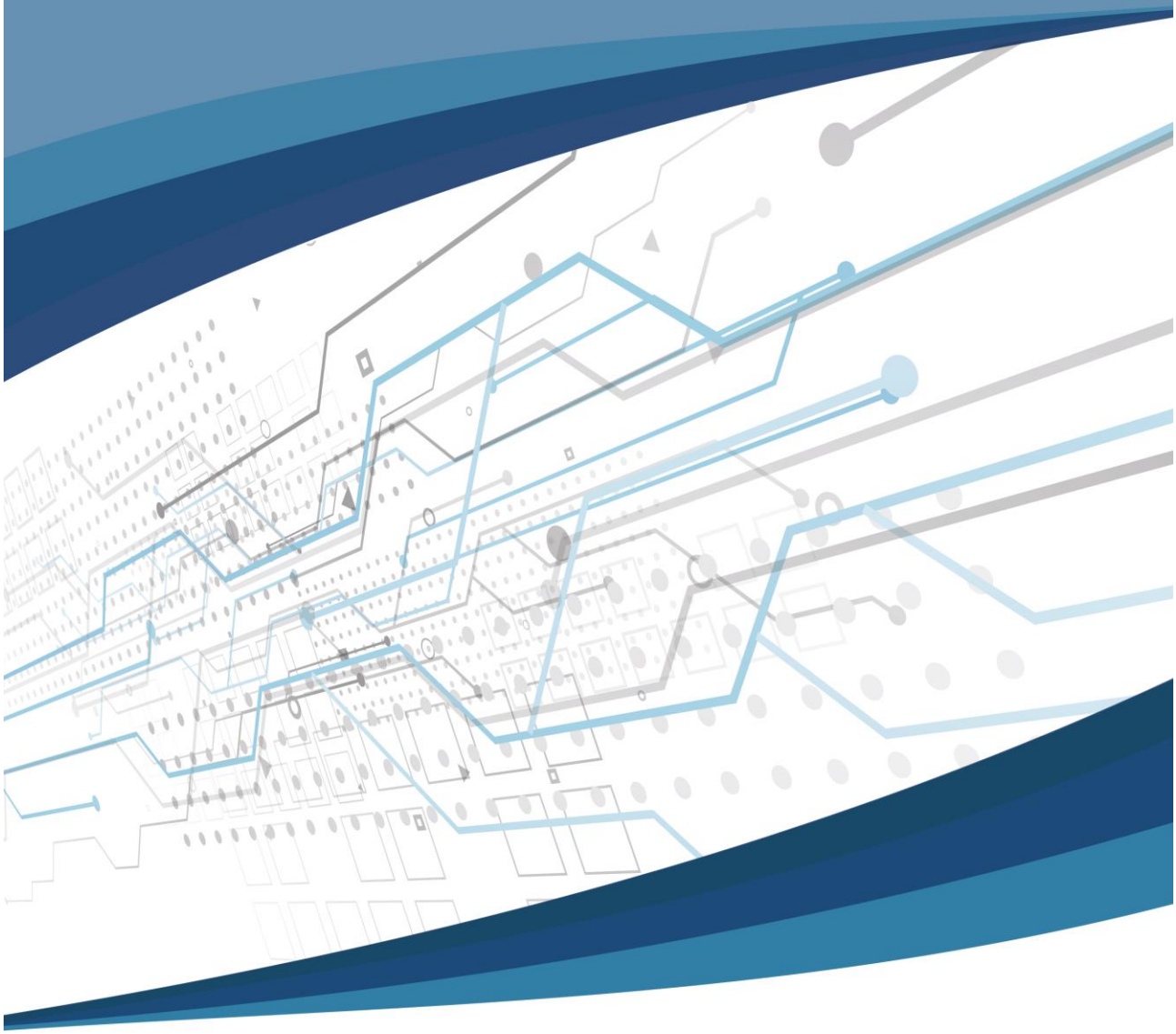
ICI-CS2022

INTERNATIONAL CONFERENCE ON INFORMATICS AND COMPUTER SCIENCE



ICI Computer Science 2022

by Informatics Institute in Gazi University

ONLINE CONFERENCE

2nd International Conference on Informatics and Computer Science (ICI-CS2022)

December 1-2, 2022

Ankara, Turkey

ISBN: 978-975-507-318-7

Preface

International Conference On Informatics and Computer Science (ICI-CS2021), held in December 2021 with participants from Turkey and abroad and hosted by Gazi University Institute of Informatics, was a great success in terms of providing an interdisciplinary platform for innovative approaches to complex global agendas. The success of the first year has led us to organize the second year of this comprehensive conference. The 2022 International Conference On Informatics and Computer Science (ICI-CS2022) aimed to continue the interdisciplinary discussions and pave the way for innovative solutions to global agendas.

ICI-CS2022 was held in December 2022, once again hosted by the Gazi University Institute of Informatics. We have invited researchers, academics, students, and industry practitioners from informatics, computer science, and other interdisciplinary fields to contribute to the conference. This year, we aimed to focus on current and raising technologies in the field. In addition to the presentations of original research articles, we have a series of invited talks from distinguished scholars and practitioners and created an environment for discussions and collaborations. This way, ICI-CS2022 also enabled participants to establish lasting connections with colleagues in their respective fields.

We would like to sincerely thank the conference program chairs, committee members, and all the reviewers for their great professionalism and efforts in reviewing the submitted papers. Additionally, we thank all the participants and sponsors for their valuable contributions and support to ICI-CS2022. We also look forward to announcing the third year of the conference, ICI-CS2023, and hope to see you all there.

