transmission procedure. Thus, packets in the mouse flows do not need to wait for additional queuing delays, and the low latency of the mouse flow is ensured. Moreover, the buffers of the other high throughput ports are also full filled, which satisfies the throughput requirements of elephant flows. Hence, by isolating the mixed traffic, HOLMES network partition policy successfully eliminated the interference of the elephant flows to the mouse flows.

The main shortcoming of the network partition solution is the inefficient use of network resources. Although the isolation of the hybrid traffic avoids the interactions of the elephant and mouse flows, the spared network resource in the low latency paths has not been fully used since the buffers of these paths are almost empty. An effective solution is to improve the buffer allocation by limiting the buffer size of the low latency sub-network and assigning the spared buffers to the high throughput sub-network.

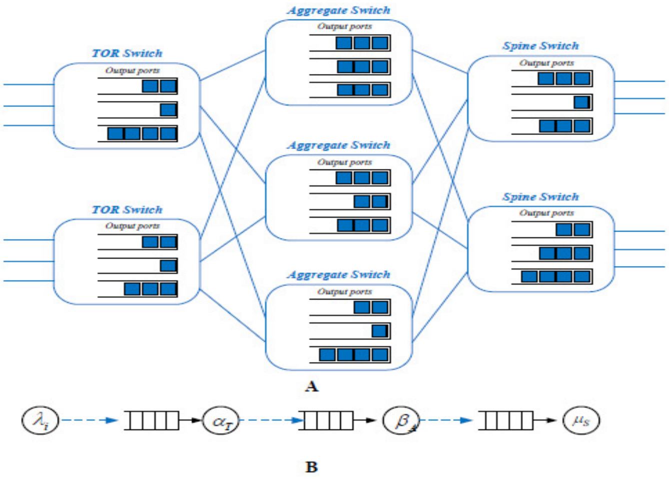
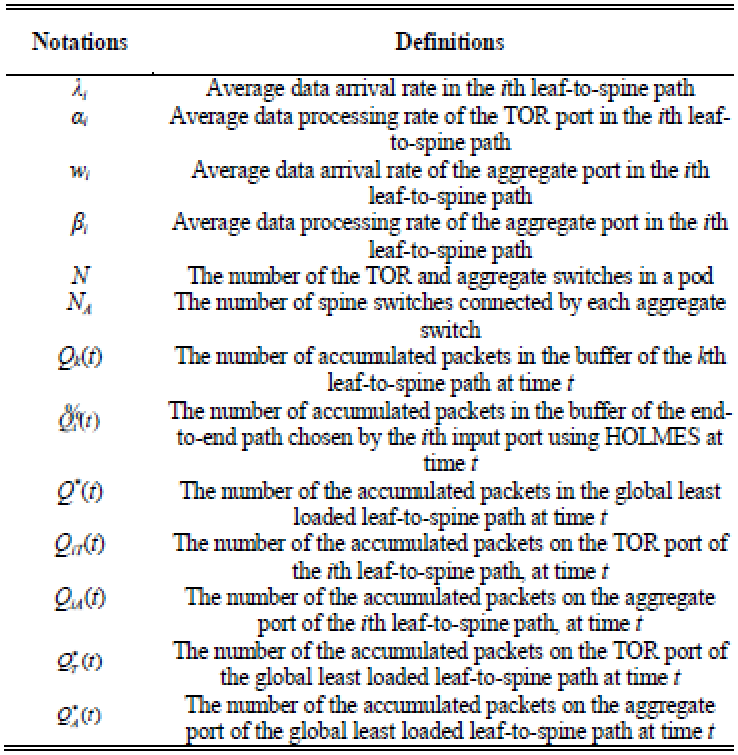


Fig. 2. Abstraction of a leaf-to-spine path in a Clos network (A) to a serial queuing system (B)

Table 2. Summary of Key Notations and Definitions in Scheduling Model



Notations	Definitions
λ_i	Average data arrival rate in the *i*th leaf-to-spine path
α_i	Average data processing rate of the TOR port in the *i*th leaf-to-spine path
w_i	Average data arrival rate of the aggregate port in the *i*th leaf-to-spine path
β_i	Average data processing rate of the aggregate port in the *i*th leaf-to-spine path
N	The number of the TOR and aggregate switches in a pod
N_A	The number of spine switches connected by each aggregate switch
$Q_k(t)$	The number of accumulated packets in the buffer of the *k*th leaf-to-spine path at time *t*
$Q_i^{\%}(t)$	The number of accumulated packets in the buffer of the end-to-end path chosen by the *i*th input port using HOLMES at time *t*
$Q^*(t)$	The number of the accumulated packets in the global least loaded leaf-to-spine path at time *t*
$Q_{iT}(t)$	The number of the accumulated packets on the TOR port of the *i*th leaf-to-spine path, at time *t*
$Q_{iA}(t)$	The number of the accumulated packets on the aggregate port of the *i*th leaf-to-spine path, at time *t*
$Q_T^*(t)$	The number of the accumulated packets on the TOR port of the global least loaded leaf-to-spine path at time *t*
$Q_A^*(t)$	The number of the accumulated packets on the aggregate port of the global least loaded leaf-to-spine path at time *t*

B. Stability Validation of HOLMES Scheduling Algorithm

We evaluate the stability of HOLMES flow scheduling algorithm. We simulate the scenario that DC traffic are scheduled in a DC with asymmetric network topology. We combine two different sized Clos networks, and construct an asymmetric DCN architecture. One of the Clos network consists of 2 leaf switches and 4 spine switches. The other is a Clos network with 5 and 4 leaf and spine switches, respectively. 10 hosts are attached to each leaf switch. We concentrate on validating the stability of HOLMES flow scheduling algorithm, rather than the network partition mechanism in this scenario. Thus, we do not deploy the HOLMES network partition mechanism in the experiment and only execute the HOLMES flow scheduling algorithm.

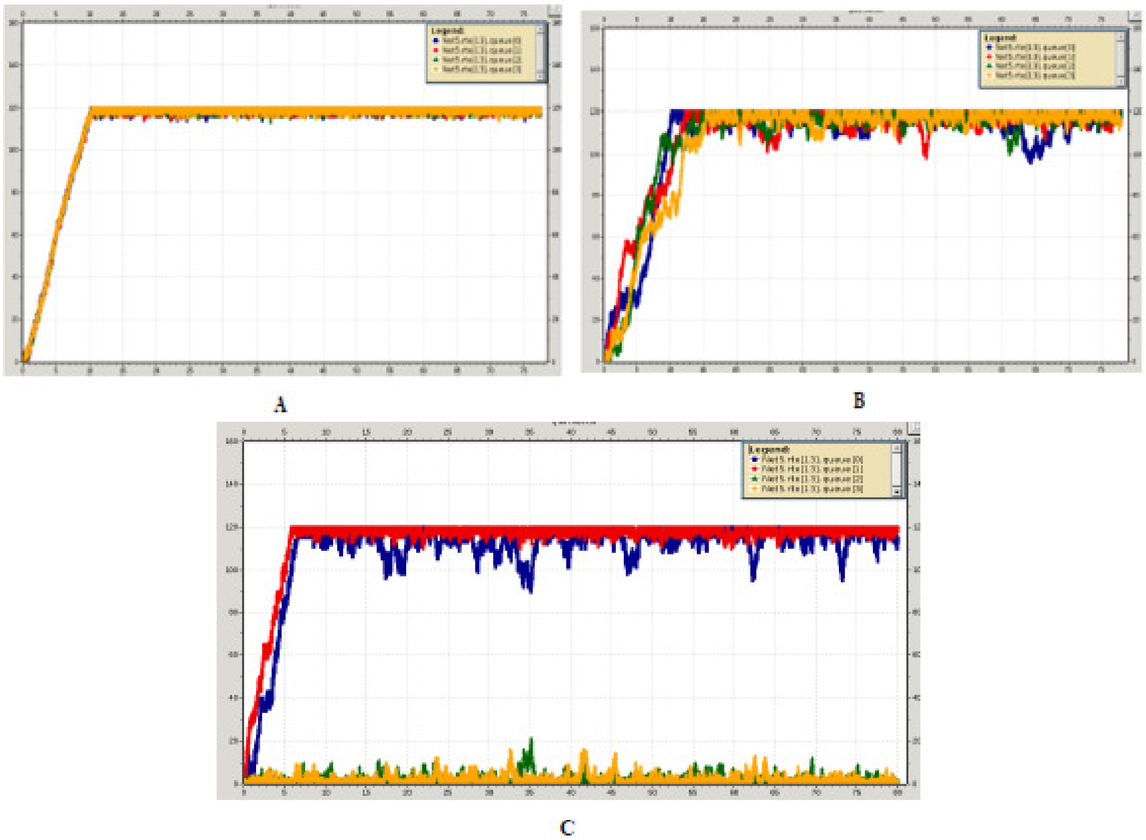


Fig. 3. Queue length changing trends of a DCN spine node's ports under policies CONGA (A), length gradient based policy (B) and HOLMES (C)

CONGA [10] and DRILL [13] are two load balancing solutions proven to be stable. Therefore, we compare them against HOLMES. All DC traffic is scheduled with the granularity of packet, and we focus on analyzing the stability of the three scheduling algorithms. When using the Power of Two Choices selections, we uniformly set d=2 and m=1. Similarly to the previous experiments, we deploy queue length as the load balance indicator to evaluate the overall load balancing condition of the DCN.

Figs. 4A-4C show the queue length changing trend of a specific leaf switch's ports, under load balancing policies CONGA, DRILL and HOLMES. We can see from Fig. 4A that the queue length changing trends of all the ports in a leaf switch are almost overlapped under CONGA, indicating that the queue lengths of all the switch ports are almost the same at each time unit. Therefore, the load balancing condition under CONGA is optimal among all the three mechanisms, since CONGA makes each scheduling decision based on the global congestion information. Without considering the time used for obtaining congestion information, CONGA obtains the global optimal load balancing result.

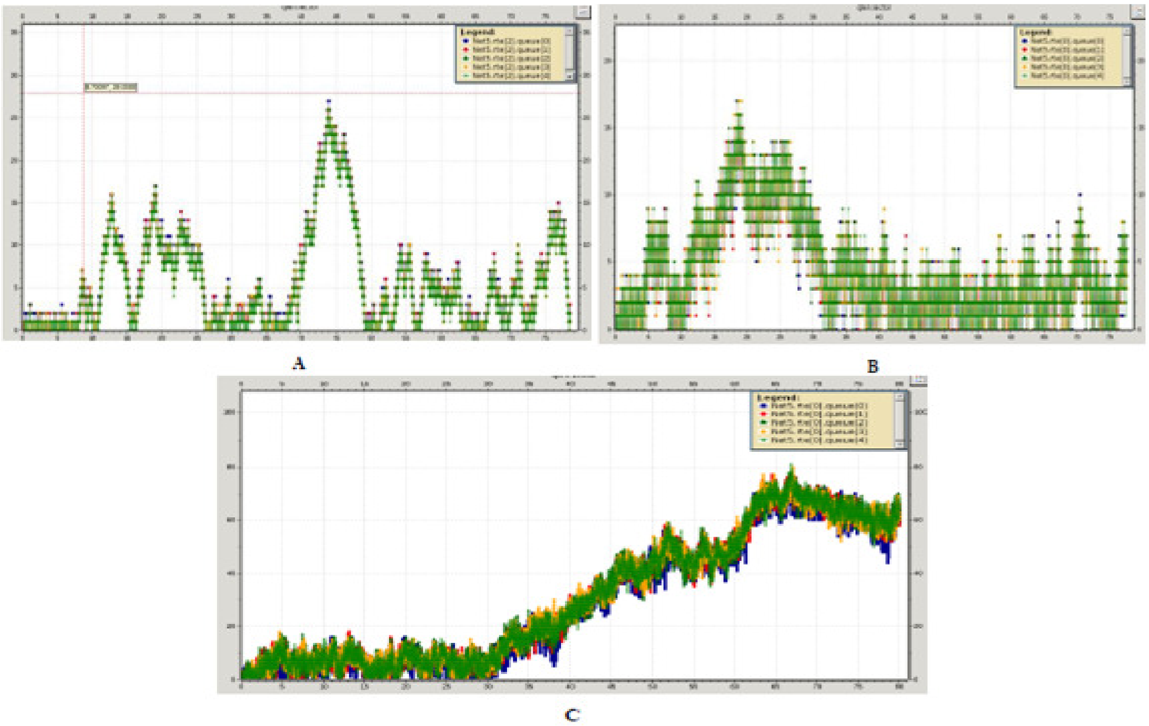


Fig. 4. Queue length changing trends of a DCN leaf node's ports under load balancing policies CONGA (A), DRILL (B) and HOLMES (C)

Fig. 4B shows the queue length changing trends of the same leaf switch ports under DRILL. Different with the former results, we find fluctuations in the queue length changing curve. In other words, the length difference of the longest queue and the shortest queue is clear. The reason is that the use of (d, m) policy in DRILL reduces the scale of the solution space, and the local optimal solutions affect the load balancing condition of the DCN.

Fig. 4C shows the queue length changing trends of the same leaf switch ports under HOLMES scheduling algorithm. The fluctuations also exist in the curve of Fig. 4C, where the amplitude of the fluctuation is more obvious. This phenomenon is also caused by the use of (d, m) policy. Compared with DRILL, the global (d, m) policy used in HOLMES further limits the solution space, and exacerbates the fluctuations. However, although the fluctuations are more obvious when executing HOLMES flow scheduling algorithm, we can also find an upper bound (about 10 packets) of the fluctuation amplitude, indicating that the length difference of the shortest and the longest queue is not infinite in HOLMES. Hence, our HOLMES flow scheduling algorithm is stable during the whole scheduling period. Moreover, limiting the solution exploration space reduces the time used to obtain the congestion information and make HOLMES more efficient and applicable for large-scale data centers.

C. Adaptability for Heterogeneous

As discussed earlier, both the stochastic flow scheduling algorithm and the weighted traffic splitting solutions show good adaptability of heterogeneous congestion states. We now evaluate the adaptability of the two solutions.

We first theoretically compare the approximate adaptability of the two solutions. Consider a simple leaf-spine DCN topology with N paths available between two racks. The loads of the N paths (number of packets accumulated in the buffers) are unevenly distributed. We compare the adaptability of the two mechanisms under the linear distribution of path loads.

For analytical tractability, we assume that the amounts of packets accumulated in the buffer of each path are: 0, 1, 2…, N-1 and the buffer size of each path is also set to N. Next, we calculate the probabilities that one arrived packet is scheduled to the least loaded path under the (2, 1) policy as well as the weighted traffic splitting policy.

The probability P1 that a packet is scheduled to the least loaded path under the (d, m) policy (d =2, m =1) can be calculated as:

$$P_1 = C_{N-1}^{d-1} / C_N^d = \frac{d}{N} = \frac{2}{N} \quad (d = 2)$$

When deploying the weighted traffic splitting solution, the probability that one packet is assigned to a specific path is proportion to the available capacity of that path. Then, the probability that the packet chooses the least loaded path is:

$$P_2 = \frac{N}{N + (N-1) + (N-2) + \ldots + 1} = N / C_{N+1}^2 = \frac{2}{N+1}$$

When the DCN scale is huge, the value of N is large enough, and then the value of P1 and P2 are almost equal. Thus, the adaptability of the (2, 1) policy and the weighted traffic splitting policy for heterogeneous are almost the same when the path loads satisfy linear distributions. However, compared with the weighted solution that maintains the load status of all the N paths, the storage complexity is dramatically reduced when using the stochastic flow scheduling mechanisms.

Next, we consider another scenario that the load distribution of the DCN paths shows stronger heterogeneity. Suppose the path loads satisfy exponential distribution and the amounts of spared buffers in N paths are:

$$N, \frac{N}{2}, \frac{N}{4}, \frac{N}{2^3}, \ldots, \frac{N}{2^{N-1}}$$

Using the weighted traffic splitting policy, the probability that a packet chooses the least loaded path is:

$$P_2 = \frac{N}{N + \frac{N}{2} + \frac{N}{4} + \frac{N}{2^3} + \ldots + \frac{N}{2^{N-1}}} = \frac{N}{2N - \frac{N}{2^{N-1}}} = \frac{1}{2 - \frac{1}{2^{N-1}}}$$

On the other hand, when deploying the (d, m) policy, the probability becomes:

$$P_1 = C_{N-1}^{d-1} / C_N^d = \frac{d}{N}$$

When $d = N/2$, we get:

$$P_2 \approx P_1 = \frac{1}{2}$$

Therefore, when the load conditions of the DC paths are heavily heterogeneous, the (d, m) policy also needs to maintain plenty of load status information to keep its adaptability as good as the weighted traffic splitting solutions. The stochastic scheduling mechanism does not show obvious advantages in this scenario.

The evaluations of the two approaches have also been made by some other researchers. Key et al [96] propose a coordinated solution that periodically samples again for new paths via random selection of stepping stone routers at the higher level, shifts the traffic load to the paths with the lowest loss rates (or lowest ECN rates, largest delays), then equates the marginal utility of its aggregate data rate to the loss rate on these "best" paths at the lower level. This approach can be considered as an improvement of the traditional weighted traffic splitting solutions, since it considers the real-time changes of the load states. Moreover, Key et al show that the approach outperforms the Power of Two Choices policy, and does almost as well as if each user saw the global list of the resources.

We simulate the execution process of the coordinate approach as well as the Power-of-Two-Choices algorithm on a same switch. Fig. 5 shows the changing trend of the overall switch load as the modeling factor (d) increases.

The load distribution of the switch ports is initialized exponentially in this experiment. We see from Fig. 6 that, as the value of d increases, the load condition of the switch is improved under the power of two choices policy ((d, m) policy); since a larger value of d increases the probability of choosing the lightest loaded output port. As we increase the value of d from 2 to 5, the power of two choices policy performs almost as well as the theoretical load optimal policy (d = 10), which validates our previous modeling results. On the contrary, when using the coordinated approach, the switch attains optimal performance when the value of d is small (d = 2) and the load state of the switch is almost as good as the theoretical load optimal policy. This simulation result is in accordance with the analysis in the literature. However, as the value of d increases, the load state of the switch becomes worse: when assigning d = 10, the load balancing condition of the switch under the coordinated policy is even worse than the (2, 1) policy.

Although the stochastic flow scheduling outperforms the weighted traffic splitting solution in most cases, it still has some limitations. The weighted traffic splitting solution maintains the load status of all the paths. It dynamically adjusts the value of each weight according to the current load status of each path (the static weight configuration has proven to be not applicable in [10]). However, when deploying the (d, m) policy, the value of d and m are constant after the initializations. Thus, when the values are not appropriately assigned, (d, m) policy will not perform as well as the weighted traffic splitting solutions. Hence, the HOLMES AI module is responsible for analyzing the overall heterogeneity degree of a DCN, and guiding the flow scheduling algorithm to set appropriate values of the algorithm factors (d and m). The detailed design and implementation of the HOLMES AI module is our future work.

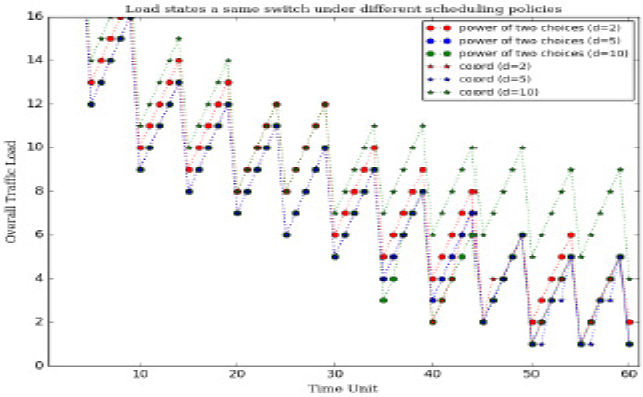


Fig. 5. Changing trend of a switch's overall traffic load under different policies

D. Technical Challenges in Hardware Implementations

Some technical challenges need to be considered to implement HOLMES in real-world data centers. We now summarize these challenges in hardware implementations.

1) Handling the packet reordering

The flow scheduling algorithm in HOLMES can be implemented with different data granularities: per packet scheduling, per flow scheduling or some intermediate data sizes, e.g. flowcell [24], flowlet [10], etc. When using the TCP transmission protocol and implementing the per packet (or flowcell) scheduling, some studies have shown that this finegrained traffic splitting techniques cause packet reordering and lead to severe TCP throughput degradations [23]. Therefore, the packet reordering problem needs to be considered when implementing the fine-grained HOLMES traffic scheduling algorithm. A viable solution is to deploy the JUGGLER network stack in data center traffic transmissions. JUGGLER exploits the small packet delays in datacenter networks and the inherent traffic bursts to eliminate the negative effects of packet reordering while keeping state for only a small number of flows at any given time. This practical reordering network stack is lightweight and can handle the packet reordering efficiently.

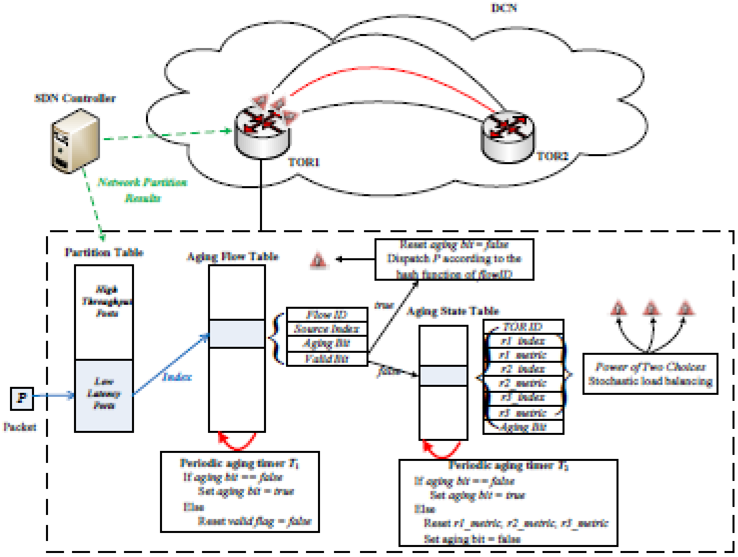


Fig. 6. Overview of HOLMES forwarding and state tables: A new flow table entry is set up by applying the (2, 1) policy in the state table. The period timer of each table is triggered every time period T1 and T2 to age out inactive flows and update the load status of each candidate end-to-end path.

2) Design of DCN forwarding and state tables

The design of the forwarding and state tables is also a noteworthy challenge. An appropriate approach should cope with the small time budget as well as the small register usage. We now propose a viable design to implement the per-flow scheduling algorithms of HOLMES

As shown in Fig. 6, a TOR switch maintains a flow table anda state table. The two tables work together to execute the load mbalancing policy attained from the SDN controller. Specifically,when a packet arrives, its flowID is hashed to map the packet to a flow table entry. If the table entry is valid, the packet is dispatched to the path indicated by the stored hash applied to the packet’s flow ID. On the contrary, if the packet's flowID is not maintained in the flow table, the TOR switch will look up the destination TOR ID in the state table. After that, the (d, m) policy is applied to compare the load states of the three candidate end-to-end paths to the destination TOR (r1_metric, r2 metric and r3_metric), and assign the packet to the optimal path. Two of the three end-to-end paths are randomly selected. The third one is the optimal path from the last selection. Finally, the flow ID and the hash of the chosen path will be inserted into the flow table.

The information of each table needs to be updated periodically to keep the real-time status of the traffic and paths. Thus, we associate an aging bit with each table entry. The aging bit of the flow table is responsible for marking inactive or idle flows: when a packet's flow ID maps the information in the forwarding table, the aging bit is cleared to indicate that the flow entry is active. A timer process visits every table entry every aging timeout T1. When the timer process visits a table entry, it either turns on the aging bit or invalidates the entry if the aging bit is already on. In other words, T1 is the timeout threshold to age out inactive flows, which is proportional to the size of the scheduling unit e.g. per-flow, per-flowcell, etc. If the packet's flow ID is not maintained in the flow table, the TOR switch will execute the (d, m) policy on the state table. Thus, T1 can also be considered as the time period to trigger the execution of the HOLMES scheduling algorithm. On the other hand, another timer process runs together with the aging bit of the state table to update the load status of each candidate end-to-end path. The timeout threshold to age out the old status information in the state table is set to T2. To ensure that the latest load state information is used when executing the (d, m) policy, the value of T1 and T2 should satisfy: $T1 \geq T2$. Moreover, in most cases, the global congestion control signals deployed in the flow scheduling algorithms are the feedback signals from the receivers of the end-to-end paths. Thus, we further get: $T1 \geq T2 \geq RTT$. Key et al have suggested that the policy that periodically sampling a random path and retaining the best paths may perform well.

The periodically sampling of path congestion states in the state table makes the real-time collection of status information becomes a technical challenge. The state collection operations should not introduce obvious transmission overheads and performance penalties. Especially in the TCP incast scenarios where multiple source nodes transmit traffic at the same time to a common destination node, the state collection operations introduce additional traffic and are prone to cause DCN throughput collapse. A viable solution for collecting the real-time congestion status is deploying RepSYN as the signal to detect the load conditions of the multiple paths, as shown in [33]: before transmitting data among multi-rooted paths, multiple TCP connections are established; however, traffic is only transmitted using the first established connection and the other connections are ended immediately. The delay experienced by an SYN reflects the latest congestion condition of the corresponding path, and thus the congestion states can be collected. Moreover, this solution only replicates SYN packets to probe the network, which does not aggravate the TCP incast in a DCN.

The state table only needs to periodically maintain the congestion states of two randomly chosen paths and the congestion-optimal path in the latest time unit. Compared with some other congestion-aware solutions e.g., CONGA [10], RepFlow [32], the storage complexity has been dramatically optimized. In order to make the scheduling results of the (d, m) policy more effective, we choose the disjoint end-to-end paths (paths with different intermediate aggregate or spine switches) to avoid the scenario that the same local congestions is shared by multiple end-to-end paths. This implementation is applicable for more complex multi-tier Leaf-Spine topologies or asymmetric DCN topologies.

3) Dealing with the stale information

When implementing HOLMES scheduling algorithm with packet granularity, the transmission latency of a packet is so small that the information refresh rate in the state table cannot catch up with. Correspondingly, the load balancing algorithm has to use the stale information to make the scheduling decisions. Mitzenmacher et al have pointed out that the delayed information leads to a herd behavior of the scheduling results: data will herd toward a previously light loaded path for much longer time than it takes to fulfill the path. Thus, another technical challenge is to deal with the stale information used in the load balancing algorithms.

To further discuss the effects of the stale information on the execution results of different scheduling policies, we design another experiment that evaluates the load balancing status of a same TOR switch under different state table timers. We simulate a scenario that traffic arrives at a TOR switch with a fixed rate (1 packet/time unit). The TOR switch contains 10 output ports with fixed data transmission rate (1 packet/ 8 time units). Two different timers are deployed to periodically update the information of the TOR state table (state information is updated every 5 or 20 time units respectively). We focus on evaluating the load balancing status of the same TOR switch under three different flow scheduling approaches in this experiment: (2, 1) policy, (5, 1) policy, as well as the deterministic policy (d =N=10 in the definition of the (d, m) policy).

Figs. 7A, 7C and 7E show the changing trend of the TOR load balancing states under the deterministic policy, (2, 1) policy and (5, 1) policy respectively, when the state table is updated every 5 time units. As comparisons, Fig. 7B, 7D and 7F show the performance of the three policies with the state table updated every 20 time units. We analyze the effects of the stale information age to different scheduling policies.

Figs. 7A and 7B show the execution results of the deterministic scheduling policy when the refresh period of the state table is set to 5 or 20 time units respectively. We find from the two figures that, although the deterministic policy provides optimal performance using the perfect run-time state information, it suffers from the effects of the stale information seriously. We see from Fig. 7A that even if the update period is set short (5 time units), the load states of the TOR ports show obvious herd phenomena: traffic will herd toward a previously light loaded port first; the next refresh of the state table will put the port high up the load list and the port will become light loaded again; then the path will be heavy loaded again after several time periods as the next herd sees that it is light loaded. From Fig. 7A we see that each TOR port's load condition shows obvious periodically changing trend (heavy loaded, light loaded, heavy loaded, and so on), and this herd phenomenon is more obvious as the age of the stale state information increases. As shown in Fig. 7B, the amplitude of the fluctuation of each port's load curve is impressively obvious compared with Fig. 7A: the maximum load of a port has increased from 5 packets to more than 15 packets. Compared with the other two approaches (shown in Fig. 7D and 7F), the load balancing condition under the deterministic policy becomes the worst as the age of the stale information increases. Thus, adaptability of the deterministic policy to the stale information is poor.

Theoretically, the load balancing condition of the TOR switch under (5, 1) policy is in second place among all the three approaches, when deploying the perfect run-time state information. Clearly, deploying a larger value of d in the (d, m) policy increases the probability of choosing the light loaded output port. However, compare Fig. 7E with Fig. 7C, we find that the TOR's load balancing condition under (5, 1) policy (shown in Fig. 7E) does not outperforms the (2, 1) policy (shown in Fig. 7C) when using the short aged stale information. Moreover, compared with Fig.7D and 7F, we find that as the age of the stale information increases, (2, 1) policy gradually outperforms (5, 1) policy. The experimental results reflect an issue that compared with the probability of choosing the optimal path, the adaptability for herd behavior becomes the dominant factor for performance optimization.

Comparing Fig. 7C with 7D, we see that increasing the age of the stale information does not affect the execution results of (2, 1) policy: the maximum load of a TOR port is still maintained less than 5 packets when the information update period increases from 5 time units to 20 time units. Comparing

Figs. 7B, 7D and 7F, we also find that (2, 1) policy performs best among all the three approaches when using the long aged stale information. The experimental results indicate that (2, 1) policy provides optimal adaptability for HOLMES with the stale information, especially when the age of the information is large. The main reason is that (2, 1) policy combines the advantages of both worlds: It deploys real load information to schedule traffic, yet rejects the herd behavior much more strongly than the other two approaches.

Besides adjusting the value of modeling factors, e.g. d, m, another applicable solution is to deploy the machine learning methods to predict the current load state based on the historical stale information. Fox et al [95] deploy queue length as the metric for load balancing and notice rapid oscillations in queue lengths, caused by the stale information. To overcome this limitation, they further change the manager stub to keep a running estimate of the change in queue lengths between successive reports and show that these estimations are sufficient to eliminate the oscillations. Dahlin [98] also estimates the job arrival rates and the age of the stale information, and proposes corresponding algorithms to balance the traffic load in a distributed system. Similar approaches can also be deployed in HOLMES, which estimate the current load states of different paths by analyzing the successive historical state information. This function will be realized in the HOLMES AI module, and will be another one of our future works.

Overall, the simulation experimental results validate the modeling results in Section IV. They show that HOLMES load balancing algorithm is stable and adaptable in heterogeneous DCNs.

6. CONCLUSION

This paper presents HOLMES, a novel DCN flow scheduling scheme, which tackles mixed (mice vs. elephants) data center traffic. Using a stochastic performance model, we first prove the necessity of isolating mice and elephants with a closed form. We then present the HOLMES architecture that partitions a DCN into high-throughput and low-latency sub-networks. We further design a stochastic and global congestion-aware load balancing algorithm that schedules the corresponding DC traffic to each sub-network. Simulation results show that HOLMES network partition policy can successfully eliminate the interference between the mouse and elephant data flows. Finally, we prove that HOLMES flow scheduling algorithm is stable and scalable for large-scale data centers.

REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” In ACM SIGCOMM Computer Communication Review, 2008, 38(4): 63-74.

[2] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, P. Patel, and S. Sengupta. “VL2: A Scalable and Flexible Data Center Network,” In ACM SIGCOMM Computer Communication Review, 2009, 39(4): 51-62.

[3] C. Hopps. Analysis of an Equal-Cost Multi-Path algorithm. RFC 2992, Network Working Group, 2000.

[4] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang and A. Vahdat, “Hedera: Dynamic flow scheduling for data center networks,” In Proc. USENIX NSDI, 2010, Vol. 10, pp. 19-19.

[5] J. Cao, R. Xia, P. Yang, C. Guo, G. Lu, L. Yuan, Y. Zheng, H. Wu, Y. Xiong, and D. Maltz, “Per-packet load-balanced, low-latency routing for Clos-based data center networks,” In Proc. ACM CoNEXT, 2013, pp. 49-60.

[6] A. Dixit, P. Prakash, Y. C. Hu, and R. R. Kompella, “On the impact of packet spraying in data center networks,” In Proc. IEEE INFOCOM, 2013, pp. 2013-2018.

[7] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, “Data center TCP (DCTCP),” In ACM SIGCOMM Computer Communication Review, 2010, 40(4): 63-74.

[8] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. M. B. Prabhakar, and S. Shenker, “pFabric: Minimal near-optimal datacenter transport,” In ACM SIGCOMM Computer Communication Review, 2013, 43(4): 435-446.

[9] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. Katz, “DeTail: Reducing the flow completion time tail in datacenter networks,” In ACM SIGCOMM Computer Communication Review, 2012, 42(4): 139-150.

[10] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, and G. Varghese, "CONGA: Distributed congestion-aware load balancing for datacenters," In ACM SIGCOMM Computer Communication Review, 2014, 44(4): 503-514.

[11] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat, "WCMP: Weighted cost multipathing for improved fairness in data centers," In In Proc. ACM EuroSys, 2014.

[12] P. Wang, H. Xu, Z. Niu and D. Han, "Expeditus: Distributed congestion-aware load balancing in clos data center networks," In Proc. ACM CoNEXT on Student Workshop, 2014, pp. 1-3.

[13] S. Ghorbani, B. Godfrey, Y. Ganjali and A. Firoozshahian, "Micro load balancing in data centers with DRILL," In Proc. ACM HotNets, 2015, pp. 17.

[14] N. Katta, M. Hira, C. Kim, A. Sivaraman and J. Rexford, "HULA: Scalable load balancing using programmable data planes," In Proc. ACM SOSR, 2016.

[15] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," In IEEE Transactions on Services Computing, 2010, 3(4): 266–278.

[16] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall and D. Zats, "TIMELY: RTT-based congestion control for the datacenter," In ACM SIGCOMM Computer Communication Review, 2015, 45(4): 537-550.

[17] D. A. Hayes and G. Armitage, "Revisiting TCP congestion control using delay gradients," In Proc. IFIP Networking, 2011.

[18] W. Wang, Y. Sun, K. Salamatian and Z. Li, "Adaptive path isolation for elephant and mouse flows by exploiting path diversity in datacenters," In IEEE Transactions on Network and Service Management, 2016, 13(1): 5-18.

[19] V. Liu, D. Halperin, A. Krishnamurthy and T. Anderson, "F10: a fault-tolerant engineered network," In Proc. USENIX OSDI, 2010.

[20] J. Kim, W. J. Dally, S. Scott and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," In ACM SIGARCH Computer Architecture News, 2007, 35(2): 126-137.

[21] J. H. Ahn, N. Binkert, A. Davis, M. McLaren and R. S. Schreiber, "HyperX: topology, routing and packaging of efficient large-scale networks," In Proc. ACM SC, 2009, p. 41.

[22] J. Kim, W. J. Dally, S. Scott and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," In ACM SIGARCH Computer Architecture News, 2008, 36(3): 77-88.

[23] A. Dixit, P. Prakash and R. R. Kompella, "On the Efficacy of Fine-Grained Traffic Splitting Protocols in Data Center Networks," In ACM SIGCOMM Computer Communication Review, 2011, 41(4): 430-431.

[24] K. He, E. Rozner, K. Agarwal, W. Felter, J. Carter, and A. Akella, "Presto: Edge-based load balancing for fast datacenter networks," In ACM SIGCOMM Computer Communication Review, 2015, 45(4): 465-478.

[25] S. Sen, D. Shue, S. Ihm, and M. J. Freedman, "Scalable, optimal flow routing in datacenters via local link balancing," In Proc. ACM CoNEXT, 2013, pp. 151-162

[26] T. Edsall, A. Fingerhut, T. Lam, R. Pan, and G. Varghese, "Flame: Efficient an robust hardware load balancing for data center routers," [Department of Computer Science and Engineering], University of California, San Diego, 2012.

[27] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," In ACM SIGCOMM Computer Communication Review, 2013, 43(4): 15-26.

[28] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," In ACM SIGCOMM Computer Communication Review, 2013, 43(4): 3-14.

[29] T. Benson, A. Anand, A. Akella, and M. Zhang, "Microte: Fine grained traffic engineering for data centers," In Proc. ACM CoNEXT, 2011, pp. 8:1-8:12.

[30] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," In ACM SIGCOMM Computer Communication Review, 2005, 35(4): 253-264.

[31] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "Mate: Mpls adaptive traffic engineering," In Proc. IEEE INFOCOM, 2001, pp. 1300-1309.

[32] H. Xu and B. Li, "RepFlow: Minimizing flow completion times with replicated flows in data centers," In Proc. IEEE INFOCOM, 2014, pp. 1581-1589.

[33] S. Liu, W. Bai, H. Xu, K. Chen and Z. Cai "RepNet: Cutting tail latency in data center networks with flow replication," In Computer Science, 2014.

[34] R. Gallager, "A minimum delay routing algorithm using distributed computation," In IEEE Transactions on Communications, 1977, 25:73-85.

[35] N. Michael and A. Tang, "Halo: Hop-by-hop adaptive link-state optimal routing," In IEEE/ACM Transactions on Networking, 2015, 23(6): 1862-1875.

[36] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic load balancing without packet reordering," In ACM SIGCOMM Computer Communication Review, 2007, 37(2): 51-62.

[37] L. G. Valiant and G. J. Brebner, "Universal schemes for parallel communication," In Proc. ACM STOC, 1981, pp. 263-277.

[38] R. Zhang and N. McKeown, "Designing a fault-tolerant network using valiant load balancing," In Proc. IEEE INFOCOM, 2008.

[39] R. Zhang and N. McKeown, "Designing a predictable internet backbone with valiant load-balancing," In Proc. IEEE/ACM IWQoS, 2005, pp. 178-192.

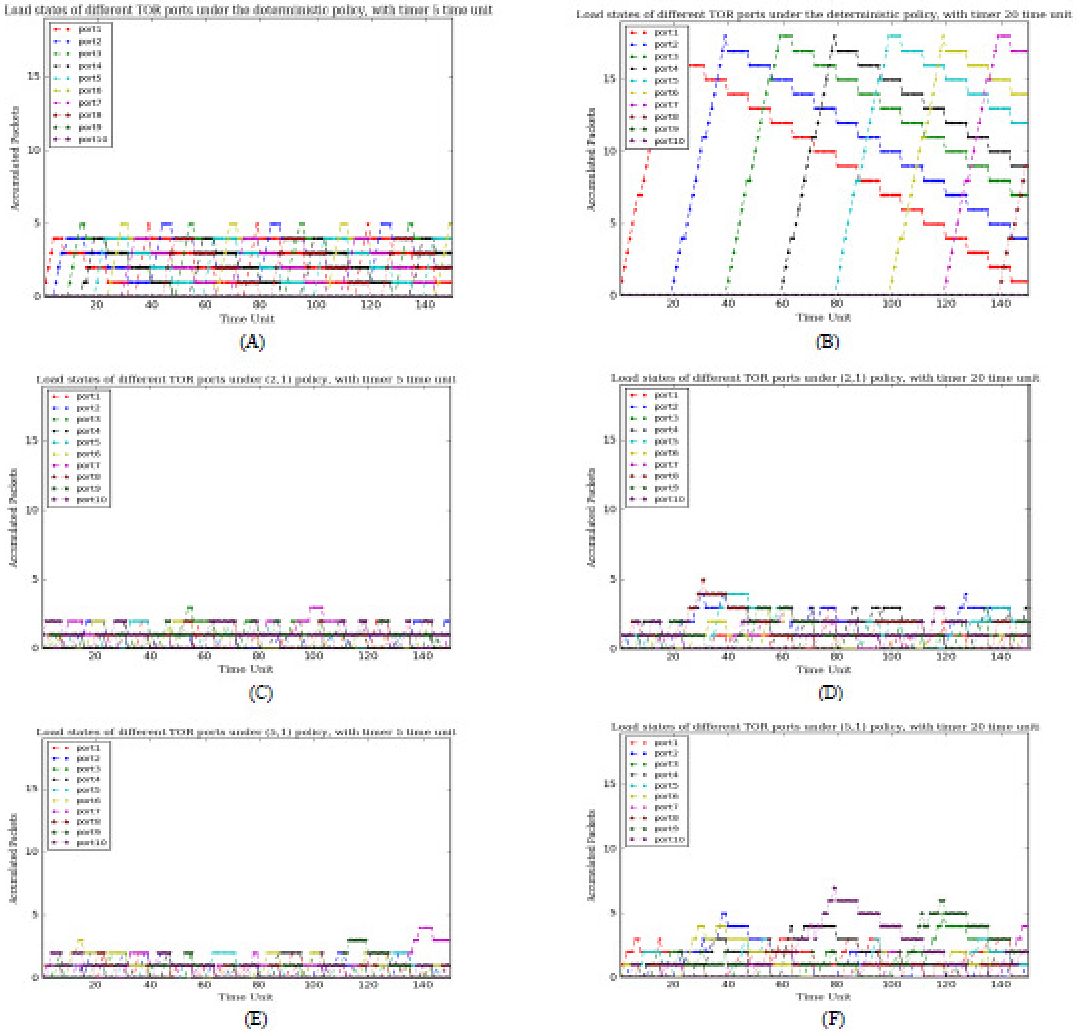
[40] D. Wischik, C. Raiciu, A. Greenhalgh , and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," In Proc. USENIX NSDI, 2011, Vol. 11, pp. 8-8.

[41] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath tcp," In ACM SIGCOMM Computer Communication Review, 2008, 38(4): 63-74.

[42] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal, "Fastpass: A centralized zero-queue datacenter network," In ACM SIGCOMM Computer Communication Review, 2014, 44(4): 307-318.

[43] M. Mizenmacher, "The power of two choices in randomized load balancing," In IEEE Transactions on Parallel and Distributed Systems, 2001, 12(10): 1094-1104.

[44] Y. T. He and D. G. Down, "Limited choice and locality considerations for load balancing," In Performance Evaluation, 2008, 65(9): 670-687.



(A)

(B)

(C)

(D)

(E)

(F)

Fig. 7. Load states changing trend of 10 TOR ports under deterministic policy (A, B), (2, 1) policy (C, D) and (5, 1) policy (E, F) with different state table timer (T2=5, T2=20) respectively. The 10 ports are located on the same TOR switch.

INTENTIONAL BLANK

PERFORMANCE ANALYSIS OF SEVERAL SCHEDULING ALGORITHMS IN NAKAGAMI-M FADING CHANNELS

Hakkı Soy

Department of Electrical and Electronics Engineering,
Necmettin Erbakan University, Konya, Turkey

ABSTRACT

This study investigates the performances of several scheduling algorithms (Round Robin, maximum SNR and opportunistic Round Robin) in the Medium Access Control (MAC) layer of wireless network with cellular architecture. The spectral efficiency and fairness are two important metrics to evaluate the performance of wireless systems. The Nakagami-m distribution is a generalized way to model small scale fading and it can model various wireless fading channel conditions by changing the value of m parameter. The channel is assumed to be slow-fading and opportunistic beamforming scheme is used to enlarge the rate and dynamic range of channel fluctuations through the employment of multiple antennas at the base station. The MATLAB based Monte Carlo simulations are implemented to show the effects of the antenna number and m parameter of the Nakagami distribution. The obtained results show that the pure opportunistic max-SNR algorithm has a spectral efficiency advantage, even though it has a bad fairness characteristic in high number of users.

KEYWORDS

Spectral Efficiency, Fairness, Scheduling, Opportunistic Beamforming, Wireless Cellular Network

1. INTRODUCTION

Wireless networks have been one of the fastest growing segments in the telecommunications industry, ever since the first one was introduced in 1970. Recently, the fast evolution of wireless technologies has enabled the high-speed data services and the demand for higher data rates is increasing exponentially with each passing day [1]. Today, the most important challenge for wireless systems is the limited availability of the radio frequency spectrum. Therefore, the spectral efficiency is a primary consideration in the design of future wireless communications systems. Over the last decades, many researchers have looked at various methods to improve the spectral efficiency of wireless systems [2, 3].

There are two types of wireless network architectures, cellular and ad hoc. The cellular architecture has evolved from mobile telephone systems and it has several potential advantages for future wireless networks because of its ability to operate using small transmission powers and

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP - 2018
pp. 23–32, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80902

to increase the spatial reuse in the network. Besides, the ad hoc network is a noninfrastructure architecture, in which users communicate with each other through several hops of short distance transmissions. The ad hoc model is specifically designed to support the military and emergency operations [4, 5].

The cellular networks are organized with a base station (BS) which is responsible for controlling the access to the wireless channel and forwarding data to the intended user. The BS provides the connectivity among the users to efficiently and fairly dispense the available bandwidth. High spectral efficiency of a wireless cellular network can be achieved by allocating the system resources to the users opportunistically. The scheduling is a method that enables multiple users to share the system resources (time-slots, transmit power, bandwidth, etc.). In BS, the scheduler makes a decision on which users it is going to serve during the next transmission period. The scheduling algorithm is said to be opportunistic when it takes the channel quality into consideration before it takes a scheduling decision [6-8].

The key characteristic of the wireless channel is the fading due to the multipath propagation of the transmitted signal, which creates replicas of the transmitted signal that arrive at the receiver with different delays [9]. Due to the fading effect, the channel conditions of users have timevarying behaviour at a certain time and each user has different channel statistics. Therefore, different users experience different channel gains at a given time and this effect is called as multiuser diversity [10]. The utilization of multiuser diversity in a wireless cellular network can increase the information theoretic capacity by allowing opportunistic usage of the channel [11].

The statistical description of the radio channels is essential for wireless communication systems. There are several existing statistical channel models are used to in the literature describe the statistics of the multipath signals [12]. Rayleigh fading channel is an excellent approximation in a large number of practical scenarios when there is non-line of sight (NLOS) component present in received signal. Rician fading is suitable for the communication environment with LOS components. But, Nakagami-m fading offers more versatile channel model due to advantage of greater flexibility and accuracy in matching some experimental data than the Rayleigh and Rician channel models [13].

The multiuser diversity gains depend on the tail of the fading distribution and increases with the dynamic range of the channel fluctuations. But, in little scattering and/or slow fading environments, the dynamic range of channel fluctuations is very small. In these cases, a practical solution can be provided with the employment of multiple antennas at the BS to induce large and fast channel fluctuations artificially in the downlink channel. Such a beamforming scheme is called opportunistic beamforming since the BS randomly varies the amplitude and phase values of transmitted signals from the antennas at each time slot. So, the multiuser diversity can still be exploited [14].

This paper focus on the spectral efficiency of wireless links and the fairness of channel access in a cellular network by considering Nakagami-m fading as a model for the channel gains of users. The spectral efficiency is defined as the average transmitted data rate per unit bandwidth for a specified average transmit power and bit error rate (BER). The fairness refers to the equal allocation of network resources among the various users. The aim is demonstration of the effect of the different fading conditions over the spectral efficiency and fairness performances.

2. SYSTEM MODEL

The system of interest is a single cell of the cellular wireless network that consists of multiple users. The BS coordinates all data transmissions within its coverage range. The BS has access to the downlink channel, while uplink channel is shared among users. The downlink system model of the considered network setup is shown in Figure 1. The BS is equipped with M antennas and serving K users whereas each user is equipped with single antenna. The channels between the transmit antennas and the users experience fading paths which are modelled as i.i.d. Nakagamim fading channels. The channel between the *k*th user and the BS is denoted by M × 1 vector

$\mathbf{h}_k = [\ h_{k,1}\ \ h_{k,2} \ldots h_{k,M}\ ]^T$

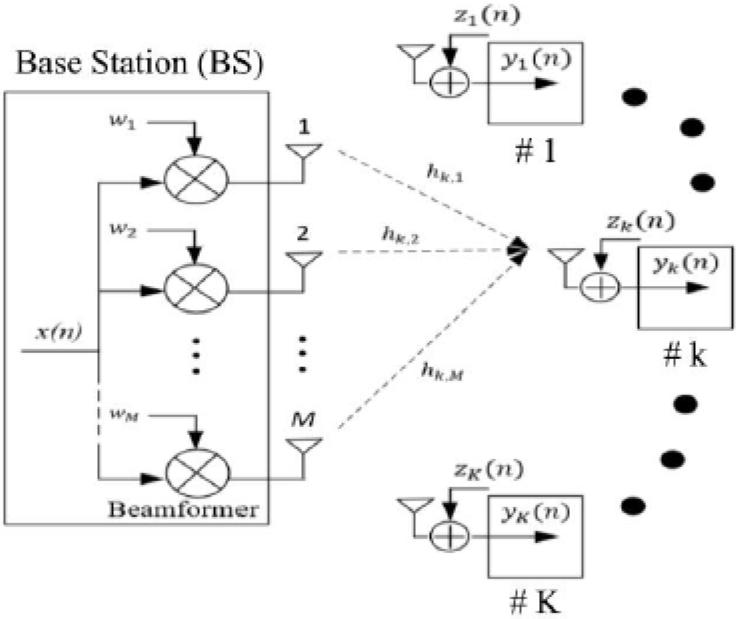


Figure 1. The system model of considered cellular network

The BS forms the beam by choosing the $M \times 1$ random beamforming vector **w** whose distribution is identical to the distribution of $\mathbf{h}_k$ but normalized to keep the transmit power fixed, $\mathbf{w} \sim \mathbf{h} / \|\mathbf{h}\|$. The pilot signal $x\ (n)$ with power E [$x^2\ (n)$] = P_t is transmitted from the BS to the users. The received signal y_k (n) at the kth user may be written as

$$y_k(n) = \sqrt{P_t}(\mathbf{w}^H\mathbf{h}_k)x(n) + z_k(n) \qquad (1)$$

where $z_k\ (n)$ is the circularly symmetric, complex, additive white Gaussian noise (AWGN) with distribution $(0,\sigma^2)$. Note that, by randomly changing the beamforming vector **w** at each time slot, the observed composite channel process of the kth user ($\mathbf{w}^H\mathbf{h}_k$) changes from time-slot to time-slot due to the time-varying beamforming vector.

In order to simplify the analysis, we assume that the channel statistics of all the users are the same and the ratio of the transmit power to the noise variance (P_t / σ^2) is 1. So, without loss of generality, the path loss together with all the other powers is lumped into the channel process.

The quality of wireless links is usually measured by using signal-to-noise ratio (SNR) metric. Thus, with these assumptions, the SNR of the kth user can be written as

$$\gamma_k = \mathbf{w}^{\mathrm{H}}\mathbf{h}_k\mathbf{h}_k^{\mathrm{H}}\mathbf{w} \ . \tag{2}$$

The probability density function (P.D.F.) of the instantaneous SNR is given as

$$f_\gamma(\gamma) = \left(\frac{m}{\gamma}\right)^m \frac{\gamma^{m-1}}{\Gamma(m)} e^{-m\frac{\gamma}{\gamma}} \tag{3}$$

where $\bar{\gamma}$ is the average SNR of the received signal, m called as Nakagami parameter and $\Gamma(.)$ denotes the Gamma function [13]. The phase φ of the Nakagami fading is uniformly distributed over [0, 2π]. Then, the corresponding cumulative distribution function (C.D.F.) is expressed as:

$$F_\gamma(\gamma) = 1 - \frac{\Gamma\left(m, \frac{m}{\gamma}\gamma\right)}{\Gamma(m)} \tag{4}$$

where $\Gamma(.,.)$ denotes the lower incomplete gamma function [13].

The Nakagami-m distribution is a generalized distribution that can be used to model fading environments by changing the value of *m* parameter. Rayleigh P.D.F. is obtained for *m* = 1 as a special case, one-sided Gaussian P.D.F. is described by *m* = ½ and the non-fading situation corresponds to *m* → ∞. The Nakagami fading model is also able to cover severe (½ ≤ *m* ≤ 1) and weak (m >1) channel conditions than Rayleigh fading model [15].

3. Scheduling Algorithms

In cellular wireless networks, the channel allocation task is organized by a scheduler runs on the BS. The users send request to BS and the requests are processed according to medium access control (MAC) protocol. Clearly, the MAC protocol determines the channel access strategy with predefined scheduling algorithm. Behind the fading channel, another important problem, the spectral efficiency of the wireless channel is also influenced by the packet collision due to the simultaneous transmissions of multiple users. In order to provide a collision-free scheduling scheme, it is assumed that only a single node can use the channel at a certain time-slot. The most of the existing MAC protocols use the time division multiple access (TDMA) mechanism where the time axis is divided into equal-sized time-slots. The scheduling algorithm selects a single user at a certain time-slot.

In general, scheduling algorithms can be classified as opportunistic, semi-opportunistic and nonopportunistic. Opportunistic algorithms maximize the spectral efficiency at the expense of full feedback from all the users. The BS requires the channel state information (CSI) from all the users to make a scheduling decision on which users it is going to serve during the next time-slot. The downlink channels can be easily estimated by users via pilot signal broadcast, but uplink channels need feedback transmissions from all users. Based on the collected CSI through feedback channel, the BS makes notifies the all users about the assigned user which access the channel alone. But, the excessive feedback overhead increases the delay which degrades the performance of bandwidth-intensive and time-critical applications. Unlike, the nonopportunistic algorithms simply assign the time-slots to users as independent from their instantaneous channel conditions. This type strategy does not require CSI feedback, but it does not offer multiuser diversity gain with spectral efficiency advantage. Between them, the semiopportunistic algorithms remove the feedback overhead with a slight loss in the spectral efficiency.

This study focuses on three scheduling algorithms which have different characteristics. In literature, these scheduling algorithms are named as round robin, maximum SNR and opportunistic round robin. In non-opportunistic Round Robin (RR) algorithm, the time-slots are allocated to the users in a sequential fashion regardless of their channel conditions. The maximum SNR (max-SNR) algorithm always selects the user with the best channel quality in an opportunistic approach. Finally, in opportunistic round robin (ORR) algorithm, the best user among all is chosen for the first time-slot. At the next time-slot, this user taken out of the competition and the best out of the remaining users is selected for channel assignment. This procedure is repeated until the last time-slot in a round, where the latter user is scheduled. The ORR algorithm ensures the constraint that the K number of users should get exactly one timeslot each within the same round as well as the RR algorithm [16].

The spectral efficiency and fairness of wireless cellular networks are depending on the selected scheduling algorithm in MAC layer. The spectral efficiency is a function of the average SNR of users that access to the channel and the fairness is related with the priority of the scheduler. In RR algorithm, the P.D.F. of assigned users is given by equation (3). So, the spectral efficiency Ψ_{RR} is expressed as follows:

$$\Psi_{\mathrm{RR}} = \int_0^\infty \log_2(1+\gamma) f_\gamma(\gamma) d\gamma = \left(\frac{m}{\bar{\gamma}}\right)^m \frac{1}{\Gamma(m)} \int_0^\infty \log_2(1+\gamma)\gamma^{m-1} e^{-m\frac{\gamma}{\bar{\gamma}}} d\gamma \, . \qquad (5)$$

In max-SNR algorithm, the spectral efficiency $\Psi_{\text{max-SNR}}$ can be formulated using a stochastic framework by using the P.D.F. and C.D.F. expressions in equation (3) and (4), respectively:

$$\Psi_{\text{max-SNR}} = \int_0^\infty \log_2(1+\gamma) f_{\gamma^*}(\gamma) d\gamma = K \int_0^\infty \log_2(1+\gamma) F_\gamma^{K-1}(\gamma) f_\gamma(\gamma) d\gamma \, . \qquad (6)$$

In this formulation, the $f_{\gamma^*}(\gamma)$ term denotes the SNR of selected user with maximum channel quality among all users.

On the other hand, in the ORR algorithm, the spectral efficiency Ψ_{ORR} has a more complex formulation due to its iterative form as expressed below:

$$\Psi_{\mathrm{ORR}} = \int_0^\infty \log_2(1+\gamma) f_{\gamma^{\#}}(\gamma) d\gamma = \frac{1}{K} \int_0^\infty \log_2(1+\gamma) \sum_{k=1}^{K} k F_\gamma^{k-1}(\gamma) f_\gamma(\gamma) d\gamma \, . \qquad (7)$$

In this formulation, the $f_{\gamma^*}(\gamma)$ term denotes the SNR of selected user with maximum channel quality among users in still competition with others. The complicated form of the spectral efficiency formulations in equation (5), (6) and (7) makes it difficult to find a closed form solution. These expressions can be calculated numerically and the details of the closed form solutions are beyond the scope of this paper.

Finally, the fairness is defined by how equally the channel assignments are allocated to users. The Jain's fairness index (JFI) is frequently used to measure fairness of different scheduling algorithms in wireless networks. By considering the time-slot allocation JFI is defined as follows:

$$\mathrm{I}_{\mathrm{JFI}} = \frac{\left|\sum_{k=1}^{K} p_k\right|^2}{K\sum_{k=1}^{K} p_k^2}$$

where p_k is the number of assigned time-slots to user k in a round [17].

4. Simulation Results

In this study, the performance of mentioned scheduling algorithms is investigated in terms of spectral efficiency and fairness over generalized Nakagami-m fading channel. The MATLAB software is used to present the statistical (Monte Carlo) simulation results of investigated scheduling algorithm's performances. The obtained results are shown for both 2 × 1 and 4 × 1 antenna configurations over Nakagami-0.6, Nakagami-1 and Nakagami-3 cases separately. In Monte Carlo simulations, the number of iterations is chosen as 1000 to achieve an acceptable convergence and the number of time slots in an iteration is taken as the number of users in the network. It is assumed that the channel is frequency flat, block fading and the channel vector $\boldsymbol{h}_\mathrm{k}$ remains constant during a fixed number of time-slots called one frame and changes between different frames independently. The average SNR is assumed to be the same for all users, while the instantaneous SNR values of users depend on the channel fading realization in each frame.

Fig. 2 shows the spectral efficiency performance of scheduling algorithms in 2 × 1 antenna configuration. The max-SNR algorithm has the best spectral efficiency, while the RR algorithm has the worst. In max-SNR algorithm, the spectral efficiency increases for larger numbers of users. It is meaning that there exists a favorable user whose instantaneous channel vector matches the beamforming vector and who can maximize the received SNR. Note that, in max-SNR algorithm, when the value of m parameter increases, the spectral efficiency performance decreases due to the insufficient multiuser diversity gain. But, the spectral efficiency performance increases with the increasing m values in RR and ORR algorithms. Because, the LOS component plays dominant role for m > 1 and it is possible to get better signal from the BS. The spectral efficiency of RR algorithm is independent from the number of users.

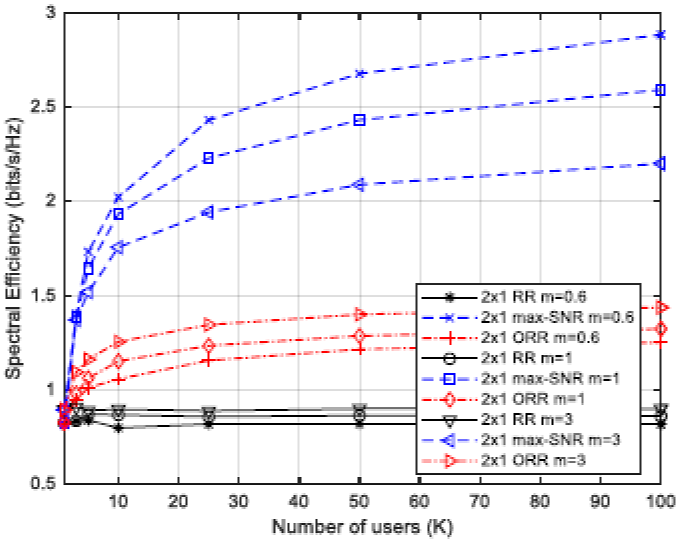


Figure 2. Spectral efficiency of scheduling algorithms in 2 × 1 antenna configuration for different m values

The fairness of scheduling algorithms in 2 × 1 antenna configuration is shown by Fig. 3 for different *m* values. It means that the higher value of JFI implies higher fairness in channel assignment. The RR and ORR algorithms have the best fairness (JFI = 1) and the max-SNR algorithm has the worst fairness performance. It is clearly shown that the fairness of the pure opportunistic max-SNR algorithm decreases with the increasing user numbers. Note also that, the increasing *m* values make a positive effect and slightly improve the fairness of max-SNR algorithm.

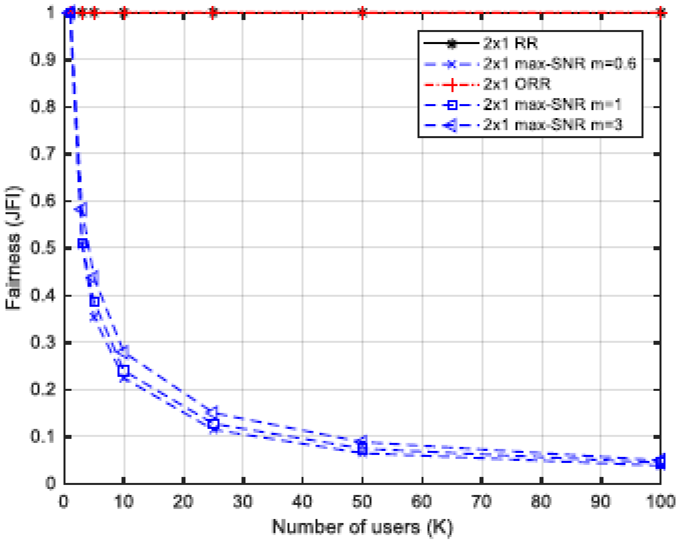


Figure 3. Fairness of scheduling algorithms in 2 × 1 antenna configuration for different m values

In order to assess the effect of adding more antennas at the BS, the spectral efficiency and fairness comparisons are repeated for 4 × 1 antenna configuration. As shown in Fig. 4, evaluated algorithms have similar characteristics as in 2 × 1 setup. It is observed that increasing the number of antennas at the BS does not increase the spectral efficiency in max-SNR algorithm for *m*=1 (Rayleigh channel) and *m*=3 cases. The max-SNR algorithm has better spectral efficiency than 2 × 1 configuration in *m*=0.6 case. Despite that, the employment of more antennas positively affects the ORR algorithm for all *m* values. Fig. 5 shows the fairness performance of scheduling algorithms in 4 × 1 configuration. As seen from this plot, the fairness performance of max-SNR algorithm can be improved by using more antennas at the BS for all m values.

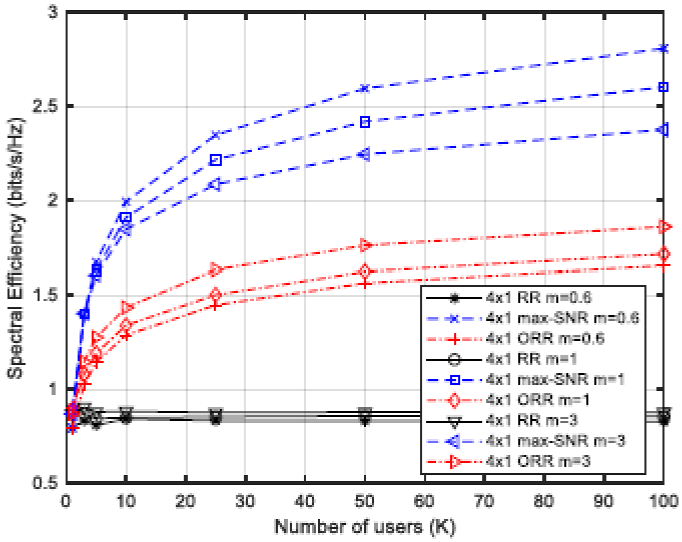


Figure 4. Spectral efficiency of scheduling algorithms in 4 × 1 antenna configuration for different m values

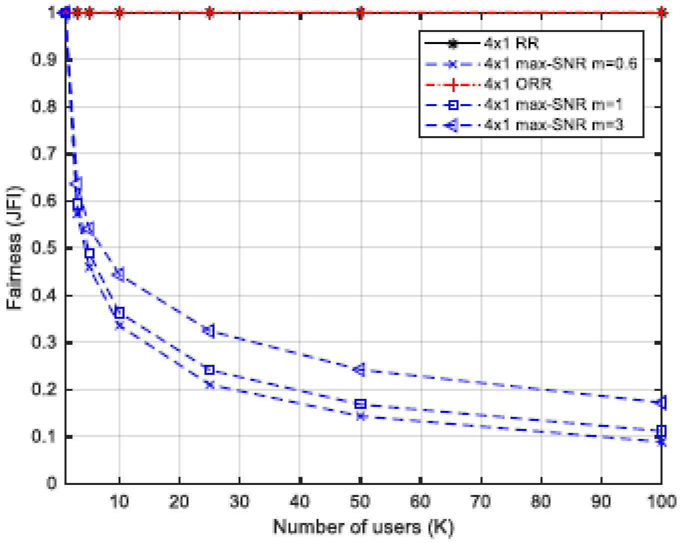


Figure 5. Fairness of scheduling algorithms in 4 × 1 antenna configuration for different m values

5. CONCLUSION

This study presents a performance analysis of the scheduling algorithms for wireless cellular networks in a Nakagami-m fading channel conditions. It was shown that the max-SNR scheduling algorithm provides a better spectral efficiency performance at the cost of unfairness among the users, while the RR and ORR algorithms guarantees the maximum user fairness at the cost of the performance loss in term of spectral efficiency.