Conference Chair

Prof. Dr. Aslıhan TÜFEKCİ - Gazi University, Informatics Institute

Conference Committees

Honorary President

Prof. Dr. Musa YILDIZ - Gazi University Rector

Chairman of the Management Committee

Prof. Dr. Aslıhan TÜFEKCİ - Gazi University

Member of the Management Committee

Prof. Dr. Mahir DURSUN - Gazi University

Prof. Dr. Nursal ARICI - Gazi University

Prof. Dr. Mustafa ALKAN - Gazi University

Prof. Dr. M. Ali AKCAYOL - Gazi University

Prof. Dr. Hasan Şakir BİLGE - Gazi University

Prof. Dr. Serdar KULA - Gazi University

Assoc. Prof. Dr. Recep BENZER - Gazi University

Assoc. Prof. Dr. Sinan TOKLU - Gazi University

Asst. Prof. Öner BARUT - Gazi University

Res. Asst. M. Ali KOŞAN - Gazi University

Res. Asst. Tuana İRKEY - Gazi University

Res. Asst. Merve CELEBİ - Gazi University

Res. Asst. Dr. Mehmet SEVRİ - Recep Tayyip Erdogan University

Dr. Sedat HAKYEMEZ - Gazi University

Dr. Nadircan DAĞLI - Gazi University

Conference Steering Committee

Prof. Dr. Kemal LEBLEBİCİOĞLU - ODTÜ

Prof. Dr. Ferat ŞAHİN - Rochester Institute of Technology

Member of the Technical Committee

Prof. Dr. Hacer KARACAN - Gazi University

Assoc. Prof. Mehmet ATAK - Gazi University

Assoc. Prof. İbrahim Alper DOĞRU - Gazi University

Res. Ass. Uğurhan KUTBAY - Gazi University

Prof. Dr. Fırat HARDALAÇ - Gazi University

Prof. Dr. Ayhan ERDEM - Gazi University

Assoc. Prof. Oktay YILDIZ - Gazi University

Assoc. Prof Erkan BOSTANCI - Ankara University

Asst. Prof. Ayetül GELEN - Bursa Technical University

Asst. Prof. Nurettin Gökhan ADAR - Bursa Technical University

Asst. Prof. Serdar ENGİNOĞLU - Çanakkale Onsekiz Mart University

Asst. Prof. İlker Özçelik, Eskişehir Osmangazi University

Assoc. Prof. Fatih Kılıç, Adana Alparslan Türkeş University

Assoc. Prof. Zerrin IŞIK - Dokuz Eylül University

Assoc. Prof. Abdullah Talha KABAKUŞ - Düzce University

Assoc. Prof Orhan DAĞDEVİREN - Ege University

Assoc. Prof Bilal BABAYİĞİT - Erciyes University

Prof. Dr. Ahmet YAZICI - Eskişehir Osmangazi University

Prof. Dr. Nezih GEREK - Eskişehir Technical University

Assoc. Prof. Ahmet Mete VURAL - Gaziantep University

Asst. Prof. İpek ABASIKELEŞ-TURGUT - Iskenderun Technical University

Asst. Prof. Uğur ERKAN - Karamanoglu Mehmetbey University

Asst. Prof. Recep Sinan ARSLAN - Kayseri University

Assoc. Prof Kazım YILDIZ - Marmara University

Asst. Prof. Kemal UÇAK - Muğla Sıtki Koçman University

Asst. Prof. Anıl UTKU - Munzur University

Assoc. Prof. Utku KÖSE - Süleyman Demirel University

Assoc. Prof. Gökçe Nur YILMAZ - TED University

Assoc. Prof Korhan KARABULUT - Yasar University

2nd International Conference on Informatics and Computer Science (ICI-CS2022) Table of Contents

Preface……...…… iii

Conference Committees……. iv

Building Goal-Oriented Processes and Team Structure in Agile and Hybrid Security Software Projects…….……. 1
Süleyman Muhammed ARIKAN

Comparison of the Sizes of Assembly Codes Generated by the GCC Compiler for Different CPU Architectures…… 8
Esra ÇELİK, Deniz DAL

Text Preprocessing Steps for Patent Documents: A Reconsideration of Standardized Approaches……... 14
Furkan İNCEDAYI, Hakan ÇERÇİOĞLU

An IaaS Instance Selection Decision Model for Cloud Computing……..... 15
Onur KOŞAR, Üzeyir FİDAN, Mehmet ATAK

Systematic Literature Review: Improving IT Helpdesk Services……..... 21
Melek BASMACI TOP, Aslıhan TÜFEKCİ

Digital Design and FPGA Implementation That Converts an Integer to A Floating Point Number…… ……..... 27
Esin MUTLU, Serkan DERELİ, Asiye Rümeysa ÜNSAL

Color Based Image Segmentation Methods and Applications: A Review of the Last Decade. 32
Tesnim GÜLŞEN, Serkan DERELİ

What Word Embeddings Tell Us About Turkish Coreference Resolution…….…… 39
Seniz DEMİR

A System for Optimization of Workloads of Test Units and Monitoring of Their Processes… 46
Pınar BÜLBÜL UĞUREL, Hasan Şakir BİLGE

National Qualifications Frameworks and Qualification Databases: Turkish Example……... 52
Ekrem YAZICI, Aslıhan TÜFEKCİ

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

Building Goal-Oriented Processes and Team Structure in Agile and Hybrid Security Software Projects

Süleyman Muhammed ARIKAN

Cyber Security Institute, TÜBİTAK BİLGEM, Ankara, Turkey
suleyman.arikan@tubitak.gov.tr

***Abstract*—** The development of today's cyber security technologies has become more difficult in direct proportion to the growth of needs, the complexity of technical solutions, and so forth. For this reason, in order to be successful in cyber security technology development projects, goal-oriented processes should be operated as efficiently as possible and resources (labour, know-how, etc.) should be better managed. In a project using a hybrid project management with the Scrum Framework, a software process improvement (SPI) has been performed to ensure that the mentioned issues are addressed in the project management. The SPI has defined many activities, from identifying milestones to integrating milestone management process into software development life cycle tools. In addition, a model has been proposed that can scale the teams so that the resources can be used efficiently for the determined milestones. In this study, the information obtained in the research phase, the mature processes as a result of the SPI, and the gains acquired by the implementation of the improved processes have been provided. With the SPI, it has been seen that software engineering techniques can be applied more efficiently, performances of project personnel (including managers, leads, and subcontractors) are capable of being observed clearly, project progress is possible to be monitored more realistically, and detailed reports can be easily created for each goal. At the same time, it has been observed that project personnel can be more proactive and effective in activities such as planning, tracking, reporting, collaboration, and even completion date prediction of a milestone. Finally, it can be mentioned that the proposed scaling model and the SPI can even be applied to all software projects in different disciplines.

***Keywords*—** milestone management, team scaling, scrum framework, software process improvement, hybrid project management

1. INTRODUCTION

Today's cyber security technologies have become more complex in direct proportion to the growth of needs. For this reason, in order to be successful in projects that develop technology to cyber security ecosystem, goal-oriented processes should be operated as efficiently as possible. The high efficiency of goal-directed and result-oriented planning, tracking, and reporting is indisputable. With the activities performed in this way, it can be easily (relatively) ensured that projects are completed on time, on budget and high quality. However, at the same time, all kinds of resources (labour, know-how, etc.) must be used efficiently. Team management is of great importance in order to better plan and use the labour (and time) of team members, to manage the transfer of know-hows efficiently, and even to increase and maintain their motivation. Since there is so much expertise in the field of cyber security, scaling is needed to achieve the goals, to better manage the team, and to provide homogeneity. Team members are expected to be homogeneous within agile methodologies, especially in the scrum [1]. But teams are often set up far from homogeneity, as large-scale projects (especially cyber security projects) often require different expertise. On the other hand, traditional scrum are not capable of managing large number of team members [2] as many agile methods are designed for single small teams, members of which are probably enough to fit in a physical room [3].

In a project that hybrid project management was adopted, a process improvement was needed in order to be able to make goal-oriented planning, monitoring, and reporting. With this experience, it is aimed to make the personnel more efficient and proactive in the mentioned activities and to make performance evaluations more objective. As there were many different specialties in the project, a structure that enables team scaling according to the needs

and current goals is aimed in order to position cooperating personnel. Besides of a traditional methodology, the scrum has been used in the project. For this reason, a model that will provide scaling within the scrum is presented. More effective use of milestone management is aimed to provide and maintain goal-oriented processes.

With this study, the followings are presented to the literature: (I) the current status of milestone management in all project management methodologies (ie., traditional, agile, hybrid), (II) a review of existing scrum scaling methodologies, (III) new processes introduced by process improvement, (IV) the achievements obtained from these processes. The next part of the study, Section II, focuses on milestone management. In Section III, how a scalable scrum team structure can be managed is explained. In Section IV, the gains of the process improvement experience and the activities operated additionally are stated. Finally, the conclusion is given in Section V.