REFERENCES

[1] Zeng, M., Annamalai, A., Bhargava, V. K., (1999) “Recent advances in cellular wireless communications”, in IEEE Communications Magazine, vol. 37, no. 9, pp. 128-138.

[2] Miao, G., Song, G., (2015) “Energy and Spectrum Efficient Wireless Network Design”, Cambridge University Press.

[3] Bölcskei, H., Gesbert, D., Papadias, C. B., Van Der Veen, A.-J., (2006) “Space-Time Wireless Systems: From Array Processing to MIMO Communications”, Cambridge University Press.

[4] Rappaport, T. (2002), “Wireless Communications: Principles and Practice”, Prentice Hall PTR.

[5] Wu, T.-Y., Huang, C.-Y., Chao, H.-C., (2003) “A survey of Mobile IP in cellular and Mobile Ad-Hoc Network environments”, Ad Hoc Networks, vol. 3, no. 3, pp. 351-370.

[6] Rasool, J., Hassel, V., de la Kethulle de Ryhove, S., Øien, G. E., (2011), “Opportunistic scheduling policies for improved throughput guarantees in wireless networks”, vol. 43, 18 pages.

[7] Asadi A., Mancuso, V., (2013) “A Survey on Opportunistic Scheduling in Wireless Communications”, in IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 1671-1688.

[8] Liu, X., Chong, E. K. P., Shroff, N. B., (2003) “A framework for opportunistic scheduling in wireless networks” Computer Networks, vol. 41, no. 4, pp. 451-474.

[9] Tse, D., Viswanath, P., (2005) “Fundamentals of Wireless Communication”, Cambridge University Press.

[10] Knopp R., Humblet, P. A. (1995) “Information capacity and power control in single-cell multiuser communications”, in Proc. IEEE Int. Conf. on Communications (ICC’95), vol. 1, pp.331–335.

[11] Gyasi-Agyei, A. (2005) “Multiuser diversity based opportunistic scheduling for wireless data networks”, in IEEE Communications Letters, vol. 9, no. 7, pp. 670-672.

[12] Pätzold, M., (2011) “Mobile Radio Channels”, Wiley.

[13] Kumbhani, B., Kshetrimayum, R. S., (2017) “MIMO Wireless Communications over Generalized Fading Channels”, CRC Press.

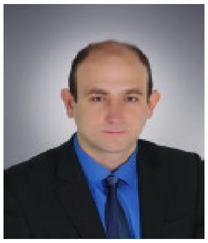
[14] Viswanath, P., Tse, D., Laroia, R., (2002) “Opportunistic beamforming using dumb antennas”, in IEEE Transactions on Information Theory, vol. 48, no. 6, pp. 1277-1294.

[15] Stefanovic, H., Savic, A., (2011) “Some general characteristics of Nakagami-m distribution”, İN Proc. International Symposium on Computing in Informatics and Mathematics (ISCIM 2011), pp. 695-705.

[16] Soy, H., Özdemir, Ö., (2016) “Fuzzy Logic Based Opportunistic Scheduler Design to Improve Fairness in Cellular Wireless Networks”, International Journal of Modern Communication Technologies & Research (IJMCTR), vol. 4, no. 9, 6 pages.

[17] Chang, C. H., Lee, Y., (2009) “Opportunistic beamforming systems with diversity combining", in Proc. 7th International Conference on Information, Communications and Signal Processing (ICICS), pp. 1-5.

AUTHOR



Hakkı Soy was born in Konya, Turkey in 1978. He received the B.S. degree in Electronics Engineering from Uludağ University, Bursa, Turkey in 1999 and the Ph.D. degree in Electrical and Electronics Engineering from Selçuk University, Konya, Turkey in 2013. He joined the Department of Electrical and Electronics Engineering, Necmettin Erbakan University, Konya, Turkey, where he is currently an Assistant Professor. His research interests include wireless sensor networks, sensor-actuator networks, MAC protocols, scheduling algorithms, MIMO communication systems and opportunistic communication.

AUTOMATED VISUALIZATION OF INPUT/ OUTPUT FOR PROCESSES IN SOFL FORMAL SPECIFICATIONS

Yu Chen[1] and Shaoying Liu[2]

[1]Graduate School of Computer and Information Sciences, Hosei University, Japan
[2]Faculty of Computer and Information Sciences, Hosei University, Japan

ABSTRACT

While formal specification is regarded as an effective means to capture accurate requirements and design, validation of the specifications remains a challenge. Specification animation has been proposed to tackle the challenge, but lacking an effective representation of the input/output data in the animation can considerably limit the understanding of the animation by clients. In this paper, we put forward a tool supported technique for visualization of the input/output data of processes in SOFL formal specifications. After discussing the motives of our work, we describe how data of each kind of data type available in the SOFL language can be visualized to facilitate the representation and understanding of input/output data. We also present a supporting tool for the technique and a case study to demonstrate the usability and effectiveness of our proposed technique. Finally, we conclude the paper and point out the future research directions.

KEYWORDS

Visualization, SOFL, Formal specification, Data type, Formal methods

1. INTRODUCTION

Formal specification has proved to be effective to help capture accurate requirements and design in software development if used properly together with practical engineering approaches [1]. While this can considerably contribute to the communication between the developers in a software project, it may not effectively facilitate the communication between the developer and the client due to the fact that mathematical expressions in the formal specification can be difficult for the client to understand in general. Therefore, a potential risk that the formal specification may not correctly and completely define what the client really wants will eventually affect the reliability of the software.

To tackle this problem, formal specification animation has been proposed [2]. The common characteristic of the existing animation techniques is to use test data (also called animation data) to dynamically demonstrate the input-output relation for operations defined in the specification. Compared to the reading and understanding technique, animation is proved to be more effective in validating formal specifications against the client's requirements [3][4]. However, our experience and study suggest that the effect of specification animation is rather limited due to the fact that input and output data with complex structures can be difficult to comprehend during

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP - 2018
pp. 33–48, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80903

animation. Without resolving this limitation, specification animation would be difficult to be transferred to industry for realistic software developments.

In this paper, the researchers put forward a tool supported visualization of input and output data of operations in formal specifications. The proposed visualization technique can be widely applicable to model-based formal notations, such as VDM-SL, Z, and Event-B, SOFL has been chosen, standing for Structured Object-Oriented Formal Language, as the formal notation in our discussions, partly because SOFL has been used in various joint software projects with industry and partly because SOFL offers a comprehensible way to use formal specifications in practical software development.

Our major contributions in this paper are twofold. One is the design of a visualized representation of each type of data provided in the SOFL language. Such a visualization aims to facilitate the expression of the data in a graphical user interface (GUI). The other is the implementation of a software tool supporting the visualization and animation of a single operation called process in SOFL.

The remainder of this paper is organized as follows. Section II briefly introduce the background and related work. Section III discusses the design of visualized representation of various data types. Section IV briefly explain how the visualized representation can be utilized in a single process animation. Section V presents the tool researchers have built to support the proposed technique. Section VI gives a small case study to demonstrate its usability. Finally, in Section VII, we conclude the paper and put forward some future research directions.

2. BACKGROUND AND RELATED RESEARCH

In SOFL, a process performs an action, task, or operation that takes input and produces output. Figure 1 shows a simple form of a process. The process is composed of five parts: name, input port, output port, pre-condition and post-condition. The name of the process always puts in the center of the box. The input port in the left part of the box receives the input data flows and the output port in the right part of the box used to connect output date flows. The pre-condition in the upper part of the box is a condition which the inputs are required to meet, and the post-condition in the lower part of the box is a condition which the outputs are required to satisfy [1].

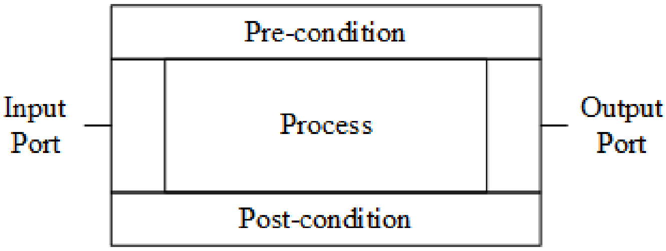


Figure 1. A simple process

Briefly, the process transforms the input data flows to the output data flows. The animation of the process will show the procedure of how the input data flows could transform to output data flows. But in the visualization process, input and output data flows are often composed of a number of complex data types which is difficult for user to understand. Therefore, the effect of animation is quite limited. Without solving this problem, the animation of a process in SOFL formal specifications will be difficult to put into use in industry.

3. RELATED WORK

Formal specification animation attracts a few developers since it provides an effective way to help people especially for simple users to understand the features defined by formal specifications. It helps people to verify whether the specification is consistent with their intended requirements. In this section, we introduce some related work on formal specification animation.

The most common idea of animation is, transforming the specification into one kind of program language. Several animation tools are built based on the specification transformation. PiZA [5] is an animator for Z formal specification. It translates Z specifications into Prolog to generate output variables.

Tim Miller and Paul Strooper introduced a framework for animating model-based specification by using testgraphs [6]. The framework provides a testgraph editor for users to edit testgraphs, and then derive sequences for animation by traversing the testgraph. Gargantini and Riccobene proposed an automatic driven approach to animating formal specifications in Parnas' SCR tabular notation [7]. An important feature of this work is the adoption of a model checker to help find counter-examples that contain a state not satisfying the property to be established by animation.

Liu and Wang introduced an animation tool called SOFL Animator for SOFL specification animation [8]. It provides syntactic level analysis and semantic level analysis of a specification. When performing animation, the tool will automatically translate the SOFL specification into Java program segments, and then use some test case to execute the program.

Li and Liu proposed a novel animation approach called Automatic Functional Scenarios-based Animation. This approach uses data as connection among independent operations involved in a specific behavior to "execute" specifications, and does not translate them to program. Researchers explain how to generate necessary data for animation by modifying an automatic operation function scenario-based test case generation method, and present a case study of applying this animation approach to SOFL specification [9].

4. DESIGN IN DATA TYPES

Data types are essential for specifications because they provide a notation for defining data structures used in specifications [1]. It is crucially important to show a variety of data types with proper kinds of visual interface to make the user understand the input and output accurately. Different data types usually represent data with different structure, quantity, and meaning. For the user, the operators defined on the data types are not interesting, the research team therefore decide to ignore them and focus on the structures and values of the data of various types. Researchers design a visual expression for each of the data types in SOFL to facilitate the user in understanding the structure and meaning for each data type accurately and rapidly. Below the definition of each data type will be presented and then its best manifestations will be explored.

4.1 Numeric, Character and Boolean Types

Numeric, Character and Boolean types are basic data types in SOFL. Numeric types contain *nat0*, *nat*, *int* and *real* representing natural numbers including zero, natural numbers without zero, integers and real numbers, respectively. The character type is the atomic unit for constructing identifiers, operators and delimiters for separating different parts in a specification and contains all characters of the SOFL character set. The boolean type contains only two values: *true* and *false*.

Those basic data types are also very easy to read just by their values. So, in the design of visualization researchers suggest to directly show the value of each types.

4.2 Enumeration Type

An enumeration type is a data type consisting of a finite set of special values called elements, members, enumeral or enumerators of the type, usually with the feature of describing a systematic phenomenon [10]. A variable that has been declared as having an enumeration type can be assigned any of the enumerators as a value. For example, the seven days of a week can be seven enumerators named *Sunday*, *Monday*, *Tuesday*, *Wednesday*, *Thursday*, *Friday* and *Saturday*, belonging to an enumerated type named *Week*. Here is an example of an enumeration of *Week* in SOFL:

Week = {<*Sunday*>, <*Monday*>, <*Tuesday*>, <*Wednesday*>, <*Thursday*>, <*Friday*>, <*Thursday*>}

Actually, an enumeration looks like a finite set, so showing the value by a list in visualization is quite convenient. In view of this idea, the *Week* can be showed like Figure 2.

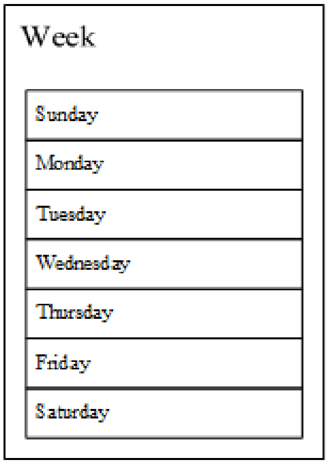


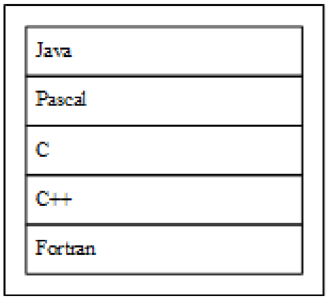
Figure 2. Visualization style for enumeration Week

4.3 Set Type

A set is an unordered collection of distinct objects where each object is known as an element of the set, without any particular order, and no repeated values. In SOFL, a set value of a set type used for process animation is usually finite. For example, a set of programing language names can be:

{"*Java*", "*Pascal*", "*C*", "*C++*", "*Fortran*"}.

Similar to the Enumeration Type, researchers also show the value by a list in visualization as shown in Figure 3. In the design of visualization, researchers focus on the two restrictions: no repeated values and finite quantity. User must give the exact quantity of the items while defining the set, and the tool could remove the repeated values entered by users automatically.



Java
Pascal
C
C++
Fortran

Figure 3. Visualization style for set type

4.4 Sequence and String Type

A sequence is an ordered collection of objects that allows duplications of objects. The difference between sequence and set is that items in a sequence is ordered and allowing element duplication.

Here is an example showing a sequence of natural numbers:

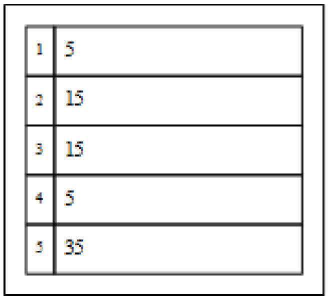
[5, 15, 15, 5, 35]

The visualization of sequence is shown in Figure 4. Researchers add a number at the beginning of each line to show its order.

String is a type which classify all sequences composed of characters. For example, the following is a string value:

"university"

Although a string is a sequence, it is better to display it directly as a whole, from the user-friendly point of view.



1	5
2	15
3	15
4	5
5	35

Figure 4. Visualization style for numeric sequence

4.5 Composite and Product Type

A composite type is a data type which can be constructed using the primitive data types and other composite types [11]. In SOFL, the general format of a composite type is:

composed of
*f*_1: *T*_1
*f*_2: *T*_2

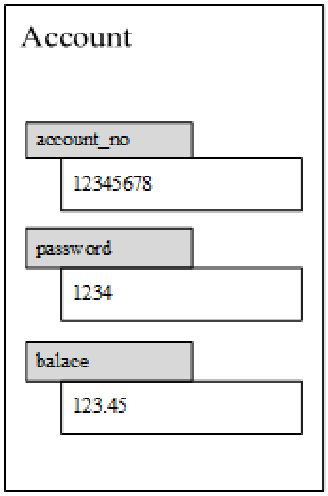
```
   …
   f_n: T_n
   end
```

where *f_i* (*i*=1…*n*) are variables called fields and *T_i* are their types.

Each field is intended to represent an attribute of a composite object of the type. For example, Account can be declared as a composite type of three fields:

```
Account = composed of
   account_no: nat
   password: nat
   balance: real
   end
```

Unlike those previous data types, using only values or lists is not sufficient to present an intuitive visualization that is easily accepted by the user. In this respect researchers learn from the concept of many UI design called tab, with different tabs to show different components. So no matter it is simple type or composed type, all can be composed into a compound data of a composed type with perfect visualization as shown in Figure 5.



Account

account_no: 12345678

password: 1234

balace: 123.45

Figure 5. Visualization style for Composite type: Account

A product type defines a set of tuples with a fixed length. A tuple is composed of a list of values of possibly different types. In SOFL, a product type could be defined as:

T = *T_*1 * *T_*2 * … * *T_n*

where *T_*1, *T_*2, …, *T_n* are *n* types.

For example, type Date is declared as

Date = *nat*0 * *nat*0 * *nat*0

The same visualization can be used as in composed type to present, and for simple types like Date, it can be designed as shown in Figure 6.

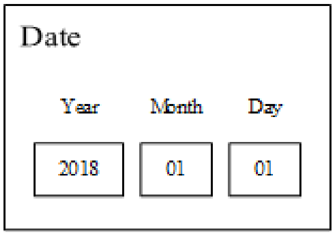


Figure 6. Visualization style for Product type: Data

In this case, it is no need to use tabs since all composed type in Date are non-composed types.

4.6 Map Type

A map, associative array, symbol table, or dictionary is an abstract data type composed of a collection of (key, value) pairs, such that each possible key appears at most once in the collection [12]. A map describes a mapping between two sets. In SOFL, a map is represented with a notation similar to the set notation but use a symbol -> to connect two sets:

{*a*_1 -> *b*_1, *a*_2 -> *b*_2, …, *a*_*n* -> *b*_*n*}

In SOFL, there are two restrictions in map types. One is that all the keys cannot be identical; another is that sets of keys and values are finite. The map type emphasizes the association or correspondence from key to value, and can be considered as a set for each association. So researchers use arrows to represent each set of associations in visualization, while associations use a list similar to set. For example, a map could be defined from twelve months to numbers like:

{*January* -> 1, *February* -> 2, *March* -> 3, *April* -> 4, *May* -> 5, *June* -> 6, *July* -> 7, *August* -> 8, *September* -> 9, *October* -> 10, *November* -> 11, *December* -> 12}

And it can designed as Figure 7.

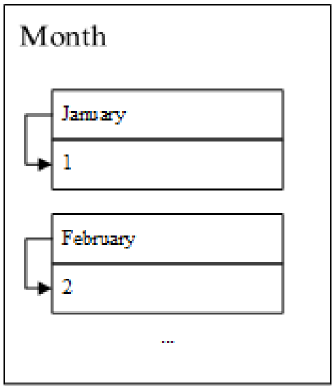


Figure 7. Visualization style for map type

4.7 Union Type

A union is a value that may have any of several representations or formats within the same position in memory; or it is a data structure that consists of a variable that may hold such a value [13]. In other words, a union type definition will specify which of a number of permitted primitive types may be stored in its instances, e.g., “float or long integer”. Contrast with a record (or structure), which could be defined to contain a float and an integer; in a union, there is only one value at any given time. In SOFL, a union type constituted of types could be declared as:

$T = T1 \mid T2 \mid \ldots \mid Tn$

where $T1$, $T2$, …, Tn denote n types.

Since union types are constituted from other different types, they can presented by using visualizations in those types.

4.8 Classes

Class types are common in most object-orient programming languages as extensible types for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods) [14]. In SOFL, a class is a user-defined type, which defines a collection of objects with the same features. The features of objects include attributes, describing their data resources, and operations offering the means for manipulating their data resources and providing functional services for other objects.

In computer science, Unified Modeling Language (UML) is often used to express classes, but users usually merely treat a class as a composite type, so researchers design its visual expression the same as that of composite types.

5. DESIGN IN SYSTEM LOGIC

In SOFL formal specifications, people often use Condition Data Flow Diagram (CDFD) to model how processes work. In the design of main interface, researchers consider the similar style of a process and put data in two ports at each side of the process which could be shown or edited by users, as illustrated in Figure 8.

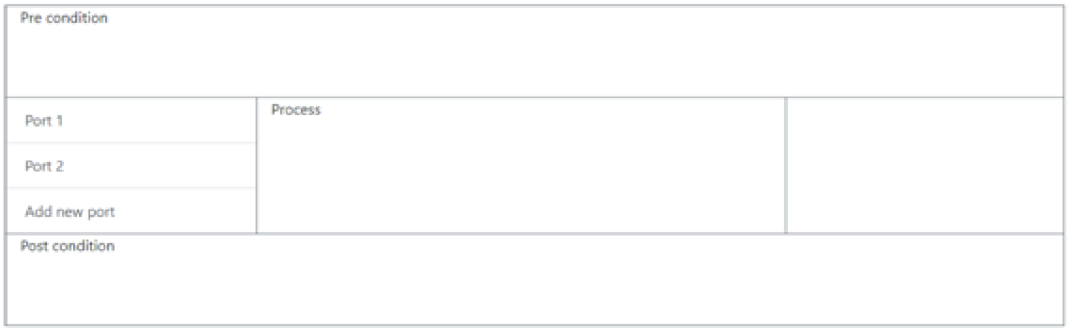


Figure 8. Main interface

Since the space of the main surface are limited, it only shows the concept of input / output data. When the user clicks on the items, the details will be shown in a pop-up box (Figure 9). For input port, the user could add or remove the quantity, types of data and view or edit the values of them. Output port is similar, but the user could only view the values. In the pop-up box, multiple variables are listed, and those composed types will be folded. User could see the detail when clicking to unfold (Figure 10).

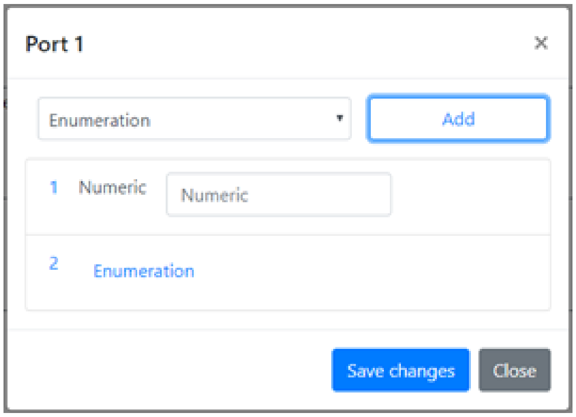


Figure 9. Details of a port

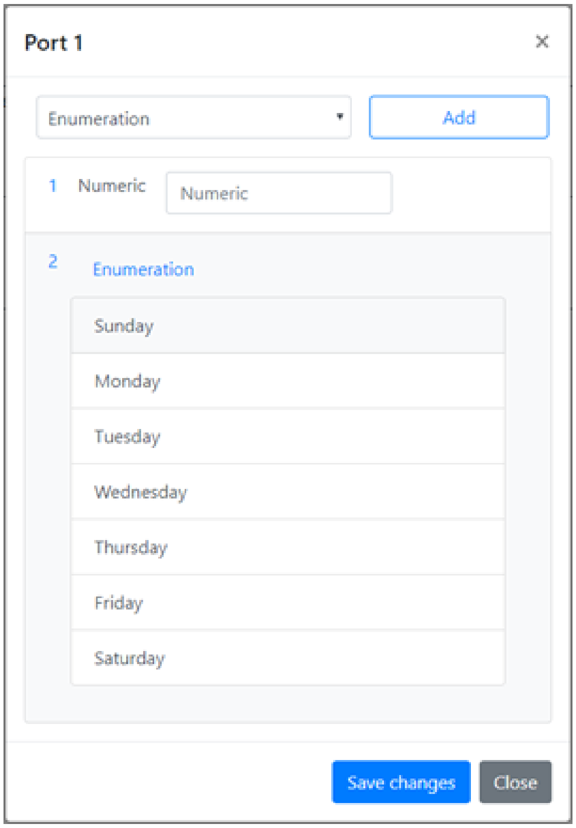


Figure 10. Unfold the enumeration after clicking

6. TOOL IMPLEMENTATION

In the design of network applications, there are mainly two kinds of system architecture. One is called Client-Server (C/S) model which is a distributed application structure that partitions tasks or workloads between servers and clients. The other one is the called Browser-Server (B/S) model which relies on the web technique and is widely used in many systems. By using the browser, users can send requests to server and get response from server by web pages [15].

For a software system, whether choosing B/S or C/S mainly depends on the scene of the user. C/S model is more suitable for those frequently used, complex software programs. B/S model is more suitable for those who are lightweight, for ordinary users of the application. For this tool, researchers choose B/S model to implement since the features for users are simple and publishing for developers are convenient. Software using C/S model always takes long steps for installation or updating which brings terrible user experience. In addition, software using B/S model could be executed in more platforms than C/S model.

The tools utilized for the implementation of our tool could be divided into *front-end tools* and *back-end tools*. In front-end, webpack was used for package management and bootstrap for style designing. In back-end, Django was used for data communication and feature realization.

6.1 Webpack

Webpack is a static module bundler for modern JavaScript applications. When webpack processes your application, it recursively builds a dependency graph that includes every module your application needs, then packages all of those modules into one or more bundles [16]. It is an open-source JavaScript module bunder and takes modules with dependencies and generates static assets representing those modules [17].

6.2 Bootstrap

Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only [18].

6.3 Django

Django is a free and open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent organization established as a 501(c)(3) non-profit [19].

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

7. CASE STUDY

In order to show the effect of our solution, a simple process of bank transfer system was designed to show its animation. The main feature is transferring money from one account to another. The structure of an account is defined as:

```
Account = composed of
  account_no: nat
  password: nat
  balance: real
  end
```

In this process, two accounts have been created, one is for transferor named transfer_out, and another is for transferee named transfer_in. Then a transfer process has been created, include the account number of transferor, the account no of transferee, the password and the amount. The process is specified as:

```
process Transfer (transfer_out: Account | transfer_in: Account |
   transfer_out_account_no: nat, transfer_in_account_no: nat,
   transfer_out_password: nat0, transfer_amount: real)
   transfer_out: Account | transfer_in: Account | transfer_result: bool

   pre (transfer_out_password = transfer_out.password and
   transfer_amount <= transfer_out.balance)

   post (transfer_out.balace = ~transfer_out.balance - transfer_amount and
   transfer_in.balance = ~transfer_in.balance + transfer_amount)

   end_process
```

The structures can be saw from the process that there are 3 input ports and 3 output ports as shown in Figure 11. In the input port of *transfer_out* and *transfer_in*, there is only one variable whose type is Account in each port. In the input port of *tranfer_session*, there are four variables including the account numbers of transferor, the account numbers of transferee, the password and the transfer amount. Output ports are similar.

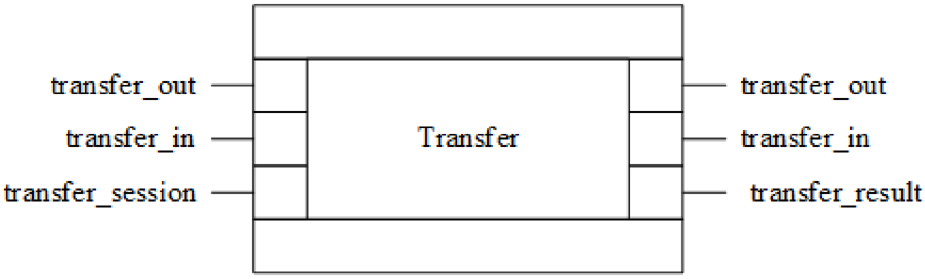


Figure 11. The ports in process Transfer

We entered all the ports in this process to our tool and could see the user interface as illustrated in Figure 12.

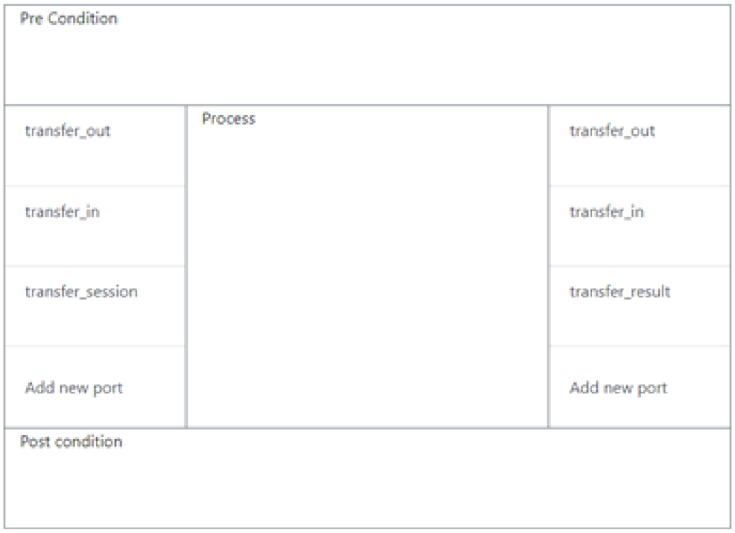
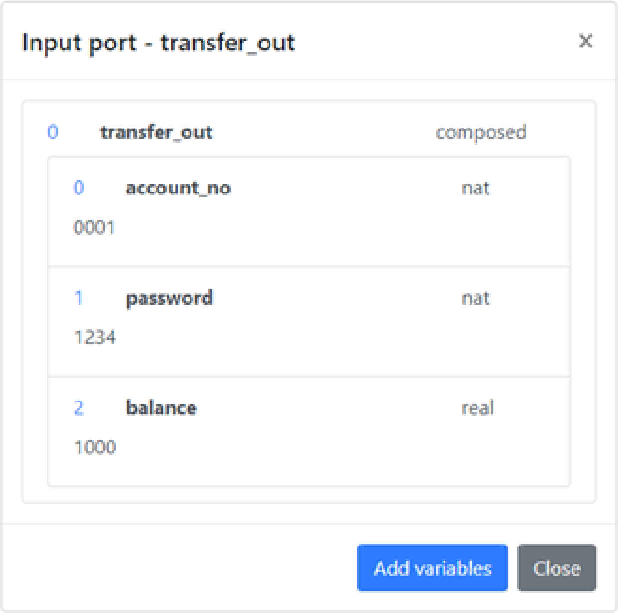


Figure 12. The user interface of process *Transfer*

Then we entered each of the ports. After clicking each port, we can add variables and values as we defined in the specification. We get a user interface shown in Figures 13 – 15.



Input port - transfer_out

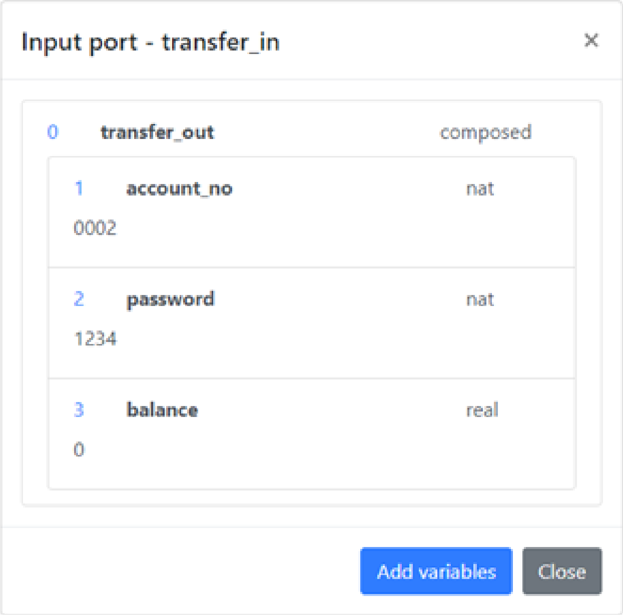
0 **transfer_out** composed

0 **account_no** nat
0001

1 **password** nat
1234

2 **balance** real
1000

Figure 13. The user interface of the input port: *transfer_out*



Input port - transfer_in

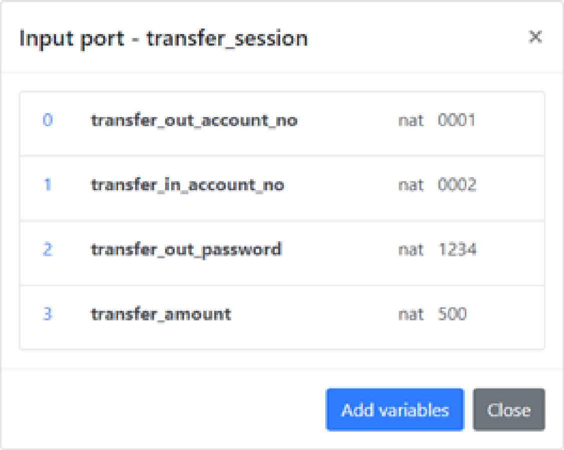
0 transfer_out composed

1 account_no nat
0002

2 password nat
1234

3 balance real
0

Figure 14. The user interface of the input port: *transfer_in*



Input port - transfer_session

0 transfer_out_account_no nat 0001

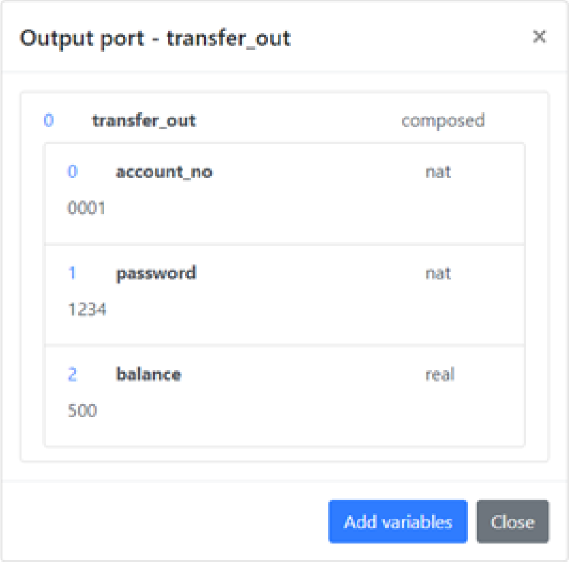
1 transfer_in_account_no nat 0002

2 transfer_out_password nat 1234

3 transfer_amount nat 500

Figure 15. The user interface of the input port: *transfer_session*

After finishing all the input data, we could run the script to animate the process. Then we can check the result from the output port. In the case study, we could see the ports like Figures 16 – 18.



Output port - transfer_out

0 transfer_out composed

0 account_no nat

0001

1 password nat

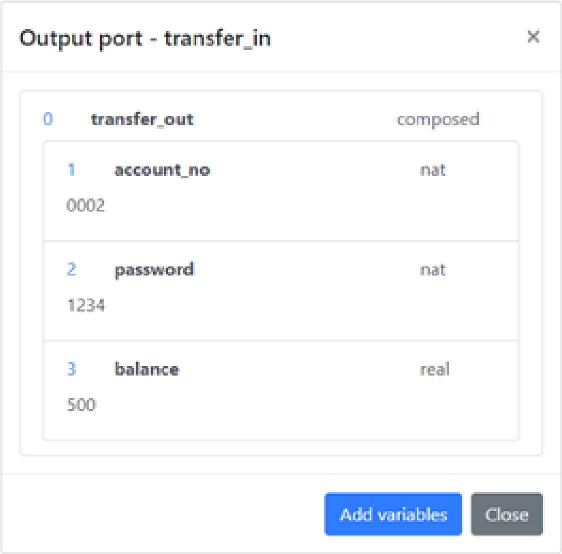
1234

2 balance real

500

Figure 16. The user interface of the input port: *transfer_session*

Let's briefly analyze the implementation process. First, the password in tranfer_session is the same as the password in transfer_out, and the transfer_amount is lower than the balance of transfer_out, so the process begins to execute the transferring steps. We can see the differences between Figure 13 and Figure 16 that the balance of account transfer_out has been reduced and between Figure 14 and Figure 17 that the balance of account transfer_in has been increased. Then from Figure 18 we get the transfer_result, the transferring has been successfully executed.



Output port - transfer_in

0 transfer_out composed

1 account_no nat

0002

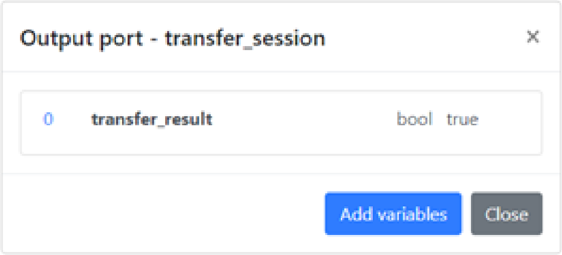
2 password nat

1234

3 balance real

500

Figure 17. The user interface of the input port: *transfer_session*



0	transfer_result	bool	true

Figure 18. The user interface of the input port: *transfer_session*

8. Conclusion and Future Work

In this paper, researchers provided an approach to animating the data types in formal specifications. Researchers focus on single process and try to show all its ports and variables in an intuitive user interface. Researchers analyzed and design the suitable styles to display the data contained in variables. Users could understand what the process is doing by tracking the changes of the values. In the meanwhile, users could enter their own test cases to observe the executing results. This could help user to validate the specification against the user's requirements accurately.

In the future, researchers will continue to finish the module of verification of pre-condition and post-condition. This could help people to verify whether their inputs meet the pre-condition and the outputs satisfy the post-condition. After completing this functionality, the automatic animation of processes will be more integral.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 26240008.

References

[1] Shaoying Liu, "Formal Engineering for Industrial Software Development Using the SOFL Method, Springer-Verlag", ISBN 3-540-20602-7, 2004.

[2] Tim Miller, and Paul Strooper. "Animation Can Show Only the Presence of Errors, Never Their Absence", Proceedings of 2011 Australian Software Engineering Conference. IEEEE CS Press, pages 76-88. 2001.

[3] M. Hewitt, C. O'Halloran, and C. Sennett, "Experiences with PiZA: an Animator for Z", in ZUM'97. 1997, vol. 1212 of LNCS, pp. 37-51, Springer.

[4] Tim Miller and Paul Strooper, "A Framework and Tool Support for the Systematic Testing of Model-Based Specications", ACM TOSEM, vol. 12, no. 4, pp. 409-439, 2003.

[5] M. Hewitt, C. O'Halloran, and C. Sennett, "Experiences with PiZA, an animator for Z. ZUM '97: The Z Formal Specification Notation", pages 37-51. 1996.

[6] Tim Miller, and Paul Strooper. Model-Based Specification Animation Using Testgraphs. The 4th International Conference on Formal Engineering Methods (ICFEM 2002). pages 192-203. 2002.

[7] Gargantini, A., and Riccobene, E. Automatic model driven animation of SCR specifications. Fundamental Approaches to Software Engineering, 6th International Conference (FASE 2003). pages 294-3. 2003.

[8] Shaoying Liu, and Hao Wang. An automated approach to specification animation for validation. The Journal of Systems and Software, No.80. pages 1271-1285. 2007.

[9] Mo Li, Shaoying Liu, "Automated Functional Scenarios-based Formal Specification Animation", 2012 19th Asia-Pacific Software Engineering Conference, 2012.

[10] Enumerated type. (2018, May 09). Retrieved from https://en.wikipedia.org/wiki/Enumerated_type

[11] Composite data type. (2018, April 10). Retrieved from https://en.wikipedia.org/wiki/Composite_data_type

[12] Associative array. (2018, May 11). Retrieved from https://en.wikipedia.org/wiki/Associative_array

[13] Associative array. (2018, May 11). Retrieved from https://en.wikipedia.org/wiki/Associative_array

[14] Kim B. Bruce, Foundations of Object-oriented Languages: Types and Semantics, ISBN 0-262-02523-X, 2002.

[15] Xin Wang, Qijun Chen, "Design and Implementation of Real-time Building Energy Visualization Platform Based on Mixed Model of B/S and C/S", 2014.

[16] Concepts, webpack documentation, (n.d.). Retrieved from https://webpack.js.org/concepts/

[17] Webpack. (2018, May 19). Retrieved from https://en.wikipedia.org/wiki/Webpack

[18] Bootstrap (front-end framework). (2018, May 17). Retrieved from https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)

[19] Django (web framework). (2018, May 18). Retrieved from https://en.wikipedia.org/wiki/Django_(web_framework)

AUTHORS

Yu Chen, now is a graduate student at Hosei University (Japan) and Science of Technology of China, researching SOFL Formal Methods.

Shaoying Liu, is Professor of Software Engineering at Hosei University, Japan.

TBFV-M: TESTING-BASED FORMAL VERIFICATION FOR SYSML ACTIVITY DIAGRAMS

Yufei Yin*,+ Shaoying Liu* and Yixiang Chen+

*Hosei University, Tokyo, Japan
+East China Normal University, Shanghai, China

ABSTRACT

SysML activity diagrams are often used as models for software systems and its correctness is likely to significantly affect the reliability of the implementation. However, how to effectively verify the correctness of SysML diagrams still remains a challenge. In this paper, we propose a testing-based formal verification (TBFV) approach to the verification of SysML diagrams, called TBFV-M, by creatively applying the existing TBFV approach for code verification. We describe the principle of TBFV-M and present a case study to demonstrate its feasibility and usability. Finally, we conclude the paper and point out future research directions.

KEYWORDS

SysML activity diagrams, TBFV, test path generation, formal verification of SysML diagram

1. INTRODUCTION

Model-Based Systems Engineering (MBSE) [1] is often applied to develop large scale software systems in order to effectively ensure their reliability and to reduce the cost for testing and verification. The systems modelling language SysML [2, 3] can support effective use of MBSE due to its well-designed mechanism for creating object-oriented models that incorporate not only software, but also people, material and other physical resources. In MBSE, SysML models are often used as the design for code. Therefore, its correctness in terms of meeting the users' requirements becomes critical to ensure the high reliability of the code. Unfortunately, to the best of our knowledge from the literature, there are few tools to support the verification of SysML models [4, 5] in particular rigorous ways of verification.

Testing-Based Formal Verification (TBFV) proposed by Liu [6-8] shows a rigorous, systematic, and effective technique for the verification and validation of code. Its primary characteristic is the integration of the specification-based testing approach and Hoare logic for correctness proof of code to guarantee the correctness of all the traversed program paths during testing. The advantage of TBFV is its potential and capability of achieving full automation for verification through testing. However, the current TBFV is mainly designed for sequential code in which all of the details are formally expressed, and there is no research on applying it to verify SysML models yet. In this paper, we discuss how the existing TBFV can be applied to SysML models for their verification and we use TBFV-M (testing-based formal verification for models) to represent the newly developed approach. Since SysML Activity Diagrams can model the systems dynamic behaviour and describe complex control and parallel activities, which are similar to code but with additional constructs such as parallel execution, our discussion in this paper focuses on the activity diagrams.

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP - 2018
pp. 49–68, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80904

The essential idea of TBFV-M is as follows. All of the functional scenarios are first extracted from a given formal specification defining the users' requirements where each functional scenario defines a meaningful functional behaviour of the system. Meanwhile, test paths are generated from corresponding SysML Activity Diagrams waiting to be verified. Then, test paths are matched with functional scenarios by comparing the collection of decision condition of each test path and the guard condition of the functional scenario. After this, the pre-condition of the test path is automatically derived by applying the assignment axiom in Hoare logic based on the functional scenario. Finally, the implication of the pre-condition of the specification in conjunction with the guard condition of the functional scenario to the derived pre-condition of the path is verified through automatic proof or testing to determine whether the path contains bugs. The details of this approach will be discussed from Section 5.

The remainder of the article will detail the TBFV-M method. Section 2 presents related work we have referenced. Section 3 introduces the Testing-Based Formal Verification technique for the verification and validation of code. Section 4 mainly details the whole development process of using Model-Based Systems Engineering and the application scenarios of TBFV-M method. Section 5 characterizes the definitions of basic terms and concepts and section 6 describes the principle of TBFV-M, showing the core technology of TBFV-M. Section 7 uses one case study to present the key point of TBFV-M. Finally. The details of the implementation of the algorithm are presented in Section 8. Section 9 concludes the paper.

2. RELATED WORK

In this section, we briefly review the existing work related to our study. For the sake of space, we focus on those we have referenced during our research. We divide the related work into four different parts, including testing-based verification, requirements verification, verification using Hoare Logic and test case generation.

Considering the shortcoming of formal verification based on Hoare logic being hard to automate, Liu proposed the TBFV (Testing-Based Formal Verification) method by combining specification-based testing with formal verification [6]. This method not only take the advantage of full automation for testing, but also the efficiency of error detection with formal verification. Liu also designed a group of algorithms [9] for test cases generation from formal specification written in SOFL [10]. A supporting tool [8] is also developed. These efforts have significantly improved the applicability of formal verification in industrial settings.

Franco Raimondi [11] addressed the problem of verifying planning domains written in the Planning Domain Definition Language (PDDL). First, he translated test cases into planning goals, then verified planning domains using the planner. A tool PDVer is also generated. In this paper, testing is also used during verification and the effectiveness and the usability is improved.

Stefano Marrone [12, 13] designed a Model-Driven Engineering approach, in which formal models are constructed and test cases are generated from UML model, utilizing UML profiles and model transformation algorithms, automatically. As they claimed, formal models can be used for quantitative analysis of non-functional properties, while test cases can be used for model checking. A railway signalling example is shown to introduce its integration, usability and reduction of manual activities.

Feng Liang [14] proposed a vVDR (Virtual Verification of Designs against Requirements) approach for verifying a system with its requirement. In his research, the system is modeled in Modelica, and requirement verification scenarios are specified in ModelicaML, an UML profile and a language extension for Modelica. vVDR approach guarantees that all requirements can be verified by running this scenario automatically. However, the deficiency appears when the number of requirements and scenarios increase.

Ralf Sasse [15] designed a tool called Java+ITP to verify a subset of the Java language. During the verification process, Maude-based continuation passing style (CPS) is used to rewrite the logical semantics of Java, and they also developed CPSbased Hoare Logic rules to justify the correctness of the rewritten fragment. Ralf Sassi's tool provides an extensibility of Hoare logics, but exceptions and objects should be considered in the future research.

Magnus O. Myreen [16] used Hoare Logic to deal with machine code. They designed a mechanized Hoare-style programming logic framework to accommodate the restrictions and features present in real machine-code, such as finite memory, data and code in the same memory space. ARM machine-code now can be verified using the proposed logic.

Jonathan Lasalle [17] utilized the existing UML/OCL Model-Based Test generation tool, Smartesting Test DesignetTM. He designed rewriting rules to translate a SysML model into an equivalent UML model. The advantage of this process is that we can use the existing UML tools to handle the SysML model.

Ashalatha Nayak [18] introduced an approach to transform the particular Activity Diagram into a model that can be used for testing, called ITM, based on its structure characteristics. The advantage of using ITM is that it can simplify the process of extracting and analyzing test scenarios based on the coverage criteria. However, it also has limitations on processing unstructured Activity Diagram because the unstructured Activity Diagrams shape is out of structure.

Oluwatolani Oluwagbemi [19] proposed a new concept called activity ow tree (AFT) and it can store the information obtained by traversing the activity diagram. Then, AFT is used as an intermediate expression to generate test cases automatically. They designed the transformation and generation algorithm and compared their achievement with the work done by the predecessors.

Inspired by Liu's work, we apply and extend the TBFV approach to models and propose the TBFV-M. A model is more intuitive than a formal specification because it requires less relevant background knowledge and is easier to communicate with customers. TBFV approach shows the treatment of code, while TBFV-M approach deals with SysML Activity Diagrams. And different with Feng Liang's work, TBFV-M approach do not use other supporting tools, like Modelica, we merely use Hoare Logic to do the verification. Referring to test case generation, TBFV- M approach can deal with unstructured diagrams, which may have stronger processing power than existing approaches.

3. INTRODUCTION OF TBFV FOR CODE

TBFV is a novel technique that makes good use of Hoare logic to strengthen testing. The essential idea is first to use specification-based testing to discover all traversed program paths and then to use Hoare logic to prove their correctness. During the proof process, all errors on the paths can be detected.

Testing is a practical technique for detecting program errors. A strong point of testing superior to formal correctness verification is that it is much easier to be performed automatically if formal specifications are adopted [20], but a weak point is that existing errors on a program path may still not be uncovered even if it has been traversed using a test case. TBFV takes advantage of testing, realized full automation for error detection efficiency.

TBFV is a specific specification-based testing approach that takes both the precondition and post-condition into account in test case generation [21]. It treats a specification as a disjunction of functional scenarios (FS), and to generate test sets and analyse test results based on the functional scenarios. A functional scenario is a logical expression that tells clearly what condition is used to constrain the output when the input satisfies some condition. To precisely describe this strategy, we first need to introduce functional scenario. S_{pre} and S_{post} denote the pre- and post-conditions of operation S. Let:

$$S_{post} = (G_1 \land D_1) \lor (G_2 \land D_2) \lor \dots \lor (G_n \land D_n) \quad (1)$$

G_i and D_i ($i \in 1, \dots, n$) are two predicates, called guard condition and defining condition, respectively. The definition of functional scenarios and FSF (functional scenario form) are list below:

$$\text{Functional Scenario} = S_{pre} \land G_i \land D_i \quad (2)$$

In the definition of functional scenario, $S_{pre} \land G_i \land D_i$ is treated as a scenario: when $S_{pre} \land G_i$ is satisfied by the initial state (or intuitively by the input variables), the final state (or the output variables) is defined by the defining condition D_i. The conjunction $S_{pre} \land G_i$ is known as the test condition of the scenario, which serves as the basis for test case generation from this scenario.

$$FSF = (S_{pre} \land G_1 \land D_1) \lor (S_{pre} \land G_2 \land D_2) \lor \dots \lor (S_{pre} \land G_n \land D_n) \quad (3)$$

A systematic transformation procedure, algorithm, and software tool support for deriving an FSF from a pre-post style specification written in SOFL have been developed in our previous work [22]. In the case study section, we will show an example to detail FSF generation. Test cases can be generated from FSF. TBFV has three main techniques. First, generate test cases from specification. Second, form path triple and the definition are below:

$$\{S_{pre} \land G_i\} P \{ D_i\} \quad (4)$$

P is called a program segment, which consists of decision (i.e., a predicate), an assignment, a return statement, or a printing statement. It means that if the pre-condition S_{pre} and the guard condition G_i of the program are both true before path P is executed, the post-condition D_i of path P will be true on its termination.

Finally, repeatedly apply the axiom for assignment to derive a pre-assertion, denoted by P_{pre}. And the correctness of the specific path is transformed into the implication $S_{pre} \land G_i \rightarrow S_{pre}$. If the implication can be proved, it means that no error exists on the path; otherwise, it indicates the existence of some error on the path.

4. TBFV-M IN MBSE

Model-Based Systems Engineering (MBSE) combines process and analysis with architecture. In the last decade, the model-driven approach for software development has gained a growing interest of both industry and research communities as it promises easy automation and reduced time to market [23]. Because of the graphical notation for defining system design as nodes and edge diagrams, SysML model addresses the ease of adoption amongst engineers [24].

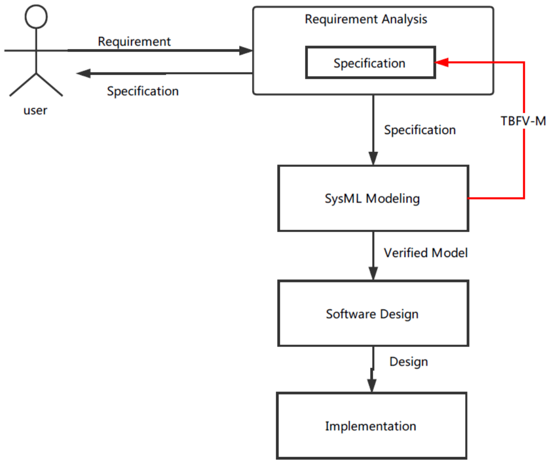


Figure 1. TBFV-M usage scenario

In the development process using MBSE, as shown below, the users' requirements are obtained first and the requirement document is usually written in natural language. This document is not only difficult for developers to understand, but also may contain ambiguities and other problems.

To obtain requirements without ambiguities, we may generate a SysML Model. During the model-driven development process, we use the SysML Model Diagram to communicate with the user, because it does not contain many mathematical symbols and syntax. In the SysML modelling phase, we will refine the SysML diagram, involved in the SysML model and specification.

During the Model-Driven process, model is an important medium for the Model based system engineering development. The TBFV-M method is mainly used to verify whether SysML Activity Diagram model meets the user's requirements written in SOFL (Structured-Object-oriented-Formal Language).

5. Basic Concept

5.1. Formal Definition of Activity Diagram

Activity Diagram Formal Definition [2] can be represented as:

$$AD = (Node; Edge) \tag{5}$$

Node is a set of nodes of which definition as follow:

$$\begin{gathered}Node = \{InitialNode; FlowFinalNode; ActivityFinalNode; ActionNode; \\ ActivityNode; \quad ForkNode; JoinNode; DecisionNode; MergeNode; \\ RecieveSignalNode; SendSignalNode\}\end{gathered} \tag{6}$$

Edges defines the relationship between nodes such that:

$$\mathrm{Edge} = \{(\mathrm{x}, \mathrm{y}) | x, y \in Node\} \tag{7}$$

There are two types of edges: control flow and object flow. Control flow edges represent the process of executing token passing in AD and object flow edges are used to show the flow of data between the activities in AD.

5.2. Test Case

From a global view, test case based on the SysML activity diagram consists of test path and test data. And the definition is as followed:

$$\mathrm{TC(AD)} = (\mathrm{Path};\ \mathrm{Data}) \tag{8}$$

For activity diagram, test scenario consists of a series of actions and edges in the diagram. Based on the formal definition of the activity diagram given above, the test path is defined as follow:

$$\mathrm{path} = (\, a'_1 \,, a'_2 \,, \ldots \,, a'_n \,) \tag{9}$$

$$a'_i = (\, t_n \,, a_n \,) \,, (\, i = 2 \,, \ldots \,, n) \tag{11}$$

$$t_n = \ a_{i-1} \ \rightarrow \ a_i \,, (\, i = 2 \,, \ldots \,, n) \tag{12}$$

In this formula, a_i means node, t_i means edge.In this case, a test path is a set of nodes, starting from node a_1 and ending with node a_n through the transition edges $t_2 \ldots t_n$. For activity diagram, a_1 and a_n represent the initial node and final node, respectively. Test data indicates the input information corresponding to a particular test scenario including various types of data, even user actions and so on.

5.3. Test Coverage Criteria

For software, the adequacy measurement of testing is reflected in the rate of coverage and effectiveness of the test case. The test coverage criteria in white box tests includes statement coverage, branch coverage, conditional coverage and so on. These coverage criteria ensure the sufficiency of testing and provide implications for the test case generation algorithm. Here are four test coverage criteria used in our design, for test case generation of SysML activity diagram [19,25,26]:

- **Action coverage criteria:** In software testing process, testers are often required to generate test cases to execute every action in the program at least once.

- **Edge coverage criteria:** In software testing process, testers are often required to generate test cases to pass every edge in the program at least once.

- **Path coverage criteria:** These coverage criteria require that all the execution paths from the programs entry to its exit are executed during testing.

- **Branch coverage criteria:** These coverage criteria generate test cases from each reachable decision made true by some actions and false by others.

5.4. Hoare Logic

Hoare Logic is a formal system developed by C. A. R. Hoare [27, 28], and it is designed for the proof of partial correctness of a program. In Hoare Logic, the Hoare Triple [29] is best known and is also referenced in our method. The Hoare triple is of this form:

$$\{P\}\ C\ \{Q\} \tag{13}$$

, where P and Q are assertions and C is a command. P is named the pre-condition, which is a predicate expression describing the initial states and Q the post-condition, which is also a predicate expression describing the final states.

Hoare also established necessary axioms to define the semantics of each program construct, including axiom of assignment, rules of consequence, axioms of composition, axioms of alternation, iteration and block. Axiom of assignment is used in our work, so we will briefly introduce it:

$$\{Q(E\backslash x)\}\ x:=E\ \{Q\} \qquad (14)$$

,where x is a variable identifier, E is an expression of a programming language without side effects, but possibly containing x, $Q[E\backslash x]$ is a predicate resulting from Q by substituting E for all occurrences of x in Q. This axiom means that to verify the correctness of the assignment, the postcondition Q should be satisfied. This equals to $Q[E\backslash x]$ is true because x is assigned by representing E after the execution.

6. Procedure of TBFV-M

The TBFV-M method takes the specification describing the users' requirements and the SysML Activity Diagram model as input and verifies the correctness of the SysML model with respect to the specification. The procedure of TBFV-M is illustrated in Figure2.

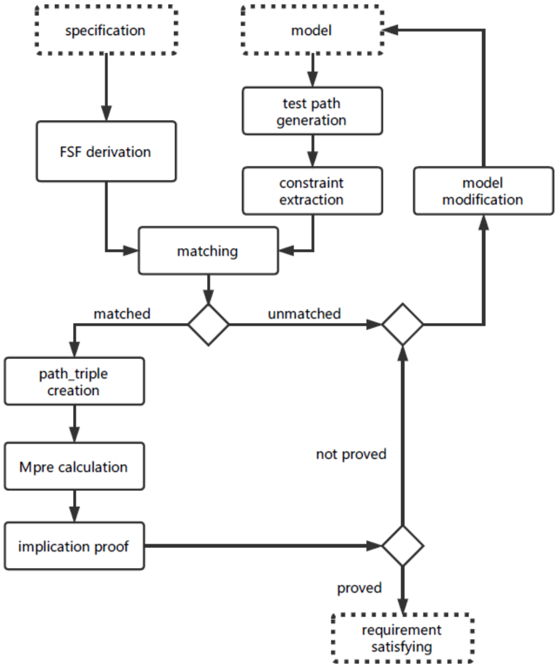


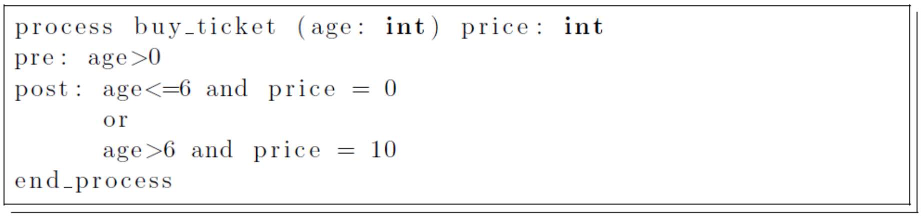
Figure 2. TBFV-M processing procedure

From this figure, we find that functional scenarios are derived from the specification written in the pre- / post-condition style, while test paths are generated from the Activity Diagram and the data constraints can be extracted from each test path. Then, the extracted data constraints are used to match with functional scenarios. A matching algorithm is defined by us. We will verify the successful matched the test path according to the requirements represented in specification. The verification part can be separated into three parts: first, create a path triple, and then use the axiom of Hoare Logic to derive pre-assertion for each test path. Finally, prove the implication of the pre-condition in the specification and pre-assertion. If we can prove all the implication of pre-assertion of all the test paths of the model and the matching pre-condition, then the model is to meet the requirements.

These critical steps in the TBFV-M method, including functional scenarios derivation, test path generation, matching algorithm, pre-assertion derivation and implication will be discussed next and comprehensive details will be described in the case study section.

6.1. Unified Formal Expression

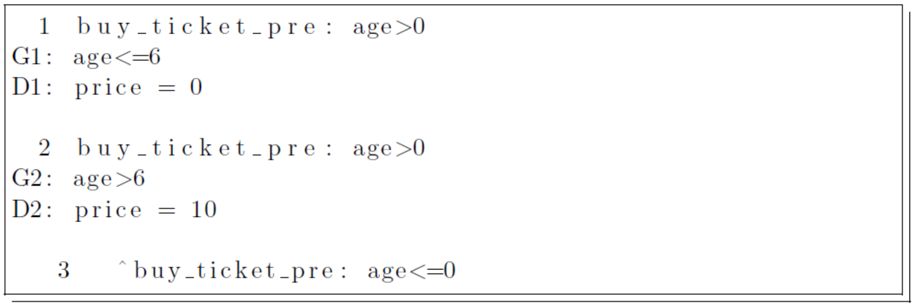
Using a unified formal expression can reduce the ambiguity between communications, We establish the unified formal expression, including specification guide and modeling guide. Specification reflects complete requirements and we chose SOFL to describe formal specification. An example specification written in SOFL is given below. It describes that if a person is smaller than 6, he will be free; otherwise, he should buy the normal price for $10.



```
process buy_ticket (age: int) price: int
pre: age>0
post: age<=6 and price = 0
      or
      age>6 and price = 10
end_process
```

6.2. Functional Scenarios Derivation

The overall goal of functional scenario derivation is to extract all functional scenarios completely in "$S_{pre} \wedge G_i \wedge D_i$" form (FSF), as mentioned above in TBFV section. Because this part is not our main topic and has been researched before. In our work, we assume that an FSF of the specification has been available somehow. The below segment of the process buy ticket, mentioned previously, shows the FSF generated from the specification described in the last one.



```
  1  buy_ticket_pre: age>0
G1: age<=6
D1: price = 0

  2  buy_ticket_pre: age>0
G2: age>6
D2: price = 10

   3   ^buy_ticket_pre: age<=0
```

6.3. Test Paths Generation

A test path auto-generation tool based on the SysML Activity Diagram model takes the model as input and generates test cases as outputs automatically. Our SysML Activity Diagram test path generation includes three parts. First, we use transformation algorithm to compress the input Activity Diagram, which may contain unstructured module. The transformation is a cyclic process, dealing with loop module, concurrent module and the problem of multiple starting nodes separately. After compressing, we transform this unstructured activity diagram into an intermediate representation form Intermediate Black box Model (IBM). IBM consists of one basic module and a map from black box to the corresponding original actions. The third phase of our approach is test path generation based on IBM. In this phase, two problems should be solved, which are basic module test path generation and black box test path generation. Details of automated test paths generation algorithm and implementation of unstructured SysML Activity Diagram has been developed in our previous work [30].

6.3.1. Loop Module

The Loop module in the SysML activity diagram can be considered as a node collection, and these nodes in the collection can be cycled multiple times. As shown in Figure3, according to the location of cyclic judgment condition located at the end of the loop module or the front, we can divide the loop module into do-while loop and while-do loop.

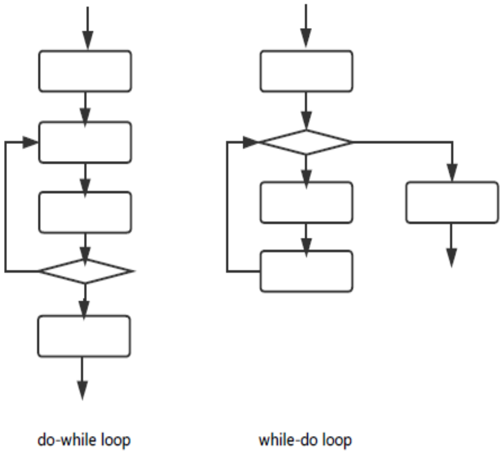


Figure 3. Classification of loop modules

The first step in the transformation algorithm of the Loop module is to identify the loop module, the second step is to compress it into a black box node loop, and finally reinsert it into the original SysML activity diagram. Figure4 shows the process.