2. PROVIDING GOAL-DRIVEN PLANNING, MONITORING AND REPORTING

2.1. State of the Process

Separating the processes in a project and dividing the separated processes into smaller tasks is a traditional and successful method used to make the project more planable, manageable, trackable and reportable. In order to meet this need, there are some processes in project management, especially for large and complex projects. One of them is to create WBS, which stands for Work Breakdown Structure. WBS is a hierarchical representation of all work elements [4]. The Project Management Institute's (PMI) A Guide to the Project Management Body of Knowledge (PMBOK) defines the WBS as a decomposition of the work to be carried out by the team to accomplish the objectives [5]. While the most common types of WBS are deliverable-based WBS and phase-based WBS, as the focus is usually on the project outputs or the project schedule, there are also different types of WBS, such as resource-based WBS, risk-based WBS, and responsibility-based WBS [6]. No matter which type(s) are preferred, the development of the WBS is often carried out as the first step in planning, after acquiring the project requirements [7]. To create a WBS, requirement list and project scope are needed [5, 8]. To obtain information about the scope of the project, Scope Management Plan can be used [5], or Project Charter (also known as Project Definition Plan) can be utilized [8]. However, WBS does not include activities, tasks, milestones, etc. All this information is presented in the associated WBS Dictionary which is a document providing detailed deliverables, work descriptions, assumptions, constraints, scheduling information, milestone list, and so on [5].

In the light of all this information, so that milestones can be defined, it is expected that the scope and requirements of the project have been defined and the WBS has been created. Also, defined milestones are supposed to be documented and tracked under the WBS Dictionary. Consequently, milestones of existing elements and their progression cannot be tracked and reported via WBS. To perform these activities, the WBS and the associated WBS Dictionary may be used together. All the information needed on the road to creating a milestone can be easily obtained in traditional project management methodologies. However, to adjust to changing requirements, many project managers have adopted hybrid project management that integrates the agile approach with traditional approaches [9]. In the hybrid project management, the aim is to benefit from the strengths of each approach [10]. The agile approach can be followed during the implementation phase, while traditional methods may be used to plan the resources, to monitor and report the progress, and to operate the information flow with the stakeholders. In such a case, the WBS can still be maintained with existing processes and a change in the milestone management may not be needed.

Although in agile projects, the scope is not clearly known from the beginning and it is supported by the backlog [11], the WBS can still be quite useful. In agile projects, a WBS (known as Agile WBS) provides the basis for breaking down work into epics, stories, and tasks [12]. After acquiring the decomposed entities, the milestone list can be created. In an agile project, a milestone can be defined as a collection of these entities, especially (and naturally) epics [13-15]. From time to time, it is also stated that milestones should consist of stories [14, 15]. Stories specify features. Tasks, on the other hand, may be individual (usually) works performed to fulfill the relevant feature. Therefore, in projects that use the terms milestone and release interchangeably and/or use them for the same purpose, tasks are not expected to exist within a milestone. Similarly, some tools do not allow inclusion of tasks, assuming that tasks are non-releasable [15]. However, it should be mentioned that literature does not clearly stated that how to map the milestones in an agile project. In our study, this situation has been clarified by making a conceptual analysis on the terms release and milestone. A release is a set of deliverable software features, while a milestone is a specific goal, event, or marker. It may be more accurate to have stories, instead of tasks, within a release because stories can correspond to features. On the other hand, in the light of the knowledge that milestones represent a particular point, it has been concluded that tasks can also be included in a milestone.

As can be seen, regardless of whether the project management methodology is traditional, agile, or hybrid, milestone defining process always needs a WBS output. However, the management of information may vary according to the preferred management approach. As it is aimed not to waste more time than necessary with the documentation in agile methodologies, artifacts such as epics, stories, and tasks are usually defined directly on the software development life cycle tools. Thus, all the defined milestones should be stored on tools. However,

milestone lists are usually created, saved, updated, monitored, and reported via various documentation and/or collaboration tools since it is not always easy to find a tool that makes it possible to define and store milestones.

2.2. Proposed Approach

Planning, monitoring, and reporting activities of the project for which software process improvement has been carried out are based on traditional methodologies since hybrid project management has been adopted. For this reason and due to some customer limits, a tool was not used in the processes of defining and recording milestones, and these needs were met through documentation tools. Thus, first, milestones have been planned, defined, and stored via a documentation tool. However, milestones are created directly on a milestone chart to save an additional step. In this context, first of all, a new milestone chart template has been created to be used in the project. The chart is provided to have a structure in the form of a gantt chart. Within the scope of the project, 3 types of milestones have been defined. Milestones expressed with the abbreviation TG are created in line with technical goals. With the abbreviation SE, milestones obtained through the analysis focused on subcontractor expectations/goals have been stated. With the abbreviation PP, milestones revealed by the analysis focused on project processes are indicated. It should be noted that naturally, the types of project-specific milestones can be diversified. Coloring is used in addition to adding filter(s) to the various columns of the chart to facilitate discrimination, readability, and reviewability. Furthermore, possible working times to reach the relevant milestone on time are highlighted with various symbols on the chart. In this way, in addition to the deadline information for the completion of a milestone, start time information (usually not expected from a milestone) has been also obtained. It is clear that this improvement in the planning and definition activities did not and will not offer a big gain compared to the existing processes. The main improvement for the milestone management, which is expected to be maintained through the document(s), has been the integration of the process into the software development life cycle tools. Milestones obtained as a result of analyzes focused on subcontractor expectations/goals and project processes are relatively suitable for management through the document(s). However, integration with task management software for tracking TG-milestones is essential for high efficiency in the process. Performing monitoring and reporting activities in the milestone management process through the task management tool will make it easier and more efficient (in many cases, enable) to identify the current status of the tasks for the associated goal, to determine the percentage of completion and the speed of completion of the work, and to reveal the personnel who have contributed the most on the way to the goals. For this reason, milestones are defined on the currently used task management tool. Although there are plugins that are available for these activities and can be used with the tool (and even the institution already has licenses for some), the builtin features of the tool have been used in these processes. With the template document and defined process, possible working times for the completion of the milestones are also specified, so a similar milestone management process has been achieved with the release planning/management. Therefore, the release management features of the existing tool have been utilized. Using the release management for milestone management allowed developers to reach milestones with ease. Because there is also direct access to the page from the Scrum Board, where tasks are created, assigned and monitored. The releases page and the fields used when defining a milestone are presented on Fig. 1.

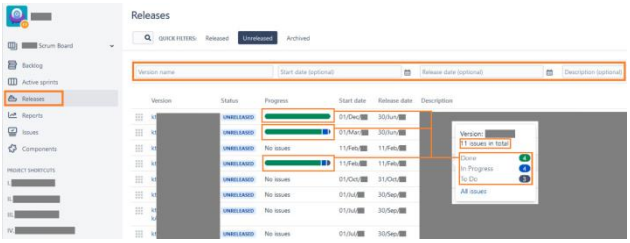


Figure 1. Listing milestones via the releases page

The abbreviation “kt” is used at the beginning of the milestones, as it is also expected to manage versions of a project in this area, and/or versions may already be used in an existing project. All milestones created within the scope of technical targets should be transferred to the system in this way. Depending on the needs, milestones obtained as a result of analyzes focused on subcontractor expectations/goals and project processes can be carried out on only one record. Defined milestones should be associated with each task. Fig. 2 shows the field used for mapping tasks with milestones. Through the field, it is possible to associate more than one milestone with the task. After mapping, associated tasks can be listed under the relevant milestone. It is ensured that the tool features provided for releases are used for milestones. In this context, status of all milestone tasks and the graphs expressing these statuses can be accessed. A screenshot is provided on Fig. 1.