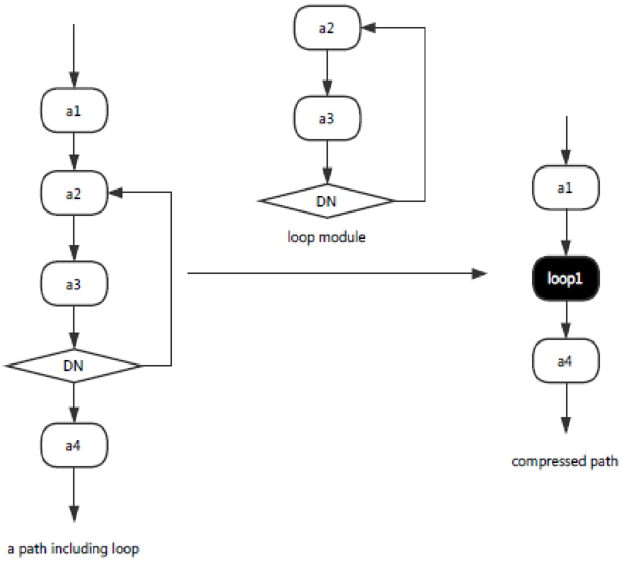


Figure 4. The transformation of loop module

Since the infinite traversal loop is not possible, it is possible to propose a different expansion algorithm for different types of loops when processing the loop module. For simple loop, you can take the following test case sets (where n is the maximum number of passes allowed):

- skip the entire loop
- go through the loop once
- go through the loop twice
- go through the loop m times
- go through the loop n-1, n, n + 1 times

6.3.2. Concurrent Module

In the SysML activity diagram, the most common form of a concurrent module is a pair of fork node and join node and all actions between these two nodes, as shown in Figure5 (a). The fork node can be represented as simultaneous start of multiple parallel streams and the join node represents the possible synchronization of multiple parallel streams, inflowing into next action. The logical representation is AND. However, the synchronization stream can also be the logical relationship OR, as shown in Figure5 (b).

Depending on how many concurrent streams can be synchronized by the join node, the parallel modules can be divided into partJoin concurrent and noJoin concurrent, as shown in Figure5 (c) and (d) below, respectively.

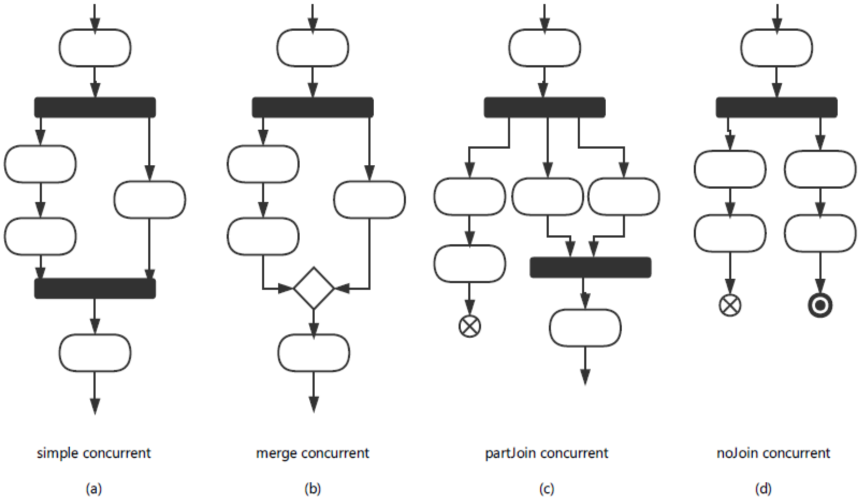


Figure 5. Classification of concurrent modules

On the test path generation algorithm for concurrent modules, the first step is to identify the concurrency module, the second step is to compress it into a black box node FJ (Fork-Join), and finally reinsert it into the original SysML activity diagram, as shown in the following Figure6.

For concurrent modules, we can use the Concurrent module path generation algorithm and generate the test path automatically. For the compressed basic path, the test path generation algorithm of the basic module can be applied. Once the basic path is generated, replace the FJ black box with the test path generated from the concurrency module.

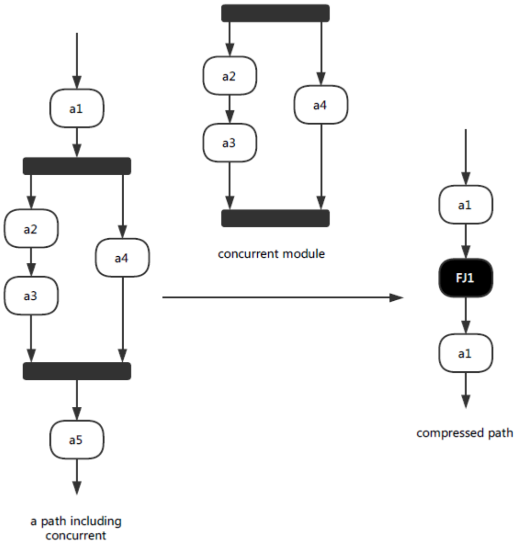


Figure 6. The process of transformation of concurrent modules

6.3.3. Test path generation with IBM

The test case generation with IBM needs to deal with three types of modules, which are basic modules, concurrent modules and loop modules. The basic module is the activity diagram that compresses the concurrent module and the loop module into the black box node respectively. The concurrent module and the Loop module are the transformed black box module which contain the unique incoming edge and the unique outgoing edge.

For basic module, without considering the concurrent module and the loop module, we can transform the SysML activity diagram model into a directed acyclic graph, using the idea of DFS (Depth First Search) algorithm. While for unconstructed module, we can use corresponding generation algorithm and generate the test path automatically. After the basic path is created, the black box can be replaced with the test path generated by the unconstructed module.

6.3.4. Motivation Example

Figure7 is an unstructured SysML activity diagram model, which contains a concurrency module and a loop module.

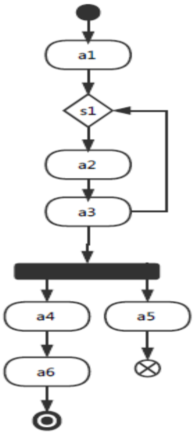


Figure 7. Motivating case

Figure8 shows how to compress an unstructured activity diagram and transform the unstructured module into a black box node. Eventually the unstructured activity diagram converts into an intermediate representation of IBM. The first step is to identify the loop module and compress it into a black box node while-do loop1, shown in Figure8(a). The compressed black box node is the intermediate representation of the loop shown in the following Figure9(a). The second step is to identify the noJoin concurrency module and compress it into a black box node No FJ1, shown in Figure8(b). The compressed black box node is shown in the following Figure9(b).

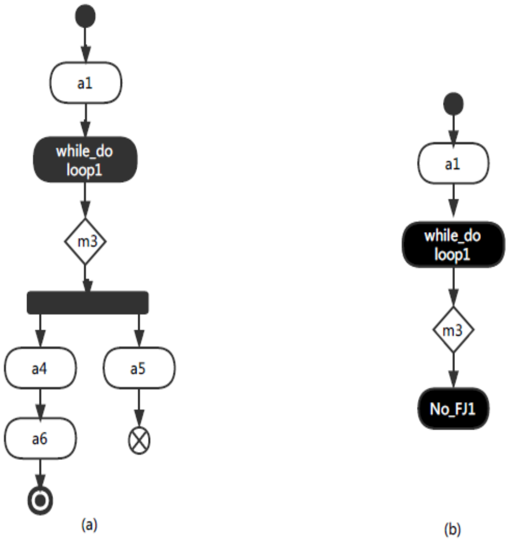


Figure 8. The process of transformation

Figure8(b) is a compressed and structured SysML activity diagram that can be used to automatically generate test cases. Finally, the black box module can be replaced.

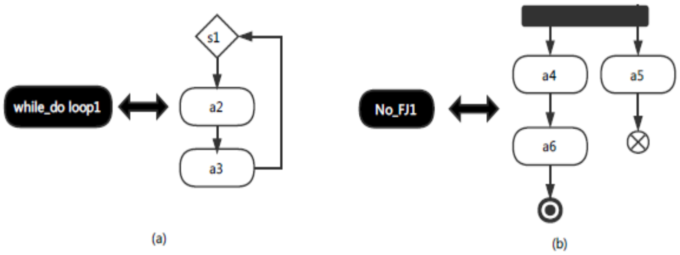
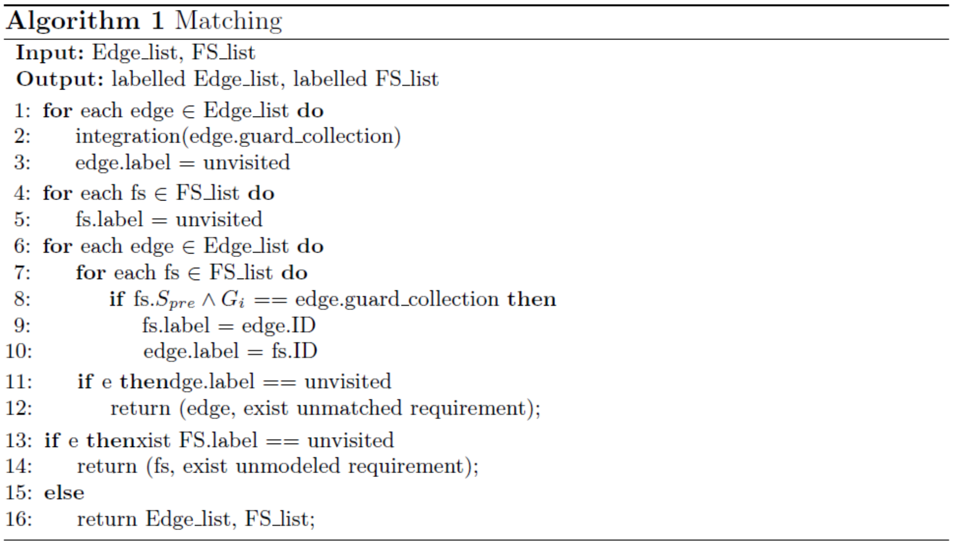


Figure 9. The map of black boxes

6.4. Matching Algorithm

Matching the test path with functional scenario is very important for verification. In order to verify the correctness of one path in Activity Diagram, we need to match it with corresponding functional scenario. The constraints of test path can be extracted from edges of each path, which are used to compare with $S_{pre} \wedge G_i$ part of functional scenario. If unmatched test paths or functional scenarios appears, it means some errors may be existed in this model. And the model needs to be modified. The matching algorithm is given below.

Matching algorithm takes the edge list and FS_list as input. Edge list is the collection of guard conditions saved from test path and FS_list is extracted functional scenario form from specification. First, the algorithm sets the label of the two lists unvisited. And for each in edge list do data integration. Data integration is like data intersection. For example, if we contain two guard conditions x < 6 and x < 60, the integration of it is x < 6.



Algorithm 1 Matching

```
Input: Edge_list, FS_list
Output: labelled Edge_list, labelled FS_list
for each edge $\in$ Edge_list do
    integration(edge.guard_collection)
    edge.label = unvisited
for each fs $\in$ FS_list do
    fs.label = unvisited
for each edge $\in$ Edge_list do
    for each fs $\in$ FS_list do
        if fs.$S_{pre} \wedge G_i$ == edge.guard_collection then
            fs.label = edge.ID
            edge.label = fs.ID
    if e thendge.label == unvisited
        return (edge, exist unmatched requirement);
if e thenxist FS.label == unvisited
    return (fs, exist unmodeled requirement);
else
    return Edge_list, FS_list;
```

After completing the initialization step, find a matching functional scenario for each element in edge list. The specific operation is: the edge after the integration compares with $S_{pre} \wedge G_i$ in the functional scenario, if exactly the same, then we find the edge with the matched functional scenario. If there is no exact matched functional scenario, then there is an inaccurate modeling problem and needs to be refined. Therefore, immediately terminate the program, the problem of the edge will also be returned. After traversing all the edge_list, we also need to check whether each in FS_list has been visited. If there is an unvisited functional scenario, then it means that there is a requirement that the model fails to be represented in the specification, and the model needs to be refined.

6.5. Path Triple Establishment

Establish Path Triple and apply each node with the axiom in Hoare Logic. "($S_{pre} \wedge G_i \wedge D_i$) (i = 2, … ,n)" denote one functional scenario and P = [$node_1$; $node_2$; … ;$node_m$] be a program path in which each $node_j$(i = 2, … , n) is called a functional node, which is a DecisionNode, ActionNode, or others. Assume each path P has its own target functional scenario, which is decided utilizing matching algorithm. To verify the correctness of P with respect to the functional scenario, we need to construct Path Triple: {S_{pre}} P {$G_i \wedge D_i$}.

The path triple is similar in structure to Hoare triple, but is specialized to a single path rather than the whole program. It means that if the pre-condition S_{pre} of the program is true before path P is executed, the post-condition $G_i \wedge D_i$ of path P will be true on its termination. By applying the axiom of assignment in Hoare Logic repeatedly, we can get pre-assertion, P_{pre}. Each node has different processing approach, and the details are listed in the form below.

Table 1. Processing approach of AD node

Node Type	Approach
ActionNode(assignment)	The axiom for assignment
ActionNode(input/output)	SKIP
Others node	SKIP

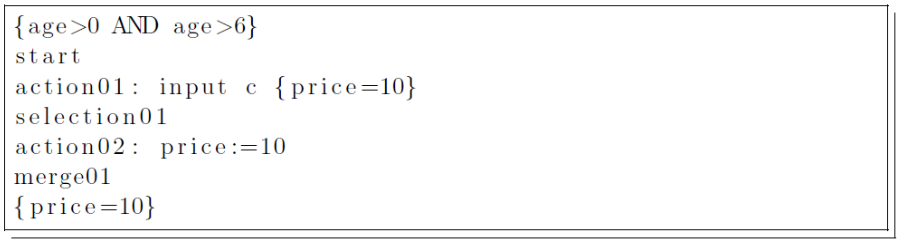
Finally, we can form the following expression:

$$\{S_{pre} \wedge G_i\}\ P\ \{D_i\} \tag{15}$$

$$\{S_{pre} \wedge G_i\} \rightarrow P_{pre} \tag{16}$$

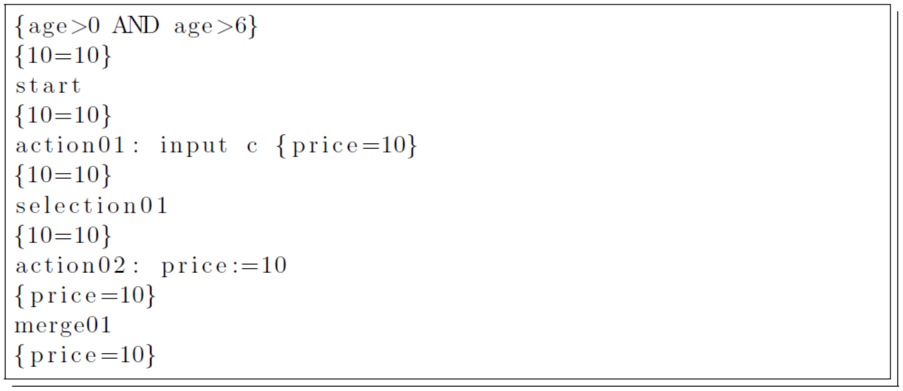
, where $S_{pre}(\sim x/x)$, $P_{pre}(\sim x/x)$ and $G_i \wedge D_i(\sim x/x)$ are a predicate resulting from substituting every decorated input variable ~x for the corresponding input variable x in the corresponding predicate, respectively.

If we get a path [start -> action01: input c -> selection01 -> action02: price: =10 -> merge01 -> end], which represents a path generating according to the above specification. Using the above table, we can form path triple:



```
{age>0 AND age>6}
start
action01: input c {price=10}
selection01
action02: price:=10
merge01
{price=10}
```

Then, we can apply for assignment to this path triple, like this:



```
{age>0 AND age>6}
{10=10}
start
{10=10}
action01: input c {price=10}
{10=10}
selection01
{10=10}
action02: price:=10
{price=10}
merge01
{price=10}
```

, where {10=10} is pre-assertion.

6.6. Implication

Prove the implication. Finally, the correctness of one path whether it meets the corresponding requirement is changed into the proof of the implication "$S_{pre} \wedge G_i \rightarrow S_{pre}$". If the implication can be proved, it means that the path can model one part of the requirement; otherwise, it indicates the existence of some error on the path.

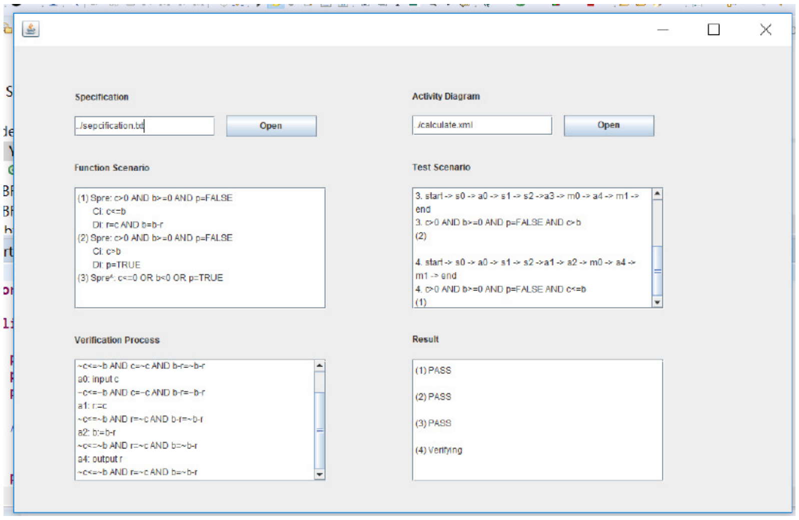
Formally proving the implication "$S_{pre} \wedge G_i \rightarrow S_{pre}$" may not be done automatically, even with the help of a theorem prover such as PVS, depending on the complexity of S_{pre} and P_{pre}. Our strategy is as follows: if the complexity of data structure is not high, we will transform the problem into solver, which can achieve full automation. Otherwise, if achieving a full automation is regarded as the highest priority, as taken in our approach, the formal proof of this implication can be "replaced" by a test. That is, we first generate sample values for variables in S_{pre} and P_{pre}, and then evaluate both of them to see whether P_{pre} is false when S_{pre} is true. If this is true, it tells that the path under examination contains an error.

For example, if we need to judge the validity of the implication "(age > 0 AND normal: = 10) → (a < 12 AND normal * 0.5 =~ normal^2 - ~normal)", use the test case (age, 6), (normal, 10) and we can easily prove the implication is not correct.

7. SUPPORTING TOOL

We have developed a prototype software tool to support the TBFV-M method. Specifically, it provides five major functions, which are functional scenario generation, test path generation, matching function scenarios to test paths, pre-condition derivation, verification of test paths, and output of verification result.

The tool interface is shown in Figure10. We can load specification and Activity Diagram. We simply use .txt file to store specification and notice that the specification file should guarantee the unified formal expression. We choose Enterprise Architect modeling tool to establish the system model. The Enterprise Architect modeling tool is powerful and supports the SysML model that will be created in Enterprise Architect and exported into XML format, so the Activity Diagram is described in XML file.



Specification: ../sepcification.txt

Activity Diagram: ./calculate.xml

Function Scenario

```
(1) Spre: c>0 AND b>=0 AND p=FALSE
    Ci: c<=b
    Di: r=c AND b=b-r
(2) Spre: c>0 AND b>=0 AND p=FALSE
    Ci: c>b
    Di: p=TRUE
(3) Spre^: c<=0 OR b<0 OR p=TRUE
```

Test Scenario

```
3. start -> s0 -> a0 -> s1 -> s2 ->a3 -> m0 -> a4 -> m1 ->
end
3. c>0 AND b>=0 AND p=FALSE AND c>b
(2)

4. start -> s0 -> a0 -> s1 -> s2 ->a1 -> a2 -> m0 -> a4 ->
m1 -> end
4. c>0 AND b>=0 AND p=FALSE AND c<=b
(1)
```

Verification Process

```
~c<=~b AND c=~c AND b-r=~b-r
a0: input c
~c<=~b AND c=~c AND b-r=~b-r
a1: r:=c
~c<=~b AND r=~c AND b-r=~b-r
a2: b:=b-r
~c<=~b AND r=~c AND b=~b-r
a4: output r
~c<=~b AND r=~c AND b=~b-r
```

Result

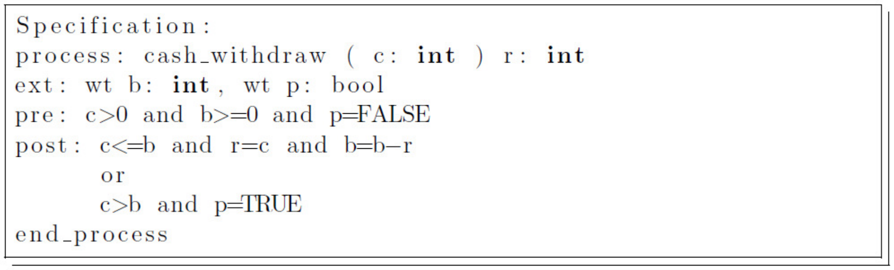
```
(1) PASS

(2) PASS

(3) PASS

(4) Verifying
```

Figure 10. Tool interface

The second step is deriving functional scenarios and generating test paths. And the tools interface is shown in Fig.10. The below two windows are used to display the intermediate outcome. When the user clicks on "match", the match result will refresh into the test paths window. And if it exists unmatched part, a popup will remind user to refine the model.

8. CASE STUDY

Now we show a motivation example to detail the process of MBSE and TBFV-M method described in the article above. First, we will get a requirement from the user, which consists of inform the description: “In a banking system, a money-withdrawing function needs to be realized. User input the required cash(c), if the cash is less than or equal to the balance(b), then the amount of money received(r), do not print information(p), the balance deducted the corresponding cash. Otherwise, user will not get money and the screen will print "insufficient balance"." This specification is formal and structured, as shown:



```
Specification:
process: cash_withdraw ( c: int ) r: int
ext: wt b: int, wt p: bool
pre: c>0 and b>=0 and p=FALSE
post: c<=b and r=c and b=b−r
      or
      c>b and p=TRUE
end_process
```

According to the specification, we can construct a set of SysML model and the Activity Diagram is shown below.

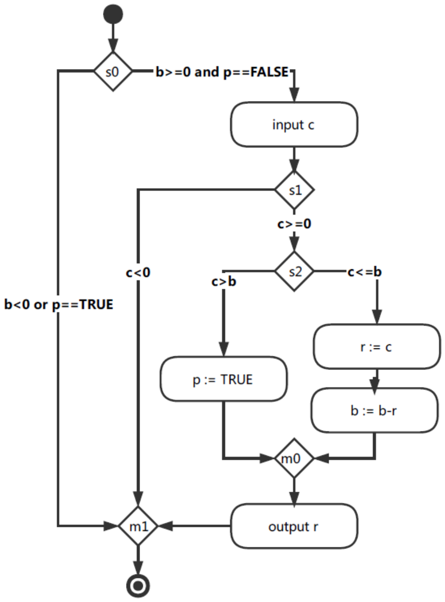
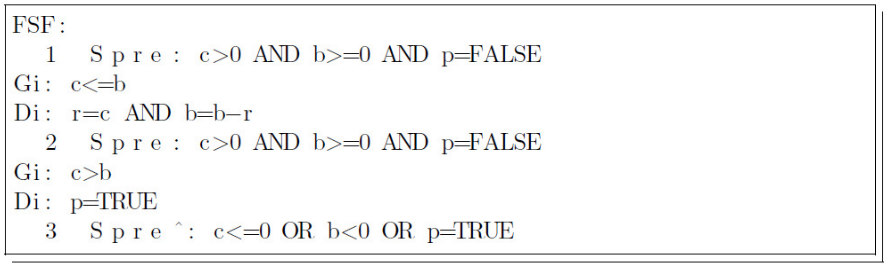
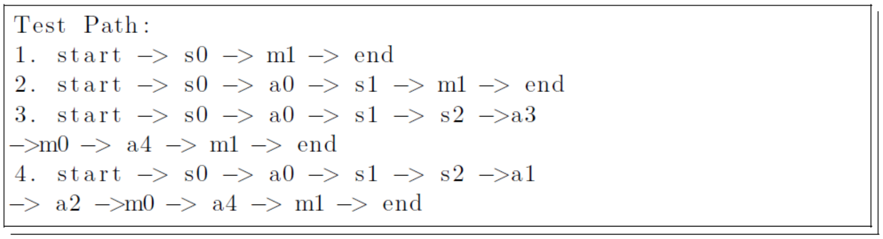


Figure 11. Activity Diagram

We can find the expression is described with SOLF. After getting ready with all the input, specification and Activity Diagram, we will start the TBFV-M method process. First, derive Functional Scenarios from specification and generate test paths from Activity Diagram. The result is shown as below.

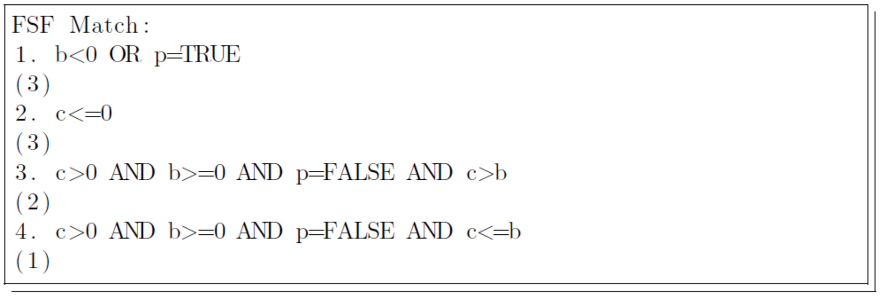


```
FSF:
   1  Spre: c>0 AND b>=0 AND p=FALSE
Gi: c<=b
Di: r=c AND b=b−r
   2  Spre: c>0 AND b>=0 AND p=FALSE
Gi: c>b
Di: p=TRUE
   3  Spre^: c<=0 OR b<0 OR p=TRUE
```



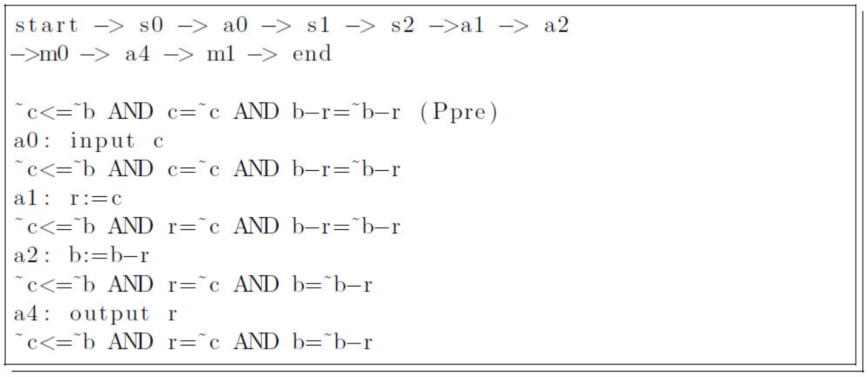
```
Test Path:
1. start -> s0 -> m1 -> end
2. start -> s0 -> a0 -> s1 -> m1 -> end
3. start -> s0 -> a0 -> s1 -> s2 ->a3
->m0 -> a4 -> m1 -> end
4. start -> s0 -> a0 -> s1 -> s2 ->a1
-> a2 ->m0 -> a4 -> m1 -> end
```

At the same time, we can extract data constraints from each test scenario, which is used for matching with functional scenario. Then, the matching process is shown below. If it does not exist a matched functional scenario, then it means that it exists a problem in the model, exactly in this unmatched test path. This path is not established accurately according to the requirements described in specification in the activity diagram model. If the match succeeds, it indicates that the test path is designed for the matched test scenario.



```
FSF Match:
1. b<0 OR p=TRUE
(3)
2. c<=0
(3)
3. c>0 AND b>=0 AND p=FALSE AND c>b
(2)
4. c>0 AND b>=0 AND p=FALSE AND c<=b
(1)
```

We will do the verification of test scenario according to the successfully matched functional scenario. First, we establish Path Triple and then apply the axiom of Hoare Logic to derive P_{pre}, pre-assertion of one path for the corresponding test path. The blow figure chose the forth path and matched the first functional scenario as an example and shows the substitution process, from bottom to up. So, the top one “~c≤ ~b AND c =~c AND b-r =~b-r" is the P_{pre}.



```
start -> s0 -> a0 -> s1 -> s2 ->a1 -> a2
->m0 -> a4 -> m1 -> end

~c<=~b AND c=~c AND b-r=~b-r (Ppre)
a0: input c
~c<=~b AND c=~c AND b-r=~b-r
a1: r:=c
~c<=~b AND r=~c AND b-r=~b-r
a2: b:=b-r
~c<=~b AND r=~c AND b=~b-r
a4: output r
~c<=~b AND r=~c AND b=~b-r
```

Finally, we turn this verification problem into proving whether the pre-condition of specification can imply P_{pre}. If it can be proved, means that the path satisfies the requirement. If not, there is a problem existing in the model, exactly in this unmatched test path. If the matched pre-condition can imply the corresponding P_{pre} of all the test paths in the model, then the model is satisfied with the user’s requirements.

From the above segment, we can see the implication (~c > 0 AND b ≥ 0 AND ~p = FALSE AND ~c ≤ ~b) → (~c ≤~ b AND c =~c AND b - r =~b - r) is true. This it means that the test path is satisfied with the corresponding functional scenario. We have proved all the test paths, due to the space limit, we omit further details.

9. CONCLUSION

We have presented an approach, known as TBFV-M (Testing-Based Formal Verification for Model), for requirement design error detection in SysML Activity Diagrams by integrating test cases generation and Hoare Logic. The principle underlying TBFV-M is first to derive functional scenarios from specifications and generate test scenarios from Activity Diagrams. Then match them and verify each test scenario according to the corresponding functional scenario. Hoare logic is used during the verification process. TBFV-M method solve the limitation of TBFV, not concerning about models and solved the problem of inconsistent, incomplete, and inaccurate models. It has advantage in reducing the probability of system error and shortening the developing time.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 26240008, and Defence Industrial Technology Development Program JCKY 2016212B004-2.

REFERENCES

[1] A. W. Wymore, Model-based systems engineering: an introduction to the mathematical theory of discrete systems and to the tricotyledon theory of system design. CRC Press, 1993.

[2] S. Friedenthal, A. Moore, and R. Steiner, “A practical guide to sysml,” San Francisco Jung Institute Library Journal, vol. 17, no. 1, pp. 41-46, 2012.

[3] T. Weilkiens,”Systems engineering with sysml/uml,” Computer, no. 6, p. 83, 2006.

[4] M. Shah, L. Chrpa, F. Jimoh, D. Kitchin, T. Mccluskey, S. Parkinson, and M. Vallati, “Knowledge engineering tools in planning: State-of-the-art and future challenges,” Computer, 01 2013.

[5] T. S. Vaquero, J. R. Silva, and C. J. Beck, \A brief review of tools and methods for knowledge engineering for planning scheduling," Computer, pp. 7-14, 2011.

[6] S. Liu, “Utilizing hoare logic to strengthen testing for error detection in programs,” Computer, vol. 50, no. 6, pp. 1-5, 2014.

[7] S. Liu and S. Nakajima, Combining Specification-Based Testing, Correctness Proof, and Inspection for Program Verification in Practice. Springer International Publishing, 2013.

[8] S. Liu, “A tool supported testing method for reducing cost and improving quality,” in IEEE International Conference on Software Quality, Reliability and Security, 2016, pp. 448-455.

[9] S. Liu, Testing-Based Formal Verification for Theorems and Its Application in Software Specification Verification. Springer International Publishing, 2016.

[10] S. Liu, A. J. Ofiutt, C. Hostuart, Y. Sun, and M. Ohba, “So: A formal engineering methodology for industrial applications,” IEEE Transactions on Software Engineering, vol. 24, no. 1,pp. 24-45, 1998.

[11] F. Raimondi, C. Pecheur, and G. Brat, “Pdver, a tool to verify pddl planning domains,” Computer, 2009.

[12] S. Marrone, F. Flammini, N. Mazzocca, R. Nardone, and V. Vittorini, “Towards model-driven v&v assessment of railway control systems,” International Journal on Software Tools for Technology Transfer, vol. 16, no. 6, pp. 669-683, 2014.

[13] F. Flammini, S. Marrone, N. Mazzocca, R. Nardone, and V. Vittorini, “Model-driven v&v processes for computer-based control systems: A unifying perspective," Computer, vol. 7610, pp. 190-204, 2012.

[14] F. Liang, W. Schamai, O. Rogovchenko, S. Sadeghi, M. Nyberg, and P. Fritzson, “Model-based requirement veri_cation : A case study,” in International Modelica Conference, Munich,Germany, 2012.

[15] R. Sasse and J. Meseguer, “Java+itp: A veri_cation tool based on hoare logic and algebraic semantics 1,” Electronic Notes in Theoretical Computer Science, vol. 176, no. 4, pp. 29-46, 2007.

[16] M. O. Myreen and M. J. C. Gordon, “Hoare logic for realistically modelled machine code.” In TOOLS and Algorithms for the Construction and Analysis of Systems, International Conference, Tacas 2007, Held As, 2007, pp. 568-582.

[17] J. Lasalle, F. Bouquet, B. Legeard, and F. Peureux, “Sysml to uml model transformation for test generation purpose,” Acm Sigsoft Software Engineering Notes, vol. 36, no. 1, pp. 1-8, 2011.

[18] A. Nayak and D. Samanta, “Synthesis of test scenarios using uml activity diagrams,” Software & Systems Modeling, vol. 10, no. 1, pp. 63-89, 2011.

[19] O. Oluwagbemi and H. Asmuni, “Automatic generation of test cases from activity diagrams for uml based testing (ubt),” Computer, vol. 77, no. 13, 2015.

[20] S. Khurshid and D. Marinov, “Testera: Speci_cation-based testing of java programs using sat, ”Automated Software Engineering, vol. 11, no. 4, pp. 403-434, 2004.

[21] S. Liu and S. Nakajima, “A decompositional approach to automatic test case generation based on formal specifications,” in International Conference on Secure Software Integration Reliability Improvement, 2010, pp. 147-155.

[22] S. Liu, T. Hayashi, K. Takahashi, K. Kimura, T. Nakayama, and S. Nakajima, “Automatic transformation from formal specifications to functional scenario forms for automatic test case generation," in New Trends in Software Methodologies, TOOLS and Techniques Proceedings of the Somet 10, September 29 October 1, 2010, Yokohama City, Japan, 2010, pp. 383-397.

[23] Kent and Stuart, Model Driven Engineering. Springer Berlin Heidelberg, 2002.

[24] M. Broy, K. Havelund, R. Kumar, and B. Steffen, Towards a Unified View of Modelling and Programming (Track Summary). Springer International Publishing, 2016.

[25] G. Gay, “Generating effective test suites by combining coverage criteria,” in International Symposium on Search Based Software Engineering, 2017, pp. 65-82.

[26] A. K. Joseph, G. Radhamani, and V. Kallimani, \Improving test efficiency through multiple criteria coverage-based test case prioritization using modified heuristic algorithm," in International Conference on Computer and Information Sciences, 2016, pp. 430-435.

[27] C. A. R. Hoare, “An axiomatic basis for computer programming,” Communications of the Acm, vol. 12, no. 1, pp. 53-56, 1969.

[28] R. W. Floyd, Assigning Meanings to Programs. Springer Netherlands, 1993.

[29] V. R. Pratt, “Semantical consideration on oyo-hoare logic," in Symposium on Foundations of Computer Science, 1976, pp. 109-121.

[30] Y. Yin, Y. Xu, W. Miao, and Y. Chen, \An automated test case generation approach based on activity diagrams of sysml," International Journal of Performability Engineering, vol. 13, no. 6, pp. 922-936, 2017.

AUTHORS

Yufei Yin
Master Student of East China Normal University
Exchange student in Hosei University

Prof. Dr. Shaoying Liu
High-Quality Software Engineering Lab
Department Faculty of Computer and Information Sciences
Hosei University

Prof. Dr. Yixiang Chen
Software and hardware co design technology and application, director of Engineering Research Centre
East China Normal University

A STUDY ON THE MANAGEMENT OF SEGREGATED BIOMETRIC DATA

YougJae Kim

Department of Business Administration, Korea Polytechnic University,
Prof. ph.D. Republic of Korea

ABSTRACT

This proposal of paper is the first international standard and research management technology of biometric data. By introducing and applying this international standards, everyone can acknowledge and approve are process for management of segregated biometric information. First, Safety requirement of segregated biometric information. Second, Growth and development of segregated biometric information. Third, Convenience promotion of financial transactions.

The demand for biometric authentication service is on a trend of significant increase owing to the advances in new technology such as the Internet of Things (IoT), artificial intelligence and others as well as the expanded use of fin-tech and electronic financing arrangements.

Biometric information is convenient because it is inherent and does not need to be remembered or stored, but leakage or misuse of the information could bring about serious problems because of its unique property. It is becoming more and more important to manage the information safely without the possibility of misuse or leakage. To store the information safely, it is effective to segregate the information into fragments in an un-usable form. Accordingly, we propose to standardize the international standard for Management of Segregated Biometric Information.

The purpose of paper is to propose to standardize the international standard of the distributed management technology of biometric data such as ISO, IEC, ITU and UNCEFACT

KEYWORDS

biometric information, registration process, authentication process, financial transactions, fin-tech, e-Banking

1. INTRODUCTION

This standard consists of the terminology, management and authentication processes needed to manage segregated biometric information. The biometric information used includes all biometric recognition technology through physical characteristics which can create encrypted biometric information such as fingerprints, veins, iris, face, voice, brainwaves, and electrocardiogram, and behavioral characteristics such as keyboarding or typing (key strokes), lip movement, eye blinking, walking, and hand gestures. This standard does not limit the types of biometric information and the precision of recognition of biometric information shall be considered by financial service providers when applying it to financial services.

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP - 2018
pp. 69–74, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80905

The proposed paper specifies in its scope the framework and process for management of segregated biometric information, including the definitions and terms, the management process and the authentication process.

First, the definitions and terms.

Second, the management process of segregated biometric information. Such as,

- Registration, deletion, inquiry and renewal process of segregated biometric information Third, the authentication process of segregated biometric information.
- Financial Institution Server Method
- Segregation Management Center Server Method

This proposal is the first international standard for the distributed management technology of biometric data.

By introducing and applying this international standards, everyone can acknowledge and approve are process for management of segregated biometric information.

- Safety requirement of segregated biometric information.
- Growth and development of segregated biometric information.
- Convenience promotion of financial transactions.

2. The Purpose and Scope of This Paper

2.1. The scope of paper

The objective of this paper is to help users by providing a way of the standardization and distribution of the International standards for the authentication method to verify the customers in non-face-to-face transactions by utilizing the biometric data registered by the customers split to and stored at multiple institutions for distributed management, the contributions can be made to the proactive response to the leakage of biometric data, prevention of the infringement of customer privacy and the activation of the biometric authentication and its related industries.

The scope of paper specifies a framework and process for management of segregated biometric information, including definition, management process of registration and authentication, deletion process, authentication module and distribute management of segregated biometric information.

First, registration process of segregated biometric information.

Second, authentication process of segregated biometric information.

- Financial Institution Server Method (Proprietary Authentication Tasks)
- Segregation management center Server Method (Commissioned Authentication Tasks)

Third, deletion process of segregated biometric information

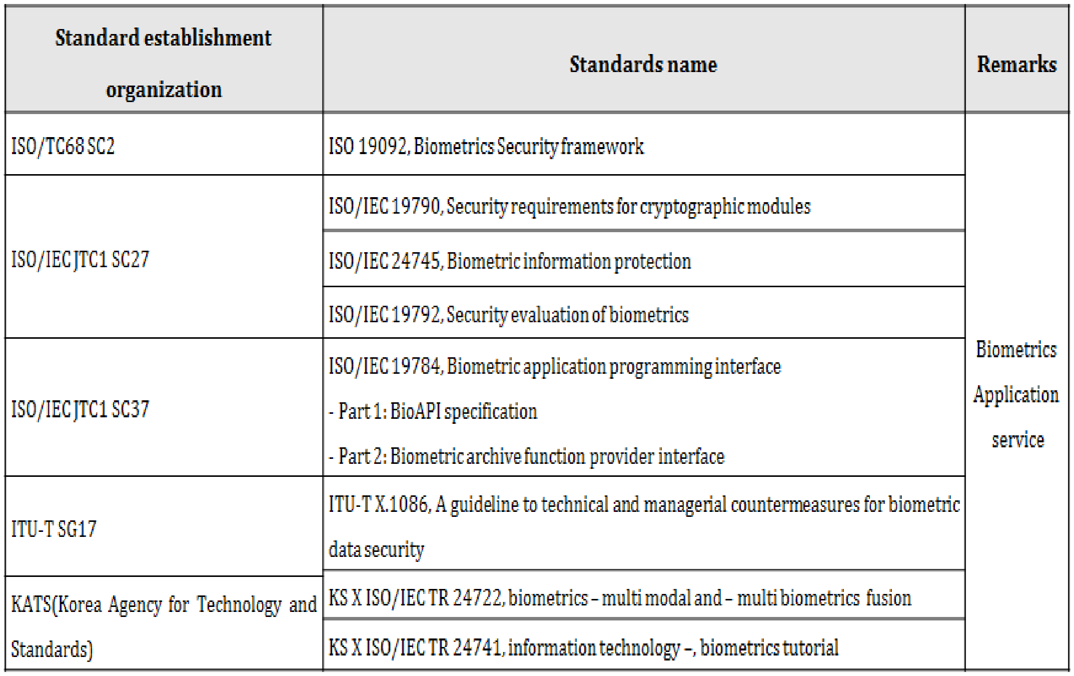
Fourth, authentication module of segregated biometric information

Fifth, distribution management of segregated biometric information

3. PREVIOUS STUDIES

The earlier study analyzed the previous studies and current state of international standards refers to the following latest versions.

Table 1. Current state of international standards



Standard establishment organization	Standards name	Remarks
ISO/TC68 SC2	ISO 19092, Biometrics Security framework	Biometrics Application service
ISO/IEC JTC1 SC27	ISO/IEC 19790, Security requirements for cryptographic modules	
	ISO/IEC 24745, Biometric information protection	
	ISO/IEC 19792, Security evaluation of biometrics	
ISO/IEC JTC1 SC37	ISO/IEC 19784, Biometric application programming interface - Part 1: BioAPI specification - Part 2: Biometric archive function provider interface	
ITU-T SG17	ITU-T X.1086, A guideline to technical and managerial countermeasures for biometric data security	
KATS(Korea Agency for Technology and Standards)	KS X ISO/IEC TR 24722, biometrics – multi modal and – multi biometrics fusion	
	KS X ISO/IEC TR 24741, information technology –, biometrics tutorial	

4. THE REGISTRATION AND AUTHENTICATION PROCESS FOR MANAGEMENT OF SEGREGATED BIOMETRIC INFORMATION

The registration and authentication Process consist of fifth process for management of segregated biometric information

First. Registration process of biometric information

Second. Authentication process of biometric information

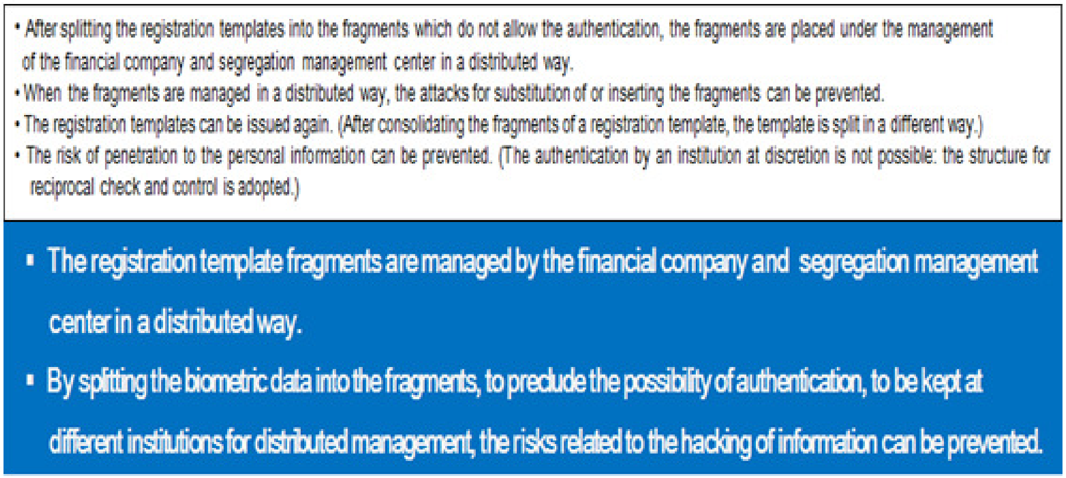
- Financial institution server method (Proprietary authentication tasks)
- Segregation management center server method (Commissioned authentication tasks)

Third. Deletion process of segregated biometric information

Fourth. Authentication module of segregated biometric information

Fifth. Distribution management of segregated biometric information

5.1. Registration process of biometric information

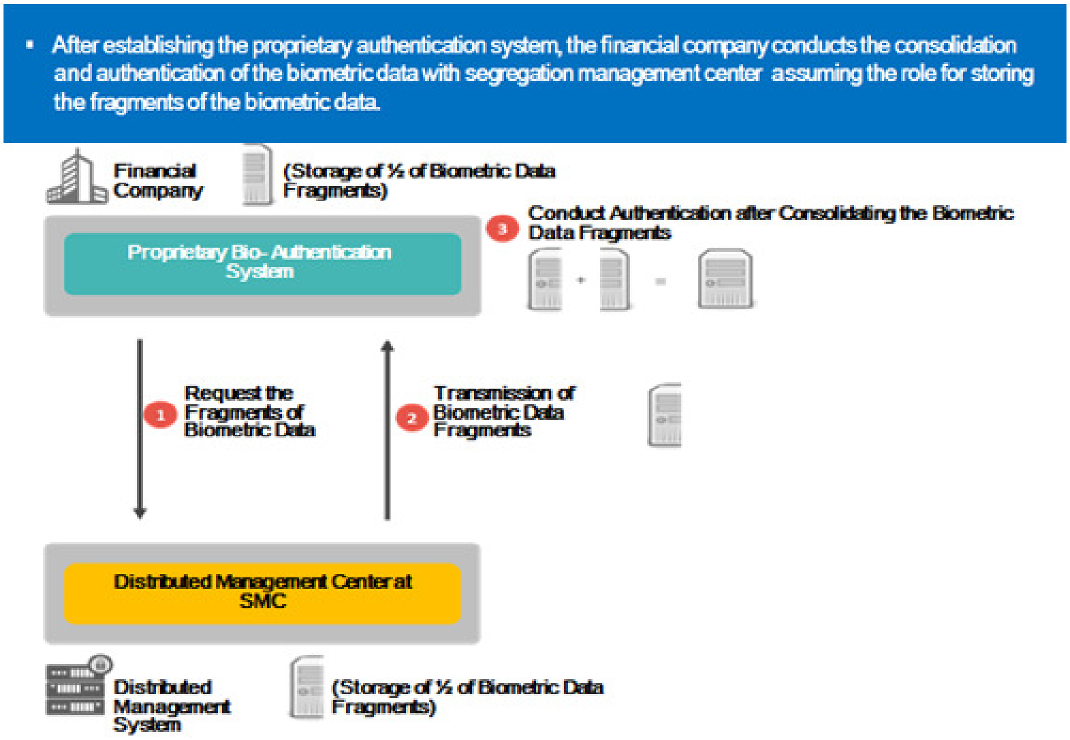


- After splitting the registration templates into the fragments which do not allow the authentication, the fragments are placed under the management of the financial company and segregation management center in a distributed way.
- When the fragments are managed in a distributed way, the attacks for substitution of or inserting the fragments can be prevented.
- The registration templates can be issued again. (After consolidating the fragments of a registration template, the template is split in a different way.)
- The risk of penetration to the personal information can be prevented. (The authentication by an institution at discretion is not possible: the structure for reciprocal check and control is adopted.)

- The registration template fragments are managed by the financial company and segregation management center in a distributed way.
- By splitting the biometric data into the fragments, to preclude the possibility of authentication, to be kept at different institutions for distributed management, the risks related to the hacking of information can be prevented.

5.2. Authentication process of biometric information

First, Financial Institution Server Method (proprietary Authentication Tasks)

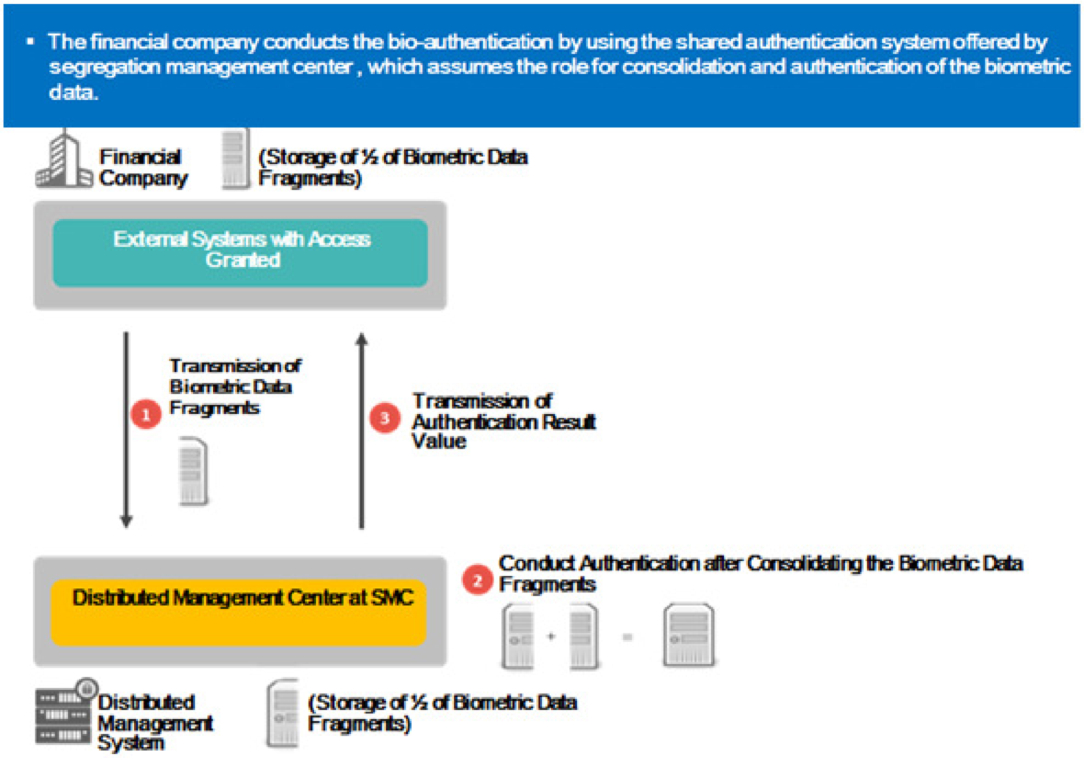


- After establishing the proprietary authentication system, the financial company conducts the consolidation and authentication of the biometric data with segregation management center assuming the role for storing the fragments of the biometric data.

Financial Company
(Storage of ½ of Biometric Data Fragments)
Proprietary Bio- Authentication System
3
Conduct Authentication after Consolidating the Biometric Data Fragments
+
=
1
Request the Fragments of Biometric Data
2
Transmission of Biometric Data Fragments
Distributed Management Center at SMC
Distributed Management System
(Storage of ½ of Biometric Data Fragments)

fig 3. Authentication Process

Second, Segregation management center Server Method (Commissioned Authentication Tasks)



- The financial company conducts the bio-authentication by using the shared authentication system offered by segregation management center , which assumes the role for consolidation and authentication of the biometric data.

Financial Company
(Storage of ½ of Biometric Data Fragments)
External Systems with Access Granted
1 Transmission of Biometric Data Fragments
3 Transmission of Authentication Result Value
Distributed Management Center at SMC
2 Conduct Authentication after Consolidating the Biometric Data Fragments
+
=
Distributed Management System
(Storage of ½ of Biometric Data Fragments)

fig 4. Financial Authentication process

6. EXPECTED EFFECT

First, the activation of biometric authentication as well as the development of the related markets and industries can be pursued by assuring the compatibility and security of the technology for distributed management of biometric data.

Second, enhancement of the possibility of overseas transplant of the domestic model for distributed management of biometric data.

Third, assurance of the initiatives in related discussions and preparation of the foothold for preoccupancy of the markets through the pre-emptive development of the related international standards in the global Fin-Tech and security markets.

7. CONCLUSION

This study of the implications and significance intends to draw its conclusion as follows.

First, the demand for the biometric authentication is increasing significantly owing to the advance in new technology such as the Internet of Things (IoT), artificial intelligence and others and the expanded use of Fin –Teck, and electronic financing.

Second, biometric information is a unique characteristic to identify individuals, such as fingerprints, faces, veins and irises, that is used for authentication in a wide range of fields such as financial transactions, e-banking, fin-tech, e-Business, distribution industry and etc. Biometric information is convenient because it is inherent and does not need to be remembered or stored, but leakage or misuse of the information could bring about serious problems because of its unique property. It is becoming more and more important to manage the information safely without the possibility of misuse or leakage. To store the information safely, it is effective to segregate the information into fragments in an un-usable form.

Third, by ensuring the safety of the biometric information stored in a distributed way, the financial institutions are not allowed to infer the sound biometric information of customers and the fragmented biometric information is not available for use even if the fragments are leaked.

Fourth, The developed framework of standard of distributed management technology of biometric data guide the international law, rule and system of standard, technical regulation, conformity assessment, inspection, test and certification such as ISO, IEC and ITU standards and guideline.

Fifth, the convenience of the standard of distributed management technology of biometric data can be enhanced in financial transactions, e-banking, fin-tech, e-Business, distribution industry and etc.

REFERENCES

[1] A Study of biometric Approach Using Fingerprint Recognition Lecture Notes on Software Engineering, Vol.1 No.2 May 2013 Ravi Subban and dattatreya P.Mankame page 211-213

[2] Biometrics Management Guidelines for the finacial security,TTA(Telecommunications Technology Association) 2016.12.02

[3] Ensuring Quality in Biometric Systems Md. Mahbubur Rahman, Amit Karmaker, Md.Mahmudul Hasan, Samsuddin Ahmed international Journal of Security and Its Applications 2015 September Vol.9 No 4

[4] ISO 19092, Biometrics Security framework, ISO/TC 68 SC2, 2015

[5] ISO/IEC 24745, Biometric information protection, ISO/IECJTC1 SC27. 2015

[6] ISO/IEC 19792, Security evaluation of biometrics, ISO/IECJTC1 SC27. 2015

[7] ISO/IEC 19784, Biometric application programming interface, ISO/IEC JTC1 SC37.2015

[8] ITU-T X.1086, A guideline to technical and managerial countermeasures for biometric data security, ITU-T SG17

[9] Segregated management of biometric data information, Korea payment settlement association Park Jung Gook 2016.12.05

[10] Standard of segregated management biometric data, ISO/TC 68 SC2 Presentation, 2018.04.25

[11] "A novel way of ICON based authentication methods", Advance Computing Conference (IACC) 2015 IEEE International, P Devaki, Raghavendra Rao, pp. 449-453, 2015.

Authors

YounJae Kim

Short Biography
ISO/IEC SC 32 WG1 Vice Convenor
ISO TC 68(Financial Service)
Uncefact International Trade and Business Group Vice Convenor

A COMPARISON OF REAL-TIME TASK SCHEDULING METHODS IN SPACECRAFT SIMULATION

Mehmet Emin Güllüoğlu[1] and Mehmet Reşit Tolun[2]

[1]Department of Computer Engineering, Baskent University, Ankara, Turkey
TAI, Turkish Aerospace Industries, Ankara, Turkey
[2]Department of Computer Engineering, Baskent University, Ankara, Turkey

ABSTRACT

Today, embedded real-time applications play an important role in modern life. Satellites are also robust embedded real-time applications. A satellite project can cost over three-hundred million dollars. As many satellite manufacturers validate their satellites before launching, satellite simulators play the most valuable role in validation infrastructures. Specifically, satellite flight software validation has become more important. In this paper, we focused on the round robin (RR), rate monotonic (RM), and event driven (ED) real-time scheduling task methods with respect to their CPU usage performance for satellite simulator infrastructures. The tasks are evaluated and tested by real-time executive for multiprocessor systems (RTEMS). Those scheduling tasks are used in polling mode in the simulation setup. In this study, we compared three task scheduler methods for attitude orbit control system tasks and MIL-STD 1553 bus data distribution controller tasks in a spacecraft simulator environment. The results were close and the values were not segregated, thus, we chose RR and ED, because RR was easy to implement and ED allowed for full control of the tasks.

KEYWORDS

Real-time embedded systems, Real-time operating system, Rate monotonic task, Round robin task, Event driven task handling, and Satellite simulations

1. INTRODUCTION

Spacecraft development has been remarkably changed and optimized by modern simulation methods. Spacecraft manufacturers need to be sure the spacecraft can achieve its mission. Satellite simulators play the most valuable role in validation infrastructures. Hence, satellite simulation is an important issue in space craft simulations, such as how well you can simulate your system, based on models and the controller. Simulation frameworks are provided to run models. The satellite equipment, space environment, and satellite dynamics have to be represented by the models. The aims of a consummate simulation contain the representative models that are given at below:

- Satellite equipment models
- Space environment model
- Satellite dynamic model

The general purpose of running satellite models in a simulation framework is spacecraft flight software validation.

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP, CMCA - 2018
pp. 75–84, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80906

Spacecraft software is run on real-time embedded system called an onboard computer (OBC). All control algorithms are run from there and it handles data management via a uniprocessor. The spacecraft controller is run on a real-time operating system (RTOS) on the OBC. The RTOS was provided with a task scheduler method to use our flight software. Task optimization and performance is directly related to the flight software performance. Therefore, the aim of this study was to compare the real-time task scheduler method in a spacecraft simulation.

In our study, we focused on two controllers, the 1553 bus data distribution controller (1553) and attitude orbit control system (AOCS), and the three task scheduler methods chosen for comparison were:

- Round Robin (RR)
- Rate Monotonic (RM)
- Event Driven (ED)

Tasks for the 1553 and AOCS were run for the three scheduler methods.

2. THE SPACECRAFT SIMULATION

Space mission projects are unique and require a big budget, thus, it must be validated for a successfully mission life before launching. Simulation is the most used process method to validate equipment and verify the satellite system's level of requirement. Validation infrastructures vary according to their purpose.

2.1. Functional Verification Bench (FVB)

For most spacecraft projects, this is used for limited algorithm verification of the AOCS. The OBC is not represented in this model; only the control algorithms are designed and its functionality is tested (Figure 1) [1]

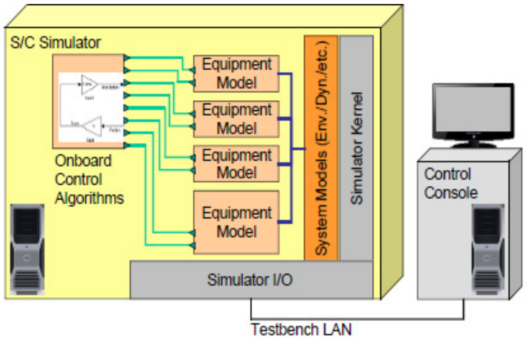


Figure 1 Functional Verification Bench

2.2. Software Verification Facility (SVF)

This facility is used for verification of the interfaces, connections, and standards to be sure that they work with each other properly. The OBC could be a model or emulator. The emulator is a perfect replica the OBC (Figure 2). [1]

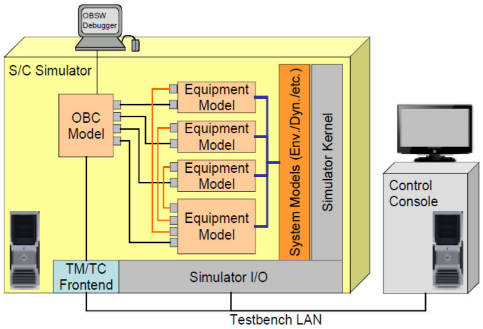


Figure 2 Software Verification Facility

2.3. Hybrid Verification Infrastructure

The hybrid verification infrastructure generally meets the real hardware OBC for verification of the satellite software on the real hardware. This infrastructure has the advantage of a directly tested I/O interface (Figure 3). [1]

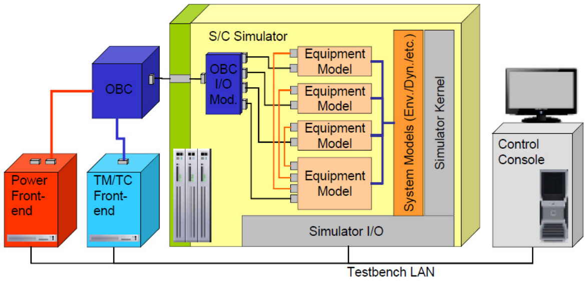


Figure 3 Hybrid Verification Infrastructure

2.4. Avionic Test Bench (ATB)

The ATB is the most improved simulation for the satellite. Verification engineers carry out their design. This bench can be added not only the equipment model, but also the real equipment that is needed to project the constraints (Figure 4). [1]

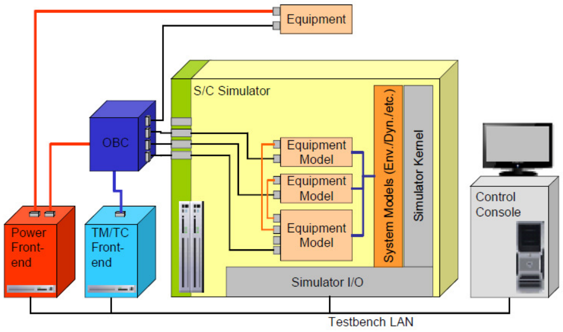


Figure 4 Avionic Test Bench (ATB)

2.5. Dynamic Satellite Simulator (DSS)

This simulator is generally used for telemetry and tele-command verification. Satellite operators are always educated about the DSSs, as it is very important in the launch and early orbit phase

(LEOP), because it aids in satellite survival. When the LEOP is not pre-worked, it causes satellite loss. The DSS provides pre-work before the LEOP, lunching, or nominal operations (Figure 5). [1]

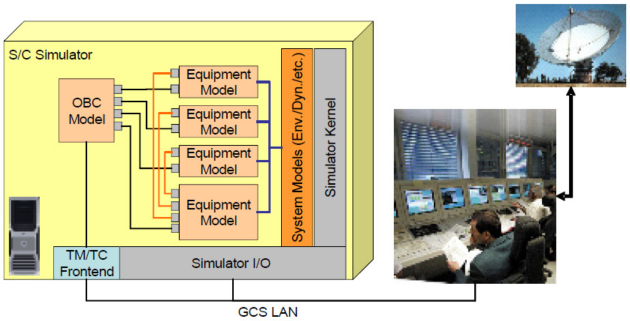


Figure 5 Dynamic Satellite Simulator (DSS)

3. RTEMS (Real-Time Executive for Multiprocessor Systems)

RTOSs run in the OBC a part of the flight software. Hard RTOSs provide a restricted time in case the safety critical systems use hard RTOSs. Satellites are safety-critical systems and any unexpected delays may cause a catastrophic output, which can result in satellite loss.