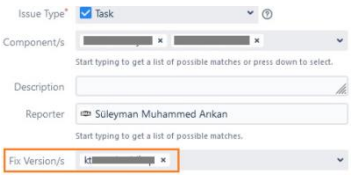


Figure 2. Mapping of tasks with milestones

It has also been observed that the following information can be listed for a milestone: the milestone description, possible start time for the associated tasks to complete the milestone successfully on time, expected deadline for reaching the milestone, number of days remaining, current tasks and status distributions, information about the personnel carrying out the task(s), and developer's branch and its status. Some information (branch, branch status, so on) is provided by the integration with the version control system. A screenshot showing the relevant information is presented in Fig. 3.

Figure 3. Viewing the details of a milestone

In addition to all these, with the task management tool, the progress for the relevant milestone can be displayed in the form of a burndown chart. This statistical information can be easily accessed by the project personnel. During each sprint, it is made visible how much progress has been made for the relevant goal. In order to reach the goal, the average number of story points completed in sprints and the estimated number of remaining sprints are also calculated. In addition, various forecasts for the end date can be obtained via the tool, including optimistic and pessimistic forecast according to progress. An example of a burndown chart created for a milestone is shown in Fig. 4. In this way, progress may be easily reported and tracked on a goal-oriented basis.

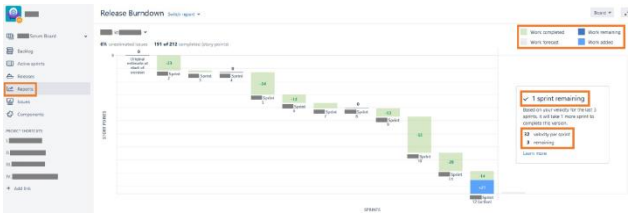


Figure 4. An example of a milestone burndown chart

To summarize the whole process, the followings can be stated: (a) works of the project have been broken down with agile WBS and the milestones have been defined with the documentation tool, (b) especially the TG-milestones have been transferred to the task management tool, (c) the transferred milestones have been associated with the tasks, (d) and features such as tracking and reporting provided by the tool for releases are also used for milestones.

As can be seen, contrary to some studies [13-15], milestones are directly associated with tasks. It has been decided that there will be no harm in using the tasks for mapping if they are not used for a release. Thus, a lot of information has been obtained such as the progress (more accurate) of a milestone, the determination of the personnel who contributed the most, the detection of the incomplete task that prevents reaching the goal and who is not completing the task. However, it should not be forgotten that every task is already linked/associated with stories. Therefore, access to the relevant story from a milestone is still possible.

3. BUILDING SCALABLE TEAM STRUCTURE

3.1. Current Approaches for Scaling Teams in Scrum

Companies may want to use agile methods in large-scale projects with large team size, cross-functional teams and geographically distributed personnel [16]. Without a detailed plan, it is becoming increasingly difficult for companies to manage scaled teams [17]. For this reason, many frameworks, approaches and models are used that offer scaling and have proven their success. It can be stated that the most used scaling approaches focused directly on the scrum framework are Scrum of Scrums (SoS, also known as Meta Scrum [18]), Scrum at Scale (S@S), Large Scale Scrum (LeSS), and Nexus [2, 3, 16, 17, 19-21].

SoS suggests splitting a large team into multiple smaller teams, each working on different tasks [22]. It has a structure where each team implements scrum separately, and an ambassador, who can be determined from each team depending on current issues, holds meetings with other ambassador(s) to coordinate/forward high-level updates and synchronize efforts [18, 23]. For the Scrum of Scrum meeting, team's scrum master can be chosen as the ambassador, as well as the product owner of the team or a member of the development team [18]. So, in SoS, there are two additional roles: (I) Chief Product Owner (CPO) who is overseeing each team's product owner and guiding them towards common goals, (II) Scrum of Scrum Master (SoSM) who assumes the role of scrum master for scrum of scrums meetings (ambassadors' meetings) [24]. Ambassadors may decide on the frequency of the scrum of scrum meeting and hold it once or twice a week instead of every day [18]. In SoS, a project usually has a single product backlog, while each team has its own sprint backlog, and for the cross-team improvements, a separate backlog are used [25].

S@S consists of scrum teams which are coordinated through SoS [26]. So processes are very similar to SoS. In parallel, CPO and SoSM roles are also defined in this approach [26-28]. In addition, S@S will need an Executive Action Team (EAT) that owns the implementation of scrum values and an Executive MetaScrum Team (EMT) that owns the vision and sets priorities [27]. In S@S, in parallel with SoS, there is common Product Backlog in which each team's Product Owner may generate independent items, and each team has its own sprint backlog [29]. Also two cycles are defined in the framework: the Scrum Master Cycle (for how to implement the system) and the Product Owner Cycle (for what should be implemented) [26, 30]. These cycles provide a strong support for managing the efforts and works of teams towards a single goal [28]. In addition to all these, there is also meeting called Scaled Daily Scrum, similar to the scrum of scrums meeting which held in SoS [27].

Nexus is a scaling framework that works on a single product backlog, similar to previous approaches [31]. The white paper prepared by Bourk and Kong [32] is provided general lines of Nexus Framework. According to the study, Nexus is based on the Scrum and introduces a new team named as Nexus Integration Team (NIT). This is responsible for producing an Integrated Increment at least every Sprint and takes ownership of any integration issues. Membership of the team can be part-time. In sprint planning, first of all, the Nexus Sprint Goal is determined. Then, Nexus Sprint Planning is carried out with representatives of each team. After this step, all representatives return to their teams and plan their teams' sprints. As a result of these activities, the Nexus Sprint Backlog is formed.

During the sprint, Nexus Daily Scrum (NDS) is held before the individual team's daily scrum meetings unlike SoS. At the end of the sprint, each team does not carry out Sprint Review separately, only the Nexus Sprint Review. After this activity, the Nexus Sprint Retrospective is held. This meeting consists of three parts: the first part identifying common challenges with representatives, the second part in which each team makes a Sprint Retrospective with its members for the identified challenges, and the third part that each representative meets again to discuss the actions for challenges. It should be noted that issues are not tried to be solved in the NDS, but only identified and expected to be fixed by the relevant individual team(s) [33]. In addition, with 10+ teams, Nexus+, more than one Nexus, should be used [32]

Similar to the Nexus, LeSS has two variations: Basic LeSS with 10-50 people and LeSS Huge that includes 8+ teams for 50-6000+ people [2, 34]. However, the 8 specified for the minimum number of teams in LeSS Huge has no special meaning and has been determined only as a result of experimental observation [35]. Conceptually, LeSS Huge allows to implement and manage multiple Basic LeSS [21]. LeSS Huge defines a new term, requirement area (RA) in which each team specialized [35]. Each RA has an Area Product Backlog (APB) that managed by an Area Product Owner (APO) who is working like Product Owner in Basic LeSS [2]. In [36], all LeSS (basic LeSS) rules are summarized and stated. According to this, Scrum Master role are supposed to be dedicated and he/she can serve up to three teams. Also, there is a single Product Owner, a single Product Backlog, and single project-wide Sprint. Sprint Planning is divided into two phases: the first phase in which the Product Owner and all members/representatives of teams selects tasks on which each team will work and the second phase in which each team decides how to perform the selected tasks. In Nexus, each team has individual daily scrum and separate Sprint Backlog.

In addition to all this, it should be noted that, to the coordination of teams with each other, there is no strict rule and it is recommended to use a decentralized informal method. For example, when a member of a team makes changes to the code, the member of the other team can see it and, if necessary, can communicate with the person who made the change. Also, before a change, if other teams need to be informed is made and this information is not shared in Sprint Planning, a meeting can be organized quickly. Lastly, at the end of the sprint, single Sprint Review and Overall Sprint Retrospective (OSP) should be held with the Product Owner, Scrum Masters, and all members/representatives of Teams. Also, before the OSP, each team is supposed to carry out Team Retrospective.

3.2. Defined Model and Activities

As cyber security projects require various expertises, specific personnel may be required to operate a task. It is ensured that these people continue as a separate team so that they can express their problems more clearly in daily scrums and coordinate more efficiently with people who are knowledgeable about the subject.

For building a scalable team structure, a LeSS-based approach has been developed due to its flexibility, lack of strict rules and not requiring many additional roles and activities. Differently, members of the associated team are not permanent and dedicated. After the Product Owner and all members/representatives of teams determine the sprint goal, they can decide to scale or shuffle teams according to the number of members who should/must work cooperatively for the goals. Equation 1 express the logic used to decide on scaling.

$$\mathrm{f}(T_i) = \begin{cases} 1 & \begin{cases} \text{if } \bar{T}_i > 10. \\ \text{if } C(T_i) \geq 4 \text{ and } \ddot{C}(T_i) \geq 4. \end{cases} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Where:

- T_i: represents the *i*th team
- $\bar{T}_i$: is the number of team members
- $C(T_i)$: is the number of cooperating members
- $\ddot{C}(T_i)$: is the number of non-cooperating members
- 1: indicates the necessity of scaling

The number 4 shown in the equation used in the scaling decision was determined as a result of both experimental observations in our project and the analysis in the study [37] conducted by Hackman and Vidmar. According to our observations and the mentioned study, productivity shows a noticeable increase in teams with 4+ members. For this reason, in team scaling, not only homogeneity and the need for each other of the personnel, but also productivity and efficiency were taken into consideration.

To summarize, in addition to the LeSS Framework, a process has been added for scaling teams based on goals or total number of members. This process can be called the Scaling and Assignment Activity (SAA) that should be done within sprint planning. In large projects, SAA can be carried out with APO, PO, and all team members/representatives.

4. EVALUATION

In the scope of milestone management, the SPI has defined many activities, from identification to integrating the process into existing software development life cycle tools. With the SPI, project personnel can be more proactive and effective in activities such as planning, tracking, reporting, collaboration, and even completion date prediction of a milestone. Additionally, software engineering techniques can be applied more efficiently. As all members are knowledgeable about objectives and their status, they can fully understand why techniques are being performed. Also, they are able to predict the progress of the project, detect the delay (if any), and offer corrective actions. They can determine which goal needs to be addressed first in sprint planning because they are able to see milestone's possible working times and expected completion time. In sprint planning, the goal is expressed with milestone identifiers (i.e., "ktMlStn1, ktMlStn2, ..."). In this way, it has been seen that the personnel can understand the goal more concretely and clearly. With this activity, the goals of the past sprints and the purpose for which they were carried out can be reported more easily and understandable. Additionally, performances of project personnel (including managers, leads, and subcontractors) are capable of being observed clearly. Because all tasks required to achieve a goal can be easily listed. When a goal cannot be reached before its deadline, it is possible to determine whose task is the reason for the delay. It can even be determined that the unfinished task is not broken down and planned properly by the manager. In addition, project progress is possible to be monitored more realistically because the current status of all tasks that necessary to achieve a goal can be clearly seen. In addition, estimates for the completion time of a milestone are made automatically based on the current performance. Also, detailed reports can be easily created for each goal. For a goal, it is possible to see which tasks have been completed and who has done these tasks, and the weight of each task (via story point) can be found. They know that the efforts made to reach the goal can be examined fairly. They are capable of comparing his own performance with the performance of other employees.

In addition, a model has been proposed that can scale (and shuffle, if necessary) the teams so that the resources can be used efficiently for the determined milestones. It has been very useful in ensuring homogeneity and preventing members from being exposed to issues that they are not related to or are not experts in. It also has been seen that creating a team for the personnel who will work closely for the relevant goal has prevented the meetings from being unnecessarily long for these members. Also, they can transfer their know-how to the relevant member(s) more efficiently. For all improvements, guides have been prepared. In addition, trainings and special meetings have been held at the institute/team level. Through these events, feedbacks have been acquired from different project managers, quality engineers, developers, and so forth.

5. CONCLUSION

In this study, the information obtained in the research phase of the SPI, the mature processes as a result of the SPI, and the gains acquired by the implementation of the improved processes have been provided. In addition, some situations that are not clearly expressed in the literature have been clarified with analyzes. With this study, it has been seen that many benefits can be obtained in planning, monitoring, and reporting by clearly defining many steps in milestone management and integrating the process into the software development life cycle tools. Also, it has been proven that milestones can be managed on the same infrastructure with release planning process and/or version management process. In addition, by scaling the teams according to a goal, members have been managed more efficiently in terms of time, motivation, and know-how transfer. Also, it can be mentioned that the proposed scaling model and the SPI can even be applied to all software projects in different disciplines. However, there are still areas for improvement in the relevant processes. In particular, milestones are defined via a template with the documentation tool. For this process, an improvement could not be made due to some limitations in the expectations of the client of the project. Future studies can focus on these processes as well. In addition, since dissemination studies have been initiated at the institute level, the experiences gained in these activities can also be shared.

REFERENCES

[1] F. Castillo, **Agile-Scrum Project Management**. Cham: Springer International Publishing, 199–210, 2016.

[2] O. Uludag, M. Kleehaus, X. Xu, and F. Matthes, “Investigating the role of architects in scaling agile frameworks”, **2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)**, Canada, 123–132, 2017.

[3] M. Paasivaara and C. Lassenius, “Scaling scrum in a large globally distributed organization: A case study”, **2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)**, USA, 74–83, 2016.

[4] M. Burghate, “Work breakdown structure: Simplifying project management”, *International Journal of Commerce and Management*, 3(2), 453–461, 2018.

[5] Project Management Institute, **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**, 6th ed., Newtown Square, PA: Project Management Institute, 2017.

[6] Internet: Institute of Project Management, What is a work breakdown structure (wbs)?, https://www.projectmanagement.ie/ blog/work-breakdown-structure/, 20.08.2022.