RTEMSs are the most useful hard RTOSs on satellites. They provide high-performance peripherals for embedded real-time systems, offering characteristics to support the development of real-time embedded applications that are available for different platforms and architectures, including SPARC leon3 FT, which is one of the architectures used in this study [6]
RTEMSs provide the following features:

- multitasking
- homogeneous multiprocessor systems
- heterogeneous multiprocessor systems
- interrupt management
- event-driven inter-task communication, priority-based, pre-emptive
- rate monotonic scheduling
- synchronization and inter-task communication
- high level of user configurability
- dynamic memory allocation
- priority inheritance

The internal architecture of RTEMSs can be observed as layered components that work in harmony to provide a set of services for a real-time application system. (Figure 6) [6]

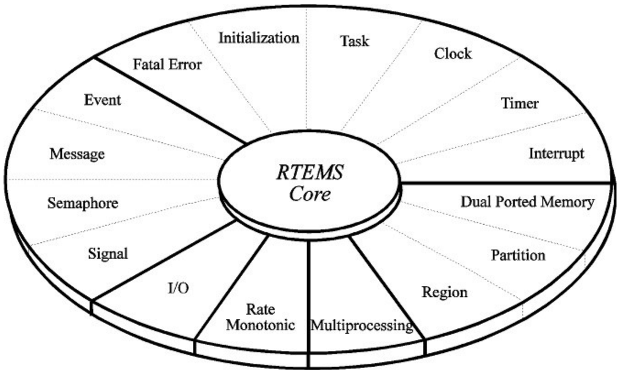


Figure 6 RTEMS resource managers

The functions utilized by multiple managers, such as scheduling, dispatching, and object management, are provided in the executive core. In this study, we focused on task manager services, rate monotonic services, and event services. [6]

4. STUDY ENVIRONMENT

An important issue in space craft simulations is how much simulations are in your system, based on models and the controller. The system must be close-looped and the controllers must be feel the same as in orbit. Hence, we developed the test environment of a hardware OBC, simulation framework, and MIL-STD 1553 data bus to meet the requirements of a spacecraft (Figure 7)

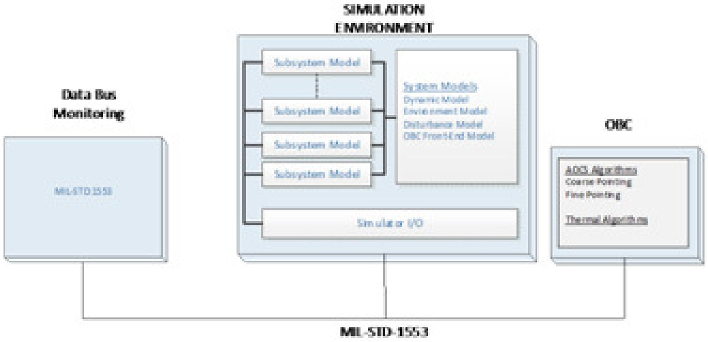


Figure 7 Test Environment

4.1. On Board Computer (OBC)

The OBC is the flight computer of a satellite. All of the control algorithms are run from there, and it handles data management via a uniprocessor [3] SPARC leon3 FT, which is especially designed for space missions. [4]

The LEON3 is a VHDL model of a 32-bit processor that is synthesizable and can accommodate IEEE-1754 (SPARC V8) architecture. LEON3 is an addendum of the LEON2 processor, with a 7-stage pipeline (compared to LEON2's 5-stage pipeline), which supports asymmetric and symmetric multiprocessing (AMP/SMP). A multiprocessing configuration of as much as 16 CPU can be utilized. The LEON3 is a highly-configurable model, which is appropriate for system-on-chip (SoC) designs, featuring the following: [5]

- A SPARC V8 instruction set equipped with V8e extensions,
- A sophisticated 7-stage pipeline
- Advanced on-chip debugging support, equipped with instructions and a data trace buffer
- A local instruction and data scratch pad RAM of 1 to 512 kilobytes
- Robust and fully-synchronous single-edged clock design, high-performance: 1.4 DMIPS/MHz, 1.8 CoreMark/MHz (gcc –4.1.2),
- A wide range of software tools, such as compilers, kernels, simulators, and debugging monitors.
- An AMBA-2.0 AHB bus interface
- Up to 125 MHz in FPGA and 400 MHz on 0.13 um ASIC technology

In the test setup, the OBC featured the following:

- LEON-3 FT core
- RTEMS operating system (OS)
- MIL-STD-1553, CAN, Ethernet
- Control Algorithm Integration:
  - AOCS Algorithm
  - Data Management
  - Sensor Data Analysis
  - Actuator Control
  - Mode management

4.2. MIL-STD 1553 Bus

The Mil-Std-1553B or Milbus has the standard defining characteristics of a serial multiplex data bus. The standard is a set of requirements covering the mechanical, electrical, and functional aspects of the bus. The bus aims at interconnecting via a single-medium avionics subsystem. [6]

MIL-STD-1553 includes three kinds of bus users, known as terminals, including a bus controller (BC), remote terminal (RT), and bus monitor (BM). The bus transaction is a command/response. The BC behaves like a master and begins all of the transactions. The RTs, controlled by the BC, supply the interface between the 1553 bus and the appropriate unit/sub-system. The BM remains passive and is a bus traffic recorder. [6]

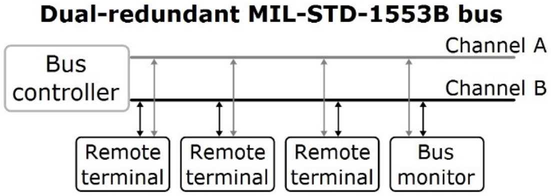


Figure 8 MIL-STD 1553B bus

4.3. Simulation Framework

The simulation framework provides the equipment and environment models in the polling mechanism. It also features the full scope of the problems to be investigated, ranging from the design and to the overall system simulations for the analyses of the dynamic system operation [1]. In our study, the simulation framework was run at ten Hz. The aims of the consummate simulations as contained representative models are given below:

- Star tracker model
- Sun sensor model
- Magnetometer model
- GPS model
- Magnetic toque bar model
- Reaction wheels' model
- Propulsion model
- Communication subsystem model
- Power subsystem model
- Space environment model
- Satellite dynamic model

5. TASK SCHEDULERS

Space craft flight software must include both robustness as well as hard real-time. Moreover, the OS should manage to tasks properly. In space craft flight software, many tasks are used separately in the control flight. The flight software's performance is related to the overall performance of the tasks; therefore, the task scheduler should operate at optimum performance for the flight controller. We implemented three different task scheduler models; round robin (RR), rate monotonic (RM), and event driven (ED). Static priority-driven pre-emptive [3] approaches [7] were also employed in our study.

In our study, all of the tasks were identified as the highest priority. RTEMS OS layers (Figure 9) is showed in the Figure 9. The AOCS and 1553 bus controllers were appointed as the test tasks that would be used to compare the CPU performances. We focused on two controllers: the 1553 bus data distribution controller (1553) and AOCS, as well as three task scheduler methods chosen for comparison, which were RR, RM, and ED.

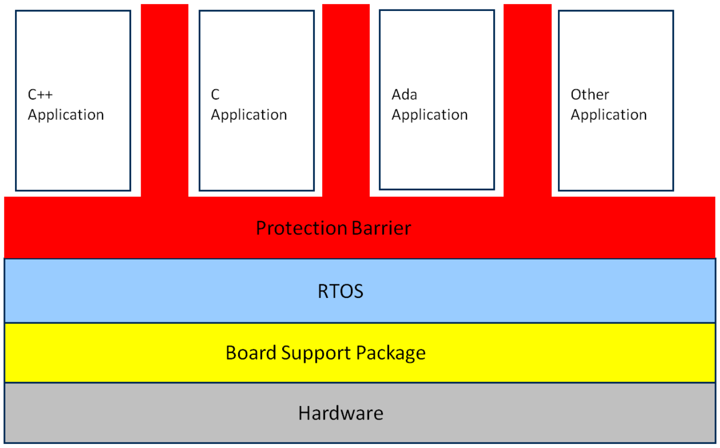


Figure 9 RTOS layer

5.1. Round Robin

RR scheduling is a task scheduling algorithm that is designed to be equitable. It uses time slices, also known as time quanta, which are assigned to each task in the queue. Each task is allowed to use the CPU for a given amount of time, and if it does not finish within the allocated time, it is pre-empted, and then returned to the back of the queue, so the next process in the queue can use the CPU for the same allocated time. [8]

5.2. Rate monotonic

The RM manager supplies facilities for implementing tasks that take place in a periodical manner. It also collects data regarding the implementation of those periods and is able to contribute relevant statistics that can be used when analysing and tuning the application. [2] [8]

5.3. Event driven

The ED is associated with interruptions and event managers that provide a high-performance method of inter-task communication and synchronization. The first task is invoked by interruptions from external devices, and after that, other tasks are invoked by previous tasks. Overload windows are able to miss deadlines; therefore, task times must be pre-calculated and those times must fit into a specific window; otherwise, the deadlines will exceeded [9].

6. DISCUSSION

Herein, we implemented three task schedulers, the RR, RM, and ED. The working frequency was appointed as ten Hz (100 ms). The AOCS and 1553 bus controller were appointed as the test tasks that would have their CPU performances compared. During the tests, every scheduler was run 1000 times and the tasks were identified as the highest priority [7]. The metric of performance was established as the “RTEMS CPU usage service” [8] for comparing the task performances. The results are given in the Table.

Table 1 CPU Usage (%)

	RR%	RM%	ED%
Idle	83.989	82.195	83.342
AOCS	15.727	17.495	16.234
1553	0.001	0.001	0.108

7. CONCLUSION

In this paper, we compared three task scheduler methods in a spacecraft simulator environment. The result were close and the values were not segregated. In this case, the task scheduler methods were investigated and we learned which task was suitable for the study environment and study cases. For example, RR was based on time slicing, so, if RR was chosen, the case should use preemptive, numerous tasks at a short rate. RM [2] always follows the rule of the shorter a task's run period, the higher its priority, or the longer a task's run period, the same is its priority [4]. Hence, the task period of the AOCS was longer than task period of the 1553 with the same priority, so the AOCS was run first. ED provides full control of the task run. If it is chosen, the task designer decides which is first, which is next, and which is last. Hence, we decided to run the 1553 task first and run the AOCS after that. In this study, we chose RR and ED because RR was easy to implement and ED provided full control of the tasks. Future work, we will add one more task, known as the thermal task and we will compare stack (memory) size of tasks.

REFERENCES

[1] J. Eickhoff, "Simulation Tools for System Analysis and Verification" in Simulating Spacecraft Systems, 1st ed., New York, 2009.

[2] C. Liu and J. Layland, "Scheduling algorithms for Multiprogramming in a Hard Real-Time Environment", Journal of the ACM, 20(1), 1973, pp. 46–61.

[3] F.F. Lindh, T. Otnes, and J. Wennerstrom, "Scheduling algorithms for real-time systems", Sch. Comput. Queen's Univ. Tech., 2005.

[4] M. Bashiri S. Ghassem "Performability Comparison of Schedulability Conditions in Real-Time Embedded Systems", 2010 Third International Conference on Dependability, 2010, pp 70–75.

[5] http://www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Computer_and_Data_Handling/Microprocessors.

[6] https://www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Computer_and_Data_Handling/Mil-STD-1553.

[7] K. Ramamritham, J.A. Stankovic "Scheduling Algorithms and Operating Systems Support for Real-Time Systems", IEEE Xplore, 1994.

[8] RTEMS C User's Guide (https://docs.rtems.org/releases/rtems-docs-4.11.2/c-user/index.html).

[9] M. Coutinho, J. Rufino, and C. Almeida. "Control of event handling timeliness in RTEMS". In: Proceedings of the 17th IASTED International Conference on Parallel and Distributed Computing Systems - PDCS 2005, Phoenix, Arizona, USA, 2005. IASTED.

AUTHORS

Mehmet Emin Güllüoğlu is Spacecraft Flight Software System Engineer at Turkish Aerospace Industries (TAI) where he has been since 2007.From 2007 to 2013 he worked at TAI as an Avionic system engineer. From January 2013 to January 2014 he worked at Thales Alenia Space – Cannes, France as an AIVV engineer. From January 2014 to October 2014 he worked at Thales Alenia Space – Cannes, France as a Functional Chain Validation engineer. From October 2014 to today he worked at TAI eventually as a Flight software engineer. His research interests is RTOS task schedulers, Spacecraft simulations and Software integration.

A WEB-BASED SOLUTION FOR POWER QUALITY DATA MANAGEMENT

Claudiu Popirlan[1], Gabriel Stoian[1], Leonardo Geo Manescu[2], Denisa Rusinaru[2], Marian Ciontu[2], Gabriel Cosmin Buzatu[2], Miron Alba[3], Adrian Cojoaca[3]

[1]Computer Science Department & INCESA Research Hub for Applied Sciences
University of Craiova, Romania
[2]Faculty of Electrical Engineering & INCESA Research Hub for Applied Sciences
University of Craiova, Romania
[3]Oltenia Distribution SA, Craiova, Romania

ABSTRACT

In this paper we present a web-based solution designed to convert (integrating an appropriate plugin) heterogeneous information provided by any type of power acquisition equipment into standard formats as PQDIF (Power Quality Data Interchange Format - IEEE® Std 1159.3-2003 standard). The power grid operators are interested for this kind of applications capable to convert huge volume of heterogeneous information into standard formats in order to be easily processed.

KEYWORDS

Web-Based Solution, Java Enterprise Application, Power Quality, Data Conversion

1. INTRODUCTION

The energy quality data management system provides a deterministic way to take the gross measurements from the points of interest (PoI), and then process them to obtain a unitary representation format, independent of the measuring equipment used. The system also provides users with a simple and easy-to-use interface as well as easy access to collected data for existing power quality analysis applications (such as PQView, see [9], [10]). The architecture of the energy quality management system is shown in Figure 1. This architecture has been described in detail in our previous papers (see [4], [5]).

Points of Interest (PoI) are those locations within the electrical distribution networks where specific equipment for the acquisition of electrical quantities is installed. These data may refer to energy quality if these points of interest are defined as part of an energy quality assessment system / process, but may also refer to other cases: post-incident analysis or simple monitoring of

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP, CMCA - 2018
pp. 85–95, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80907

some electrical quantities. Depending on the installed purchase equipment, data can be collected from them either manually or automatically ready for the next step: loading into SYMPQD.
To reduce the risk of error, users have an extremely convenient way to upload data: the web application "PQDM WebApp". It runs across an application server and provides graphical interface with simple methods to characterize data sets before they are actually loaded into the system. Also, "PQDM WebApp" is responsible for organizing the resulting files according to the needs of their users.

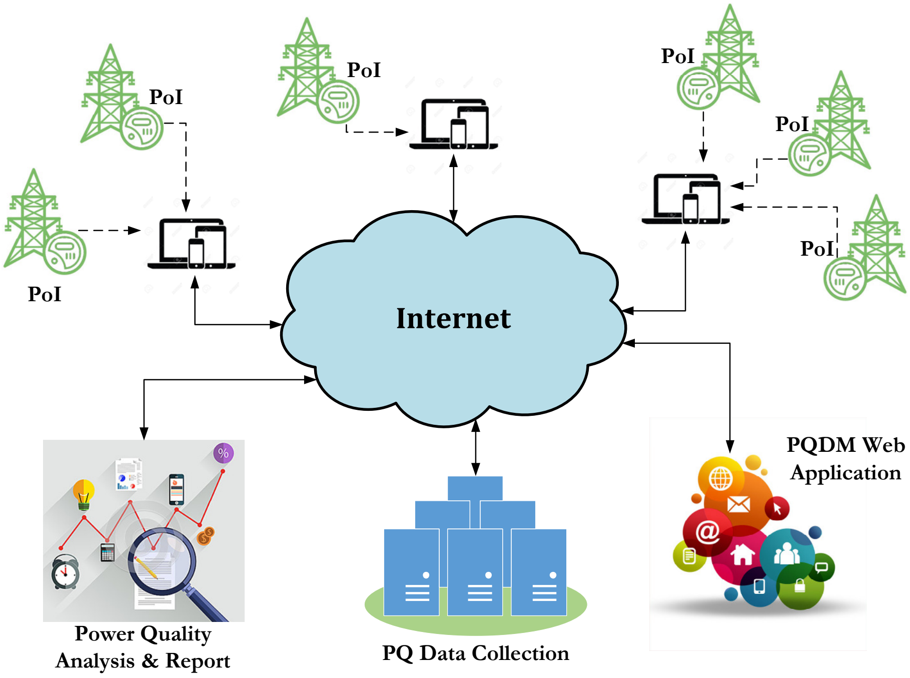


Figure 1. The system architecture of the energy quality management

The user interface is one of the greatest advantages of "PQDM WebApp" because it hides all the details on how to process, internalize and organize the collected data. Beyond this interface, the application is designed to be easily expandable when a new acquisition device is supposed to be supported by SYMPQD. Thus, any such equipment will be analysed to determine its particularities and will then be integrated into SYMPQD by implementing a plugin capable of processing the data collected from it. In this way, the system becomes easy to extend and easy to maintain.

Another strong point of SYMPQD is choosing PQDIF (see [1], [2], [3]) as a format for representing the collected data. PQDIF (Power Quality Data Interchange Format) is a binary file format defined and standardized by the "IEEE 1159.3 PQDIF Task Force" and is the most appropriate way to transfer energy quality data between different software applications of interest. Also, PQDIF is adopted by the overwhelming majority of energy quality analysis and reporting systems, such as PQView (see [9], [10]).

2. WEB-BASED SOLUTION IMPLEMENTATION TECHNOLOGIES

From the point of view of implementation, technologies have followed the natural steps of any software development. We started by performing experiments, measurements, and processing data collected using the Matlab software suite [14]. So we could have access to simple and extremely powerful tools for validating data structures specific to energy systems. Later on for ease of use we opted for a client-server solution where we preferred Java-based technologies [15]. Matlab-based tools have continued to be used as test and validation tools for the resulting files. Using Java, we have benefited from all the benefits of Oracle J2EE [15] that provide robustness, platform independence, efficiency and the ability to use a variety of application servers: Oracle GlassFish [17], Apache Tomcat [16], Eclipse Jetty [18], etc.

PQDM WebApp stores the resulting files in the PQDC (Power Quality Data Collection) location. The PQDC location may be the same as the web application, or it may be different by getting a distributed solution. As a rule, smaller beneficiaries will choose the first option, while large beneficiaries with important IT resources will turn to the distributed version. Regardless of the choice chosen, PQDC offers a unique place where all information on the quality of electricity can be obtained by specialized analysis and reporting software packages.

By choosing PQDIF as the format for the resulting files, we offer our customers the possibility to use a wide range of software solutions for energy quality analysis. Also, by implementing the architecture presented in Figure 1, we significantly reduce the risk of human errors in the processing and organization of collected data, and offer a modern, flexible, scalable, portable and distributed solution.

Implementation technologies and development tools are open, which means we have minimal costs.

2.1. Technical description of PQDM WebApp

At the current stage of the development, the PQDM-WebApp application is developed using Java technologies (https://www.oracle.com/java/index.html), using the NetBeans (https://netbeans.org) development environment (IDE), and is intended to run in the Web.

Because of its functionalities, PQDM-WebApp is classified as a web application, which means that both the server and the client components are working collectively to transcend information from the technical level to a level intended for a much wider audience.

To achieve this goal, java technology was chosen, considering that Oracle J2EE (http://www.oracle.com) offers a lot of advantages: robustness, platform independence, efficiency and the possibility to use a wide range of server applications like Apache Tomcat [16].

The application is based on a three-tier architecture: a presentation tier, a domain logic tier, and a data storage tier. By segregating the application into tiers, offers the advantage to modifying or adding a specific layer, instead of reworking the entire application. The architecture of the web application is given bellow in Figure 2.

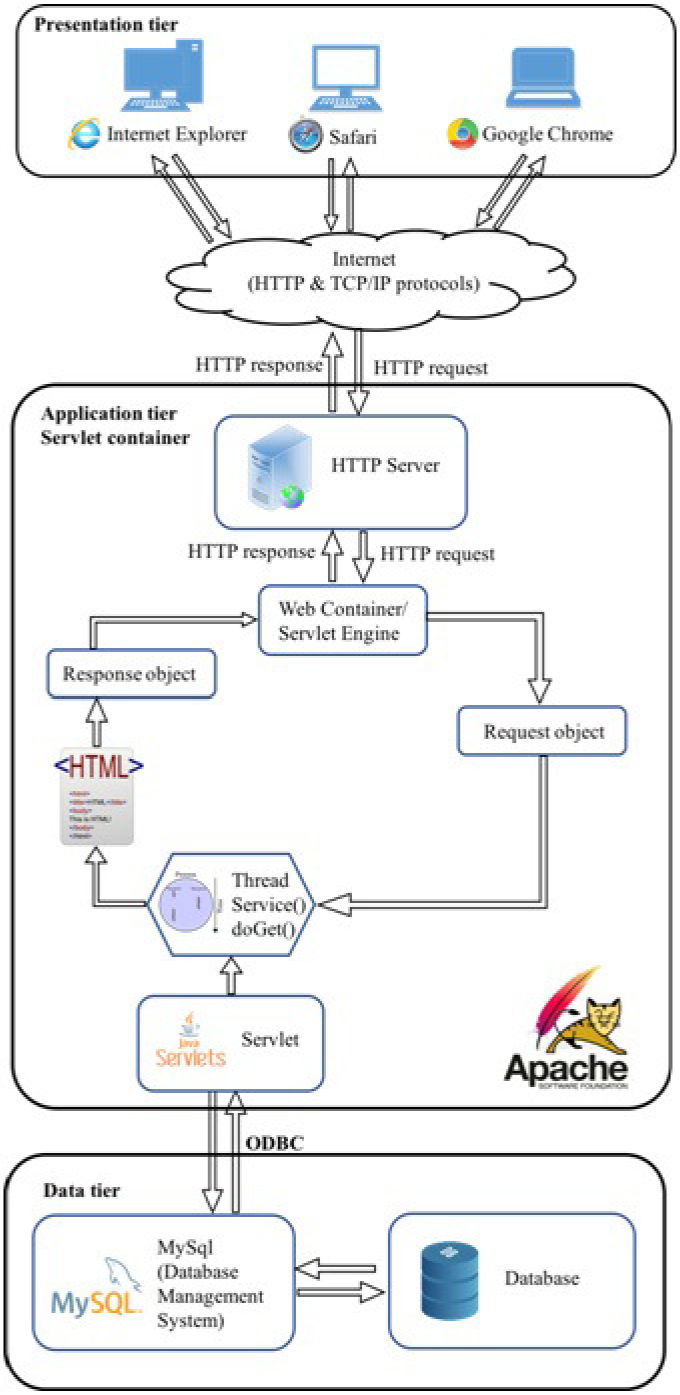


Figure 2. The PQDM-WebApp architecture

This three-tire client-server software architecture pattern with its functionalities, and their subcomponents determined by the chosen technologies are as following:

1. Presentation tier: The topmost level application is the user interface. The main function of the interface is to translate tasks and results the user can understand and access directly

(such as a web page, or an operating system's GUI). In this case, the tire is composed of the web browsers: Internet Explorer, Safari, Google Chrome, etc.

2. Application tier: This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers. By choosing to use java technology, a suited environment to run the code is Apache Tomcat, which is an open-source Java Servlet Container.

3. Data tier: Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

A short description in a deductive way of the terms that are further used, is given. Abstract data type is a data type defined by the operations that can be applied on them, and are implemented through a class, where operations are called methods or functions. An object is an instance of a class and represents what actually is running on a computer. A container is an object which contains multiple objects.

Apache Tomcat server operates according to the principle of request-response model. A request is send when from the client level is demanded a web page that exists in the context of the server. By executing the corresponding Java Servlet class, a response is generated, which consists in a HTML document (HyperText Markup Language).

Java Virtual Machine is created when the Apache Tomcat server is started and destroyed when exited. Its main purpose is to enable a computer to run a Java program.

Inside the Java Virtual Machine is running a single server which represents the entire Tomcat instance. Its purpose is to manage the life cycle of its contained services. A server can contain one or more Services, where each Service represents the set of request processing components within Tomcat.

The service object is a structural element that combines one or more Connector components and a single Catalina servlet Engine element.

Giving the dynamic nature of the application, Tomcat is configured to work in conjunction mode with a separate HTTP web server, therefore two connectors are used:

- The HTTP/1.1 Connector manages the communication between the client and the web server through the HTTP protocol (HyperText Transfer Protocol), which listen on port 8080. If a HTTP request is needed to be routed to the Tomcat container for processing, the web server will communicate this request using the AJP protocol (Apache JServer Protocol), which is a binary protocol more efficient than the text-based HTTP.

- The AJP/1.3 Connector, placed inside the service, is listening on port 8009 for requests, and translate them into a request object which Catalina engine can process.

The Catalina container's components that are responsible for processing those requests and generate response objects are: the engine, virtual host, and context.

The Host component is a container for web application(s) or context(s), and it is defined by two fundamental concepts:

(1) Domain name: this is the value expected to be sent by the client browser (http://localhost:8084/PQDM-WebApp);

(2) Application base name: in this folder is located the context that will be deployed to this host. The context or web application (PQDM-WebApp) is the location where the application specific servlet and JSP (JavaServer Page) and their associated files live.

A Wrapper object is a child of the context container and represents an individual servlet (or a JSP file converted to servlet).

In the life cycle of a servlet, three methods are fundamental: *init()*, *service()* and *destroy()*, which are implemented by every servlet and are invoked at specific times by the server. The course of action of these three methods are in the following order:

1. During initialization stage of the servlet life cycle, the web container initializes the servlet instance by calling the *init()* method, passing an object implementing the *javax.servlet.ServletConfig* (https://docs.oracle.com) interface. This configuration object allows the servlet to access name-value initialization parameters from the web application.

2. After initialization, the servlet instance can service client request. Each request is serviced in its own separate thread. The web container calls the *service()* method of the servlet for each request. The *service()* method determines the kind of request being made and dispatches it to the appropriate methods: *doGet()*, *doPost()*, *doPut()*, *doDelete()*, and so on; according to the HTTP request.

3. Finally, the web container calls the *destroy()* method that takes the servlet out of service. The *destroy()* method, like *init()*, is called only once in the lifecycle of a servlet.

A thread represents the actual execution of a servlet class deployed by the processor to generate a response to the request. During the class execution, in order to generate the response object, it has to interact with the data tire passing a set of SQL (Structured Query Language) queries to the Database Management System (DBMS). The access of DBMS is realized through an ODBC (Open Database Connectivity).

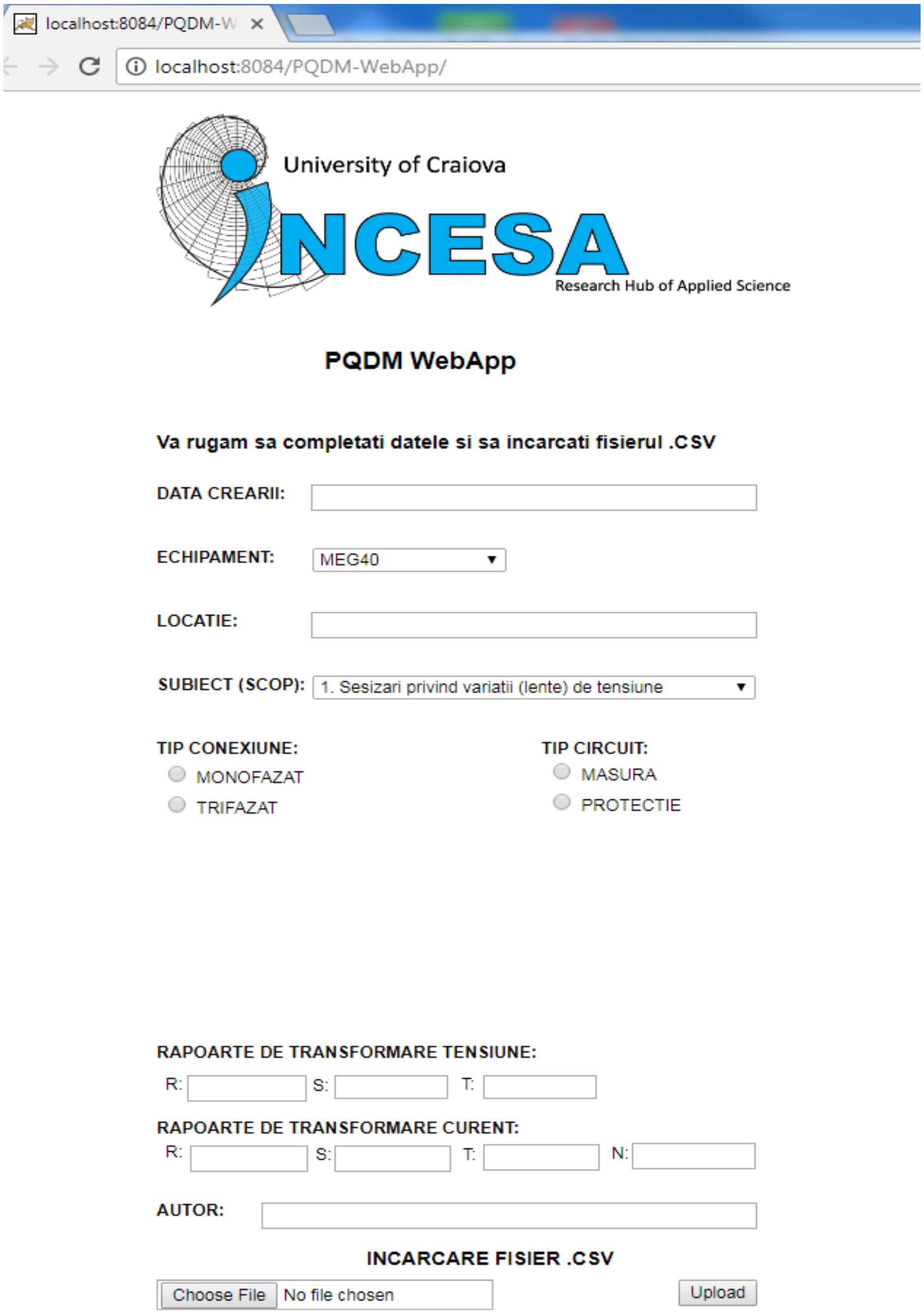
The DBMS used to store the data collected from the power analyser's equipment is MySQL (http://www.oracle.com), which is an open-source relational database management system.

2.2. PQDM-WebApp in running mode. Utilisation

The user accesses the web application PQDM-WebApp by entering the address provided by the network administrator in the browser according to the application server it will run.

For example, we used the server on a local machine: http://localhost:8080/PQDM-WebApp/
The PQDM-WebApp application receives the file for the selected device along with the user metadata's and returns the PQDIF (http://grouper.ieee.org/groups/1159/3/).

The user must introduce the metadata's in PQDM-WebApp interface and select the corresponding file for the selected power equipment as is shown in Figure 3.



PQDM WebApp

Va rugam sa completati datele si sa incarcati fisierul .CSV

DATA CREARII:

ECHIPAMENT: MEG40 ▼

LOCATIE:

SUBIECT (SCOP): 1. Sesizari privind variatii (lente) de tensiune ▼

TIP CONEXIUNE:
○ MONOFAZAT
○ TRIFAZAT

TIP CIRCUIT:
○ MASURA
○ PROTECTIE

RAPOARTE DE TRANSFORMARE TENSIUNE:
R: S: T:

RAPOARTE DE TRANSFORMARE CURENT:
R: S: T: N:

AUTOR:

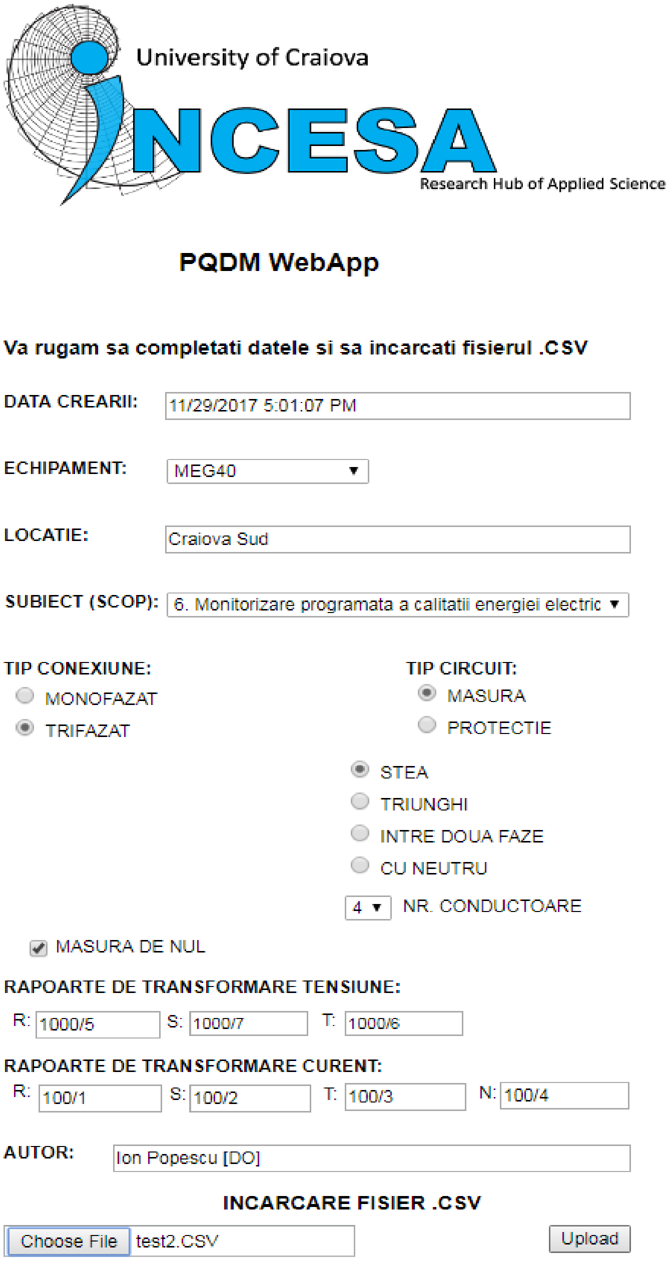
INCARCARE FISIER .CSV

Figure 3. PQDM-WebApp User Interface

Consistent with the structure of the PQDIF file that will be generated through the PQDM-WebApp application, the user will fill in the information in the web interface formats. The details of the completion of the information will be presented in the next section.