[7] M. Rianty, Y. Latief, and L. S. Riantini, “Development of risk-based standardized wbs (work breakdown structure) for quality planning of high rise building architectural works”, **MATEC Web of Conferences**, Indonesia, 159, 2018.

[8] Internet: G. Tekir, Wbs oluşturma – pmbok, https://gokremtekir.wordpress.com/2009/02/17/wbs-olusturma/, 18.08.2022.

[9] O. Adelakun, R. Garcia, T. Tabaka, and R. Ismail, "Hybrid project management: Agile with discipline", **International Conference on Information Resources Management (CONF-IRM)**, Chile, 2017.

[10] J. Reiff and D. Schlegel, "Hybrid project management–a systematic literature review", *International Journal of Information Systems and Project Management*, 10(2), 45–63, 2022.

[11] Internet: S. N. Dash, Work breakdown structure (wbs) in traditional and agile life cycles with ms project, https://www.mpug.com/work-breakdown-structure-wbs-intraditional-and-agile-life-cycles-with-ms-project/, 26.08.2022.

[12] C. Kenney, **Agile improvements to critical path method**, Tech. Rep., Naval Postgraduate School, Monterey, 2021.

[13] Internet: H. Purdy, How we use milestones & epics for the product management of shortcut, https://shortcut.com/blog/how-we-use-milestones-epics-product-management-clubhouse, 22.07.2022.

[14] Internet: D. Baranauskas, Automations for agile teams, https://help.favro.com/en/articles/4823164-automations-for-agile-teams, 14.08.2022.

[15] Internet: Shabbir, Use stories and epics to work more efficiently, https://articles.assembla.com/en/articles/1022084-use-stories-and-epics-to-work-more-efficiently, 15.08.2022.

[16] P. Diebold, A. Schmitt, and S. Theobald, "Scaling agile: how to select the most appropriate framework", *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 1–4, 2018.

[17] Internet: J. Berman, Agile at scale - movin' on up: scaling agile in large organizations, https://www.atlassian.com/agile/agile-at-scale, 24.08.2022.

[18] Internet: Wrike, Scrum of scrums meeting, https://www.wrike.com /scrum-guide/scrum-of-scrums/, 24.08.2022.

[19] H. Edison, X. Wang, and K. Conboy, "Comparing methods for large-scale agile software development: A systematic literature review", *IEEE Transactions on Software Engineering*, 48(8), 2709–2731, 2021.

[20] K. Conboy and N. Carroll, "Implementing large-scale agile frameworks: Challenges and recommendations", *IEEE Software*, 36(2), 44–50, 2019.

[21] F. Almeida and E. Espinheira, "Large-scale agile frameworks: A comparative review", *Journal of Applied Sciences, Management and Engineering Technology*, 2(1), 16–29, 2021.

[22] A. Almutairi and M. R. Qureshi, "The proposal of scaling the roles in scrum of scrums for distributed large projects", *Journal of Information Technology and Computer Science (IJITCS)*, 7, 68–74, 2015.

[23] A. Keller, A. R¨ossle, R. Sheik, H. Thell, and F. Westmark, **Issues with scrum-of-scrums**, Bachelor's Thesis, Chalmers Tekniska Högskola, 2019.

[24] Internet: C. Spanner, Scrum of scrums, https://www.atlassian.com /agile/scrum/scrum-of-scrums, 23.08.2022.

[25] M. C. Layton and D. Morrow, **Scrum for Dummies (For Dummies (Computers))**, 2nd ed. John Wiley & Sons, 2008.

[26] J. Sutherland, **The scrum@scale guide - the definitive guide to scrum@scale: Scaling that works**, draft Version 0.93, Tech. Rep., Scrum@Scale, 2018.

[27] Internet: C. Spanner, Organizational agility with scrum@scale, https://www.atlassian.com/agile/agile-at-scale/scrum-at-scale, 8.08.2022.

[28] J. Sutherland, **The scrum@scale guide - the definitive guide to scrum@scale: Scaling that works**, version 2.1, Tech. Rep., Scrum@Scale, 2022.

[29] V. Sithole and F. Solms, "Synchronized agile", **Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists**, 1–9, 2016.

[30] W. Alsaqaf, M. Daneva, and R. J. Wieringa, "Do the scaled agile practices from s@ s help with quality requirements challenges and if so, how do they do it?", **Proceedings of the 16th International Conference on Software Technologies (ICSOFT 2021)**, 441–452, 2021.

[31] K. Bittner, P. Kong, E. Naiburg, and D. West, **The Nexus Framework for scaling Scrum: Continuously Delivering an integrated product with multiple Scrum teams**. Addison-Wesley Professional, 2017.

[32] S. Bourk and P. Kong, **An introduction to the nexus framework**, Tech. Rep., Scrum.org, 2016.

[33] Internet: S. Flossmann, Why the nexus daily scrum is not scrum of scrums: 4 differences every scrum master needs to know, https://www.scrum.org /resources/blog/why-nexus-daily-scrum-not-scrum-scrums-4-differences-every-scrum-master-needs-know, 26.08.2022.

[34] Internet: T. E. OConnor, The large-scale scrum (less) framework, https://www.atlassian.com/agile/agile-at-scale/less, 18.08.2022.

[35] C. Larman and B. Vodde, **Large-scale scrum: More with LeSS**. Addison-Wesley Professional, 2016.

[36] Internet: The LeSS Company B.V., Less rules (december 2020) - large scale scrum (less), https://less.works/less/rules, 22.08.2022.

[37] J. R. Hackman and N. Vidmar, "Effects of size and task type on group performance and member reactions", *Sociometry*, 33(1), 37–54, 1970.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

Comparison of the Sizes of Assembly Codes Generated by the GCC Compiler for Different CPU Architectures

Esra ÇELİK, Deniz DAL

Ataturk University, Faculty of Engineering, Department of Computer Engineering, Erzurum, TURKEY
esra.celik@atauni.edu.tr, ddal@atauni.edu.tr

***Abstract*—** The effort to optimize all available resources has always been an important goal of mankind. This is also true for compiler technology. The purpose of a compiler is to produce code that leads to improved processor performance. Processor architectures play an essential role in this production and influence the runtime of the application or the required memory space. In this study, the sizes of assembly codes generated by the latest version of the GCC g++ compiler for six different processor architectures, namely ARM64, MIPS64, POWER64, s390x, x86_64, and RISC-V rv64gc, were investigated for the first time in the literature. In this context, a tool called Compiler Explorer was used. In the first phase of the study, a comparison dataset containing 24 different benchmarks was created in the C++ language, with known control and data structures, and functions. In the second phase, the Compiler Explorer tool was utilized to create six different assembly codes for each benchmark, using the appropriate compiler and processor architectures. In the final phase, the assembly code sizes were determined using the same tool. Experimental analysis showed that the x86_64 and RISC-V rv64gc processor architectures generated the most memory-efficient assembly codes, while the MIPS64 architecture produced the least efficient code in terms of the code size for all benchmarks among all architectures.

***Keywords*—** compiler, gcc, processor architecture, assembly code, code size, compiler explorer

1. INTRODUCTION

Today, the variety of computing applications, ranging from high-performance computing to artificial intelligence, from telecommunications to the automotive industry, is rapidly increasing. These applications, which seek high performance, require low memory usage, low power consumption, and high computational speed. To meet these requirements, developers try to create different types of processors that can handle different workloads.

A processor, also known as a Central Processing Unit (CPU), is a piece of hardware made up of billions of transistors and is considered the brain of a computer. It consists of three main components: Arithmetic Logic Unit (ALU), Control Unit (CU), and Register File (RF). The ALU performs arithmetic and logic operations. The CU is responsible for coordination between the ALU and the input/output elements. The RF is the component that contains all general and special-purpose registers within the CPU. The CPU controls the operation of the units in the computer and the flow of data between them. It also processes the instructions and sends the results back to the corresponding peripherals or memory.

Manufacturers produce different processor architectures to meet user expectations. Processor architecture refers to the technology, design and organization of the processor to perform operations. The history of processor architecture dates back to the von Neumann architecture, in which both instructions and data are stored in main memory [1]. Today, there are several processor architectures developed based on this architecture. Among these architectures, ARM, MIPS, POWER, s390, x86_64, and RISC-V are the ones commonly used in the literature.

ARM (*A*dvanced *R*ISC *M*achine) was developed by ARM Ltd in 1985 [2]. It is a RISC (*R*educed *I*nstruction *S*et *C*omputing) type processor architecture designed to execute fewer types of computer instructions [3]. This architecture, whose first version is ARMv1, also has different models with 32 (ARM32) and 64 (ARM64) bits. Thanks to its low power consumption and high-performance capabilities, it is the most widely used processor in embedded systems today.

MIPS (*M*icroprocessor without *I*nterlocked *P*ipeline *S*tages) is a RISC architecture developed by researchers at Stanford University in 1985 [4]. One of the three RISC processors produced in 1990 was based on the MIPS architecture. Although early designs of the architecture had 32 bits, newer generations have 64 (MIPS64) bits. MIPS is commonly used in SGI computers and embedded systems.

POWER is a RISC architecture developed by IBM in 1990 [5]. This architecture, whose first version was RIOS-1, also has different models with 32 and 64 (POWER64) bits. It is mainly used for virtualized workloads and high-performance work, such as password processing.

The s390 (System/390) was developed by IBM in 1990 [6]. It is a CISC (*C*omplex *I*nstruction *S*et *C*omputing) type processor architecture ([7]) designed for the development of simpler compilers. The s390 model of the architecture is 32-bit, while the s390x model is 64-bit. It is used in various Linux operating systems and open-source applications.

x86_64 (x64, AMD64 or Intel 64) is a CISC architecture developed by AMD in 2000 [8]. Today, it is also used by Intel. The x86_64 is a 64-bit superset of x86 and supports it natively. It has been widely used in Windows PCs in recent years.

RISC-V (risk-five) is an open-source RISC architecture developed at the University of California Berkeley in 2010 [9]. This architecture, whose first version was RVWMO, also has different models with 32 (RISC-V rv32gc) and 64 (RISC-V rv64gc) bits. It is ideal for embedded applications, such as the IoT ecosystem and automotive applications.

The codes that are meant to be executed on the processor must first be converted into a machine language that is compatible with the processor's instruction set architecture. One of the most basic tools for this transformation are compilers. Thanks to compilers, it is possible to effectively execute software on hardware. Today, there are many different compilers for each processor architecture. GCC (*G*NU *C*ompiler *C*ollection) is one of these compilers and offers a wide range of architectural support.

The GCC was developed in 1984 by Richard Stallman, founder of the GNU project. It was originally named GNU C because it initially supported only the C programming language [10]. Over time, the language support was extended and it became the GNU Compiler Collection. GCC is a system compiler for Linux-based operating systems and is 100% free software [10, 11]. GCC g++ is an open-source C++ compiler that is included in GCC. The latest version of the compiler is GCC 12 which is also utilized in this study.

The GCC performs the conversion by selecting and combining the appropriate instructions from the processor's instruction set architecture. Depending on which instructions are selected from this architecture set in this process, different conversions with the same functionality are possible. Therefore, performance differences due to the occupied memory spaces of the codes generated by the same compiler for different processor architectures are very likely. Motivated by this fact, this study aims to analyze the sizes of the assembly codes generated by GCC g++ for ARM64, MIPS64, POWER64, s390x, RISC-V rv64gc, and x86_64 architectures for the first time in the literature.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the dataset and the software tool used, and explains the research methodology. The experimental results of the study are analyzed in Section 4. Finally, Section 5 concludes the paper and highlights opportunities for future work.

2. RELATED WORK

Some studies in the literature compare the performance of different compilers in terms of runtime, memory, and energy consumption criteria. There are several studies on the importance of compilers for runtime optimization [12-15]. One study examines the result-based performance of the default optimization levels of the GCC 4.6.1 compiler for Intel Core i7 and Intel P4 processors [16]. In another study, GCC and LLVM compilers for the Intel Skylake architecture were used to determine energy efficiency [17]. The results of the corresponding study show that different compilers have a significant impact on the quality of the generated code in terms of execution times and energy consumption. In memory-oriented scientific approaches, there are studies on reducing program code size and saving memory by using compiler optimization flags [18-21]. In addition, there are some studies in the literature on saving memory through methods, such as reducing code size by memory optimization, eliminating sub-expressions, and cleaning up dead code [22, 23].

As can be concluded from the discussion above, researchers are very interested in studies that compare different compilers in terms of runtime, memory, and energy. However, to the best of our knowledge, this study is the first one that analyzes assembly code sizes generated by the same compiler for different processor architectures.

3. MATERIAL AND METHOD

3.1. Dataset

In the world of data science, there is a wide variety of data to explore. This data can play a critical role in solving many problems. However, the existence of special data needed for some studies is a different reality. In this study, for which a special dataset is needed, a dataset containing a total of 24 different comparative benchmarks is created in the C++ language. This so-called dataset consists of four subsets: (1) Control statement-related benchmarks, which include if and switch cases, for and while loops, and break and continue statements. (2) Data structure-related

benchmarks, which include static and dynamic arrays, STL vector, stack and queue. (3) Function type-related benchmarks, which include standard library functions, user-defined function calls by value and reference, and recursive functions. (4) Function overload-related benchmarks, which include overloads with various data types, numbers, sequences and parameter types.

3.2. Compiler Explorer Tool

Compiler Explorer is an extremely useful and interactive web application developed by Matt Godbolt in 2012 [24]. It is sponsored by Solid Sands, the leading provider of compiler and library technology.

The tool supports many different programming languages, processor architectures, and compilers. It converts any C++ source code (or code from any of the other supported languages) into assembly code for a processor architecture using a compiler supported by that particular architecture. At this stage, the tool outputs such as warnings/errors generated by the compiler, assembly code, runtime and amount of memory used are displayed. Compiler Explorer is a great way to learn assembly code, understand how compilers work, see what optimizations are possible, and prove or disprove myths about what makes efficient code.

3.3. Methodology

There are many different processor architectures supported by a compiler. What distinguishes processor architectures from each other is the way they perform operations, depending on which instruction set architecture they utilize. This difference allows the same compiler to produce many different assembly codes with the same functionality for a given architecture from one high-level code. In parallel with this situation, the runtime and memory requirements of the application may also change. Motivated by this fact, this study examines the memory requirements of the compiled assembly codes generated by the latest version of GCC g++ for ARM64, MIPS64, POWER64, s390x, x86_64, and RISC-V rv64gc processor architectures.