2.3. Case Study

An analysis was carried out for a MEG40 device (http://www.e-mega.cz/meg-40) together with the following user input information's: Version Info, File Name, Creation, Subject, Author, Application, Copyright, Trademark, Notes, Language, Owner.



PQDM WebApp

Va rugam sa completati datele si sa incarcati fisierul .CSV

DATA CREARII: 11/29/2017 5:01:07 PM

ECHIPAMENT: MEG40 ▾

LOCATIE: Craiova Sud

SUBIECT (SCOP): 6. Monitorizare programata a calitatii energiei electric ▾

TIP CONEXIUNE:
○ MONOFAZAT
● TRIFAZAT

TIP CIRCUIT:
● MASURA
○ PROTECTIE

● STEA
○ TRIUNGHI
○ INTRE DOUA FAZE
○ CU NEUTRU

4 ▾ NR. CONDUCTOARE

☑ MASURA DE NUL

RAPOARTE DE TRANSFORMARE TENSIUNE:
R: 1000/5 S: 1000/7 T: 1000/6

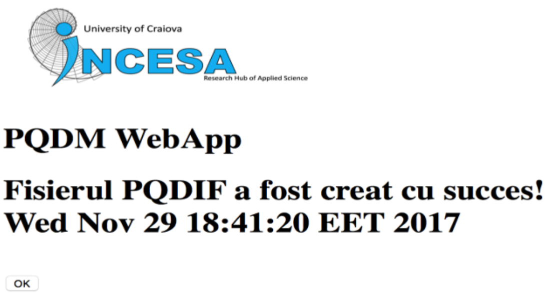
RAPOARTE DE TRANSFORMARE CURENT:
R: 100/1 S: 100/2 T: 100/3 N: 100/4

AUTOR: Ion Popescu [DO]

INCARCARE FISIER .CSV

test2.CSV

Figure 4. PQDM-WebApp utilization



PQDM WebApp

Fisierul PQDIF a fost creat cu succes!
Wed Nov 29 18:41:20 EET 2017

Figure 5. PQDM-WebApp – PQDIF success file creation

After the metadata and the .CSV file for MEG40 are inserted into the web interface, the PQDM-WebApp creates the corresponding PQDIF file, as shown in Figure 5.

3. CONCLUSIONS

In this paper a web-based solution, which manage and convert data regarding the power equipment's and the power quality in the power distribution networks, was presented.

During the research, we were forced to automate the common process in order to minimize the time spent in transfer/copying data. As a future work we try to improve this automation in order to optimize the whole process.

This web-based solution integrates several data sources and analysis tools. It associates the capabilities of the commercial software platforms with those of an in-house designed software named PQDM-WebApp.

This application allowing us to extend the capabilities of the commercial software packages and to develop a complex data management of the power grid networks based on its typical operation regimes, as well as on the real historical evolution of its electric values. The management and the conversion modules of web application enable the system configuring, as well as report generation and data visualization for a heterogeneous large volume of data. The architecture is flexible, highly portable, distributed and platform independent.

ACKNOWLEDGEMENTS

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI – UEFISCDI, project number PN-III-P2-2.1-BG-2016-0269, within PNCDI III".

REFERENCES

[1] IEEE, 2008, IEEE Std 1159.3 Recommended Practice for the Transfer of Power Quality Data.

[2] http://grouper.ieee.org/groups/1159/3/ (Accesed date: March 2018)

[3] D. Sabin, A. R. Dettloff, P. Golden, “Automatic Subtransmission Fault Location System using Power Quality Monitors”, Transmission and Distribution Conference and Exposition (T&D), 2016, DOI: 10.1109/TDC.2016.7519887

[4] L.G. Manescu, D. Rusinaru, C. Popirlan, G. Stoian, M. Ciontu, G.C. Buzatu, M. Alba, A. Cojoaca, “Complex Software System for Data Management and Analysis of Power Distribution Grids”, MCSI 2017, Corfu, Greece, mcsi-142.

[5] D. Rusinaru, L.G. Manescu, M. Ciontu, P.M. Mircea, G.C. Buzatu, G. Stoian, C. Popirlan, T. Vilceanu, A. Negoita, M. Alba, “Powering knowledge transfer at INCESA-BRIDGE projects”, Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP), 2017 International Conference on, 25-27 May 2017, Brasov, Romania, http://ieeexplore.ieee.org/abstract/document/7974969/, pp. 190-195, DOI: 10.1109/ OPTIM.2017.7974969, Publisher IEEE

[6] P. Martí, M. Velasco, J. Torres-Martínez, J. Miret and M. Castilla, "Reactive power control for loss minimization in low-voltage distributed generation systems," 2016 12th IEEE International Conference on Control and Automation (ICCA), Kathmandu, 2016, pp. 371-376.

[7] L. G. Manescu and D. Ruşinaru, "Loss based performance index for the reactive power control," Optimization of Electrical and Electronic Equipment (OPTIM), 2012 13th International Conference on, Brasov, 2012, pp. 307-312.

[8] L. Bam, W. Jewell, “Review: power system analysis software tools”, Power Engineering Society General Meet., 2005. IEEE, vol 1, pp. 139–44.

[9] www.pqview.com (Accesed date: May 2018)

[10] PQView User Manual and Quick Start Guide, Electrotek Concepts, Inc., 2013.

[11] Rusinaru, D., Manescu, L.G., Merfu, M., Postolache, P., “Power quality general levels in distribution networks”, 16th International Conference on Harmonics and Quality of Power (ICHQP), 2014 IEEE, Bucharest, Romania, 25-28 May 2014, pp. 58 – 62.

[12] C.I. Popirlan, “A solution based on intelligent software agents to improve the data searching in the contact centers”, 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, IEEE ITAIC 2011, Chongqing, China, ISBN(Print) 978-1-4244-8623-6, Vol. 2, pp. 1-5.

[13] G. Stoian, C.I. Popirlan, “Using Mobile Agents for Handoff Tuning in Mobile WiMAX Networks”, Annals of the University of Craiova - Mathematics and Computer Science Series, ISSN 1223-6934, Vol. 37, No. 3, 2010, pp. 22-28

[14] https://www.mathworks.com/products/matlab.html

[15] https://www.oracle.com/java/technologies/index.html

[16] http://tomcat.apache.org

[17] http://www.oracle.com/technetwork/middleware/glassfish/overview/index.html

[18] https://www.eclipse.org/jetty/

[19] http://www.e-mega.cz/meg-40

AUTHORS

INCESA (Research Hub for Applied Sciences) is intended to be an inspiring co-working space for driving effective innovation for the benefit of ever wider communities. www.incesa.ro

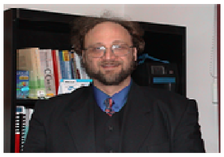
Claudiu Popirlan and Gabriel Stoian are members at Computer Science Department, University of Craiova and researchers at Laboratory of Formal Intelligence Integration in Analysis, Simulation, Development, Testing and Certification of Computation Infrastructures, INCESA (http://incesa.ro/#/computer_science/computation_infrastructure).

Leonardo Geo Manescu, Denisa Rusinaru, Marian Ciontu and Gabriel Cosmin Buzatu are members at Faculty of Electrical Engineering, University of Craiova and researchers at Laboratory of Innovative Techniques and Processes in Complex Electromechanical Systems, INCESA (http://incesa.ro/#/electrical_engeneering/smart_grids)

Miron Alba and Adrian Cojoaca are members of the company Oltenia Distribution SA (https://distributieoltenia.ro/en) and have a great experience in developing applications in the field of electrical engineering.

The authors have numerous scientific papers in the fields of computer science and electrical engineering and many participations in international conferences around the world.

Claudiu Popirlan



Gabriel Stoian

Leonardo Geo Manescu

Marian Ciontu

Miron Alba

Adrian Cojoaca

INTENTIONAL BLANK

SOFTWARE FOR THE APPLICATION OF MULTIFRACTAL ANALYSIS IN THE CLASSIFICATION OF METASTATIC BONE TUMORS

Aleksandar Stančić[1], Marija Prokopijević , Jelena Vasiljević[2,1],
Željko Stojković[2], Goran Dimić[2], Jelena Sopta[4], Dragan Kreculj[5] and
Dhinaharan Nagamalai[6]

[1]School of Computing, Union University, Belgrade, Serbia
[2]Institute Mihajlo Pupin, University of Belgrade
[4]Medical Faculty, Institute of Pathology, University of Belgrade
[5]Faculty of Mechanical Engineering, Belgrade, Serbia
[5]Wireill, Australia

ABSTRACT

Multifractal analysis emerged as an accurate tool for cancer classification, although the process needs complete automation in order to be applied in medical diagnostics. After the multifractal analysis has been performed, the results must be processed and properly classified. This paper presents the automation of the diagnostic process. The goal is to successfully apply multifractal analysis as an auxiliary diagnostic tool.

KEYWORDS

Diagnostic agent, metastatic tumor, multifractal analysis

1. INTRODUCTION

The World Health Organization warns that in the next 50 years the number of cancer patients will increase by 50% compared to 2000. That is, the number of cancer patients will be 15 million people annually. Due to the constant increase in the number of patients, it is necessary to introduce new diagnostic tools for better and faster diagnosis. In this paper an auxiliary diagnostic agent for the classification of metastatic bone tumors by primary carcinoma was introduced. Namely, it has been established that there are statistically significant differences between microscopic medical images with a view to their classification, in terms of determining primary carcinoma in cases of intraoseal metastatic carcinoma [1]. In order to apply the results in a diagnostic, it is necessary to automate the process of obtaining the results from the microscopic image to the classified sample.

2. MULTIFRACTAL ANALYSIS

Multifractal analysis is a continuation of fractal analysis, introduced by mathematician Benotit Mandelbrot, in the second half of the twentieth century. Fractals related to natural forms, such as mountains, shores and river basins, the structure of plants, blood vessels, lungs, which can not be described by Euclidean geometry.

David C. Wyld et al. (Eds) : NeCoM, SEAS, SP, CMCA - 2018
pp. 97–104, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.80908

Natural objects and phenomena do not exhibit fractal properties strictly, even when they are self-similar, but can have statistical self-similarity. For example, the structure of the seashore, the appearance of a relief or cloud, the structure of some biological systems or signals, exhibit self-similar properties, but in different scales the shape is not exactly the same [1]. While fractal analysis describes forms characterized by strict mathematical properties, the multifractal analysis defines fractal properties in natural objects and phenomena. Their application in medical diagnostics is presented in paper [2].

The cancer cell is one of the natural forms that can be expressed through fractals. It is characterized by chaotic, poorly regulated cell growth [1], which is not a feature of healthy organisms. A healthy cell defines a form that helps in their functioning, while the appearance of cancer cells is usually abnormal. Abnormalities are expressed in a size that is either lower or higher than a healthy cell, because cancer cells do not have a particular function. Irregular growth also occurs in the nucleus and cytoplasm of the cell. Namely, the core of the malignant cells is higher than in healthy cells, while the cytoplasm is scarce and its color is intensively colored or very faint [3]. For the multifractal analysis of the obtained digital medical images in [1], as well as obtaining multifractal analysis parameters, the following programs were used: ImageJ, for image analysis [4], and ImageJ Plugin, FracLac [5], multifractall analysis program.

Digital images of three groups of metastases of the intraoseal cancer were observed:

- metastatic renal cell cancer, shown in Figure 1,
- metastatic breast cancer, shown in Figure 2,
- metastatic lung cancer, shown in Figure 3.

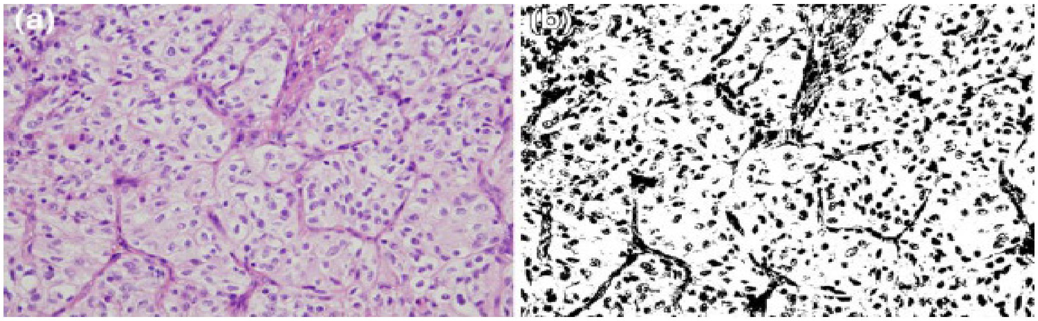


Figure 1. Metastatic renal cell carcinoma: (a) Microscopic imaging; (b) Binary form, obtained using the FracLac program

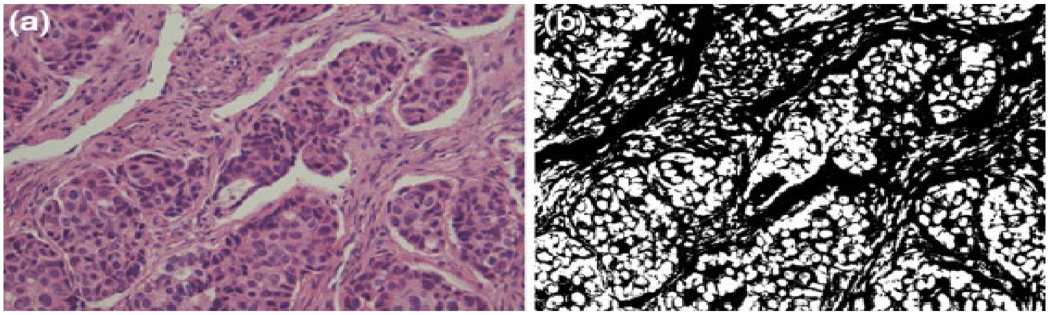


Figure 2. Metastatic breast cancer: (a) Microscopic; (b) Binary form, obtained using the FracLac program

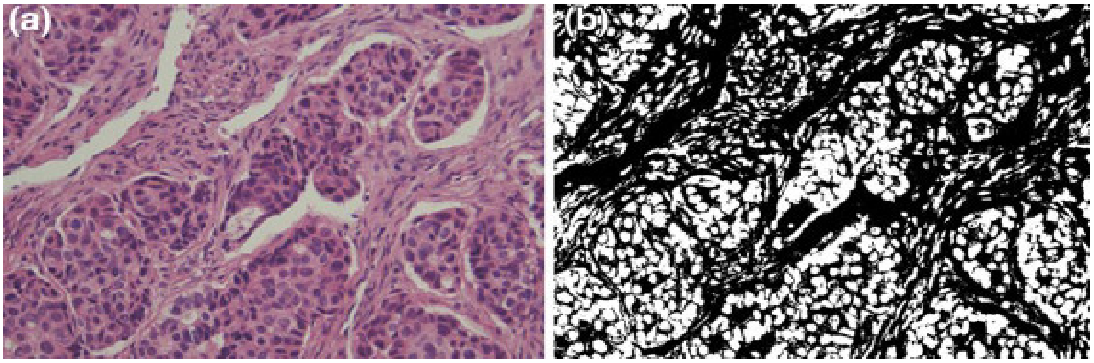


Figure 3. Metastatic lung carcinoma: (a) Microscopic image; (b) Binary form, obtained using the FracLac program

After applying the FracLac software to the metastasis images, the following multifunctional parameters for data classification are obtained:

- D_{max} - fractal dimension
- Q - is the exponent of the fractal dimension
- α_{min} - the minimum value of the Hõlder exponent in the results
- $f(\alpha)_{min}$ - the value of the continual function for α_{min}
- α_{max} - the maximum value of the Hõlder exponent in the results
- $f(\alpha)_{max}$ - the value of the continuous function for α_{min}.

3. AUTOMATIZATION

The idea of this project is to create a platform that will enable an easy and simple work environment for the end user, as well as a basis for further research and improvement of the described techniques. The end user will receive a unique interface through which he can even get the results in just a few clicks even without detailed knowledge of multifractal analysis and neural networks (or even software packages that served as the basis, ImageJ and FracLac).

The project is called "DECIM" according to the techniques underlying its processing. The great obstacle in the practical application of these techniques is the complexity and heterogeneity of the software that needs to be used to get the result. The classification currently requires more complex steps in several software packages, with manual inter-steps, such as copying and converting files. The goal of DECIM is to be a unique solution, robust software that takes on the complexity of the process and allows the user to arrive literally from within the next ten seconds from the microscopic image to the classification result, with minimal interaction.

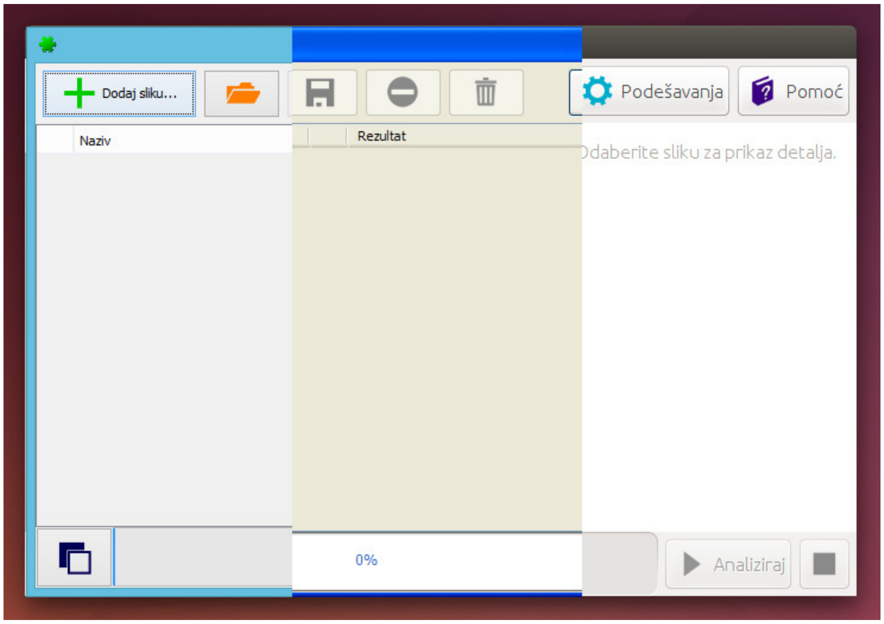
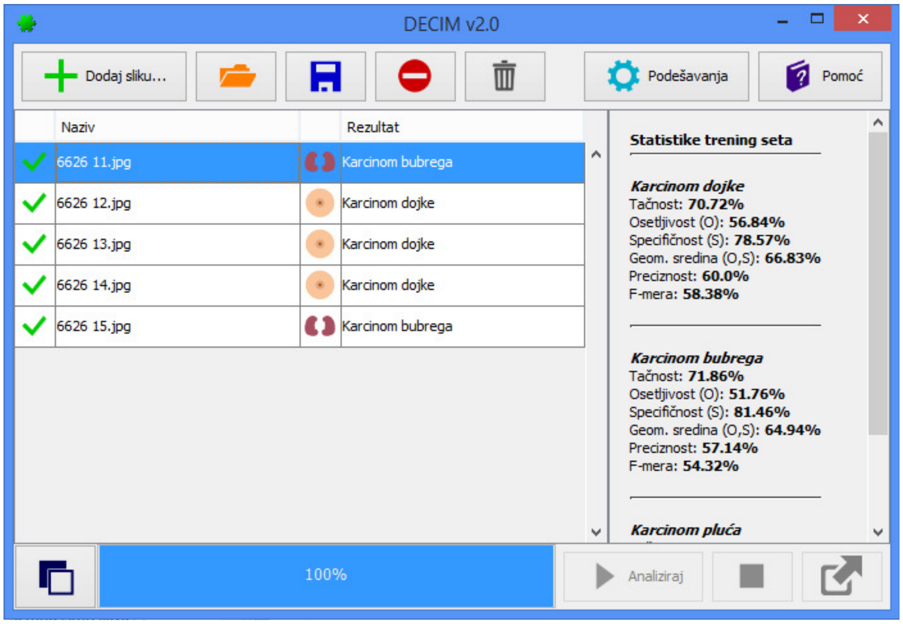


Figure 4. Start window of DECIM program on Windows 8, XP and Ubuntu 14.04

In the initial window (Figure 4), microscopic images can be added in standard digital formats (BMP, JPEG, PNG), either through the normal "Open" dialog, either by dragging images onto the DECIM window. During the typical work, this is all that is needed for preparation, only it is left to start the analysis using the "Analyze" button and the results will be displayed in the window as soon as they are finished (Figure 5). By selecting individual images, more detailed information about each classification can be found.



	Naziv		Rezultat
✓	6626 11.jpg		Karcinom bubrega
✓	6626 12.jpg		Karcinom dojke
✓	6626 13.jpg		Karcinom dojke
✓	6626 14.jpg		Karcinom dojke
✓	6626 15.jpg		Karcinom bubrega

Statistike trening seta

Karcinom dojke
Tačnost: **70.72%**
Osetljivost (O): **56.84%**
Specifičnost (S): **78.57%**
Geom. sredina (O,S): **66.83%**
Preciznost: **60.0%**
F-mera: **58.38%**

Karcinom bubrega
Tačnost: **71.86%**
Osetljivost (O): **51.76%**
Specifičnost (S): **81.46%**
Geom. sredina (O,S): **64.94%**
Preciznost: **57.14%**
F-mera: **54.32%**

Karcinom pluća

Figure 5. Classification of 5 images completed, with details in the right frame

With the ultimate goal of simplicity in mind, special attention is paid to the installation process of this program, which is compiled to automatically check and adjust the environment needed for the operation of DECIM, such as the presence of Java JRE environment (which can come and packaged with the program), the presence of Octave software [6] (it can also come bundled, if not already present) and the installation of the VC 2010 runtime component required for Octave on Windows platforms. In the program, many checks of errors and auto-detection were installed, so as to minimize the user's technical details. The whole project is designed so that for most average users, without special technical knowledge and without the need for manual interventions, this software "simply works".

4. PLATFORM

The software is made in Java programming language, which allows it to portability and work on multiple platforms (Figure 4). It is based on free and no cost software (Octave, ImageJ, FracLac), and can also run on free Java implementation (OpenJDK), on a free operating system (Ubuntu and other GNU/Linux distributions), which means that analysis through it without any additional costs for the software is possible. Of course, common commercial platforms, such as Oracle Java JRE and all modern versions of Windows and MacOS operating systems are supported. Apart from the simplicity of using the initial version of this program, the ambition of this project is that it is also a platform for further improvements in the process of analysis and classification. Due to the nature of this work, it is expected that the analysis will become better over time, or that specific variants will be developed for some specific cases.

One of the modules in the initial version is an improved method of converting color microscopic photos into a binary (black and white) form in relation to the method that ImageJ uses (Figure 6). Conversion to binary form is necessary, as the multifractal analysis method works with binary images. When converting a color image into a binary image, the key is to select the threshold of the light, that we consider the boundary between black and white. ImageJ calculates the global threshold by constructing histogram and selecting threshold that converts roughly the same image surface into black and white (Figure 7).

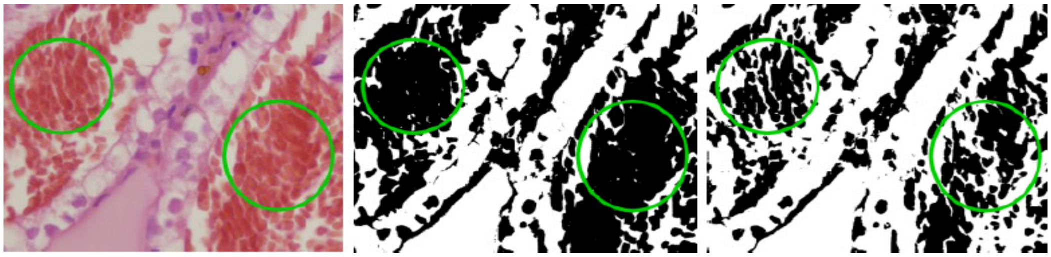


Figure 6. Different image processing techniques for fractal analysis: (a) Original; (b) The technique used by ImageJ; (c) A special technique in DECIM that emphasizes local contrast and thus reveals more detail

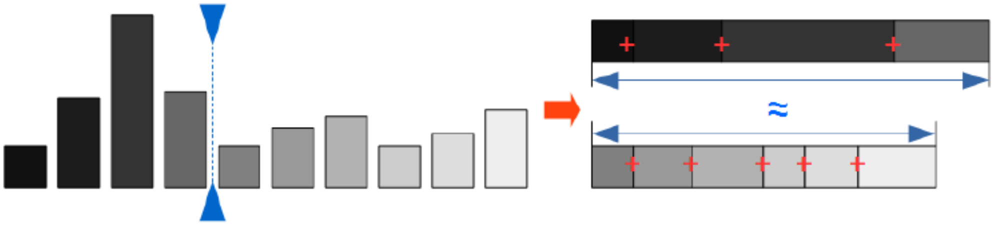


Figure 7. The principle of selecting the threshold for binarization as implements ImageJ

This will not always give the most accurate image repression, especially if some parts are brighter or darker than the global ones. More details in a binary image can be obtained by calculating a local threshold based on the environment, a particular radius, each pixel, instead of using one global value for the whole picture (Figure 8). Such a binarization is achieved by constructing a blurred version of the image (Gaussian blur) in which then each pixel will actually be the weight sum of its environment. Then, calculating the delta image, each pixel resolution will describe the difference between the original image pixels and its immediate surroundings, promoting local contrast.

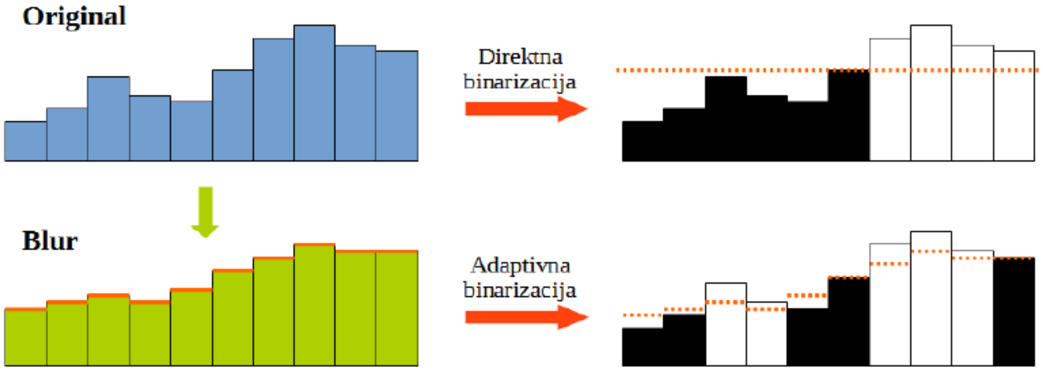


Figure 8. Principle of adaptive binarization work

Although the program is configured to "easily run" immediately after installation and can be used immediately for classification, it also allows modularity and extensibility. Whether it is new training data for the neural network, or new software components for preprocessing and image analysis, DECIM will allow easy use of new components through an already familiar interface. Again, this should be easy for users and, accordingly, whenever possible, the use of new components will be reduced to inserting one, easy to distribute, files into the program, similar to how the same sample images are used in the program. The idea is that DECIM will be a platform and it will present a simple interface that users will be familiar with and that will not change significantly, regardless of any new advances in fractal analysis or neural network classification, or some completely new techniques.

5. Modularity

DECIM realizes its modularity through the plug-in system, some of which are already installed in the initial version of the program, while others can easily be added by users after installation. In accordance with the purpose and mode of the program, a system has been developed that supports five module types, namely: image loader, image preprocessor (image processing), image analyzer, classificator and data exporter (export of classification results). The system extends through special "plug-in" files that can be automatically retrieved from a specific folder, if the user places it there, or during work, by dragging a new file into the program window, similar to images and sheets. A user can have multiple modules of the same type, choosing which one to use in the settings box.

The connection between the modules makes a special "Wiring" object (analogue to a wiring, a serial connection of the module) which at each moment has one ordered five modules (one of each type) that make up the image processing chain. Wiring also allows intercommunication between the modules through the provided interfaces, in order to allow the modules to check their environment and requirements (modules can depend on other modules), and then to call the desired parameters to other modules. This is important, because for the correct results the complete chain must be adjusted exactly as it was set up, when the neural network is created, otherwise the correct correlation between the analysis parameters and the classification results is lost.

Modules can be very easily created in the form of new Java projects that include the attached JAR library and implement one of the predefined interfaces (module type). This project can then be executed as a new JAR file, put into the plug-in folder and will be automatically scanned without any additional steps. The role and functionality of the modules are as follows:

A. Image Loader

The first module in the chain is the one that reads the microscopic photo from the file. Along with the program, a simple module of this type is included, which uses built-in image capture capabilities in Java implementations (JPEG, BMP, GIF, PNG and WBMP are guaranteed to be present). However, due to the application in medicine, it should also be possible for further extensions to read somewhat less popular, specialized image formats.

B. Image Preprocessor

The second phase is a pre-image processing, intended for use with multifractal analysis or some other method of analysis requiring conversion of colors or some other kind of pre-processing. There are two modules for binarization, described in the previous chapter, as well as a special "pass-through" module, which only transmits the image, in the event that no special processing is required (without disturbing the processing chain).

C. Image analyzer

The first of the two main modules is the image analysis module, in the form obtained after the pre-processing phase. Currently, this module is implemented the fractal analysis technique described above. The module is implemented as a compatibility shell for the FracLac plug-in to the ImageJ software, allowing it to work within the DECIM module. The output from this module can be in the form of a file, a sequence of whole or real numbers, or a block of binary data, in order to support possible future analysis methods, that have a different concept and format of output data.

D. Classificator

The second of the two most important modules is the one that, on the basis of the data obtained from the analysis, makes the final classification and gives an answer for each individual image. Here, the described neural network is introduced, which takes the parameters of multifractal analysis for inputs, and gives one of the three organs at the output, which considers to have the best correlation. This module is also responsible for determining the number of classes and their names - the rest of the system respects the information obtained from here. So with the new classifier, it is possible to classify with a different set of organs, and considerably more than three organs, or classify some completely different kind of objects and properties.

E. Result exporter

Although the results are visible within the program window, the need for export and further processing of results is envisaged, enabling complete DECIM to be part of a larger processing chain. A module of this type can save files (HTML report, or Excel table, etc.) or to display an arbitrary GUI window, that implements the module itself, having all the results and original images available.

6. Conclusions

Fractal analysis and classification through a neural network are interesting new techniques with a lot of potential and they are useful diagnostic tools. The purpose of this project is to bridge the typical distance between scientific research papers and new experimental methods/tools that are really applicable in practice. The practice requires robust, stable and reliable tools, and this is precisely what this project seeks to be - a tool that is practical, which will bring with it the convenience of the described classification techniques, as well as the potential for simple future extensions and improvements.

Acknowledgements

This work was supported by the Ministry of Education and Science, Republic of Serbia, Science and Technological Development grant III 43002.

References

[1] J. Vasiljevic, (2012) "Application of multifaceted analysis of microscopic images in the classification of intraoseal metastatic carcinoma", Doctoral dissertation, Faculty of Electrical Engineering, University of Belgrade, Belgrade, Serbia.

[2] Lopes, Renaud, and Nacim Betrouni (2009) "Fractal and multifractal analysis: a review." Medical image analysis 13.4, pp. 634-649.

[3] A. I. Baba, C. Câtoi. (2007) “Comparative oncology”, The Publishing House of the Romanian Academy, First Section: Unit 3.

[4] W. Rasband, ImageJ, Available on: http://rsb.info.nih.gov/ij.

[5] A. Karperien, FracLac for ImageJ, Available on: http://rsb.info.nih.gov/ij/plugins/fraclac/fraclac.html

[6] James B. Rawlings, John G. Ekerdt, GNU Octave, Avalilable on:
https://www.gnu.org/software/octave.

AUTHOR INDEX