The basic programming structures (loops, arrays, functions, etc.) used in a program affect its runtime and the amount of memory occupied. Similarly, the sizes of the assembly codes generated by the same compiler for different processor architectures may vary. Based on this observation, 24 different benchmarks were created in C++ during the first phase of the study. The high-level representation of the while-loop benchmark, which is the running example of the study, is shown in Figure 1.

In the second phase of the study, six different assembly codes were generated for each benchmark using the Compiler Explorer Tool, the latest version of the GCC g++ compiler, and the ARM64, MIPS64, s390x, POWER64, RISC-V rv64gc, and x86_64 processor architectures. Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, and Figure 7 show the assembly code outputs generated by the GCC g++ compiler for the while-loop benchmark for each architecture. These outputs include assembly code with filtered data of unused labels, library functions, directives and comments, as well as the compile time and size of the assembly code in byte (indicated by the red box in the figures). Given the same compiler and architecture, the runtime varies depending on the intensity of use of the application, but the memory occupied by the code remains the same. This makes it possible to examine the memory requirements as a function of the processor architecture.

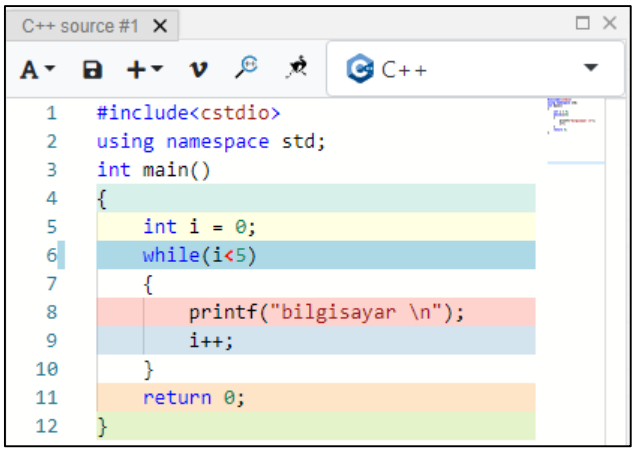


Figure 1. C++ code of the while-loop benchmark

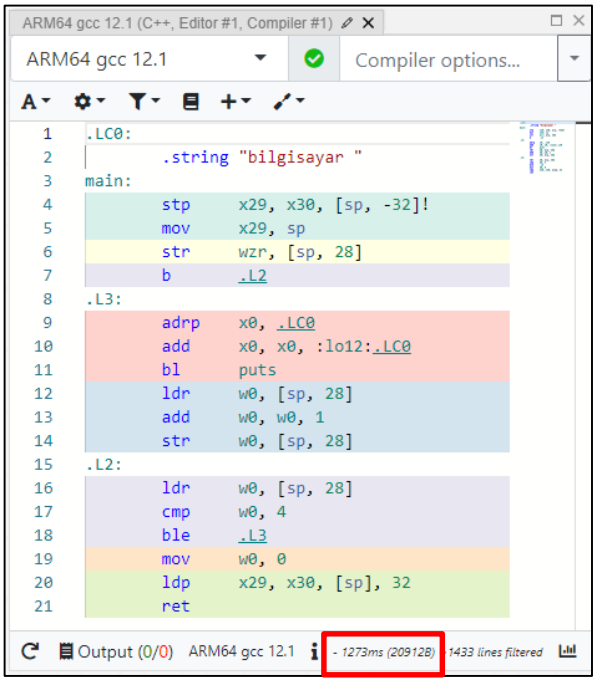


Figure 2. Assembly code of the while-loop benchmark generated for the ARM64 architecture

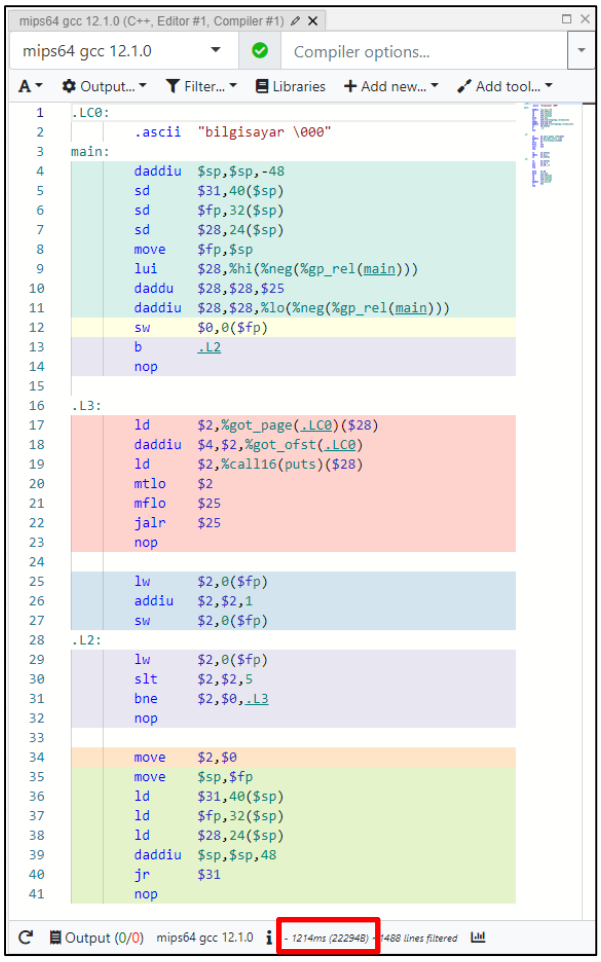


Figure 3. Assembly code of the while-loop benchmark generated for the MIPS64 architecture

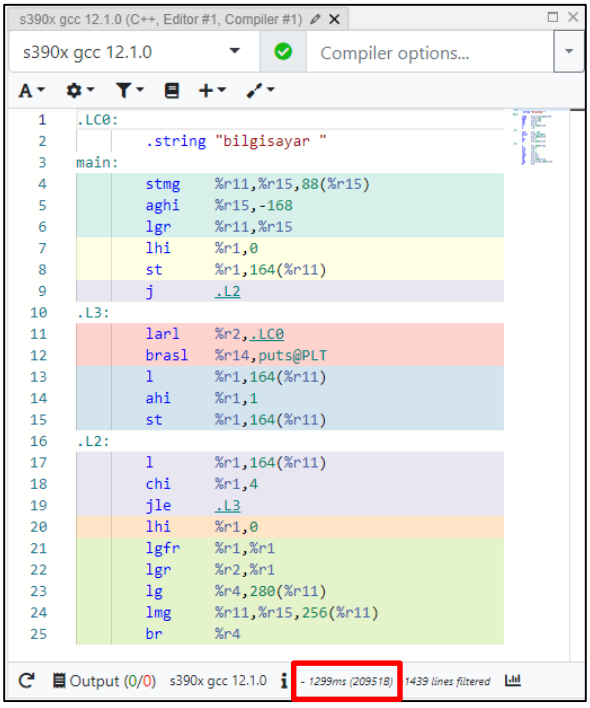


Figure 4. Assembly code of the while-loop benchmark generated for the s390x architecture

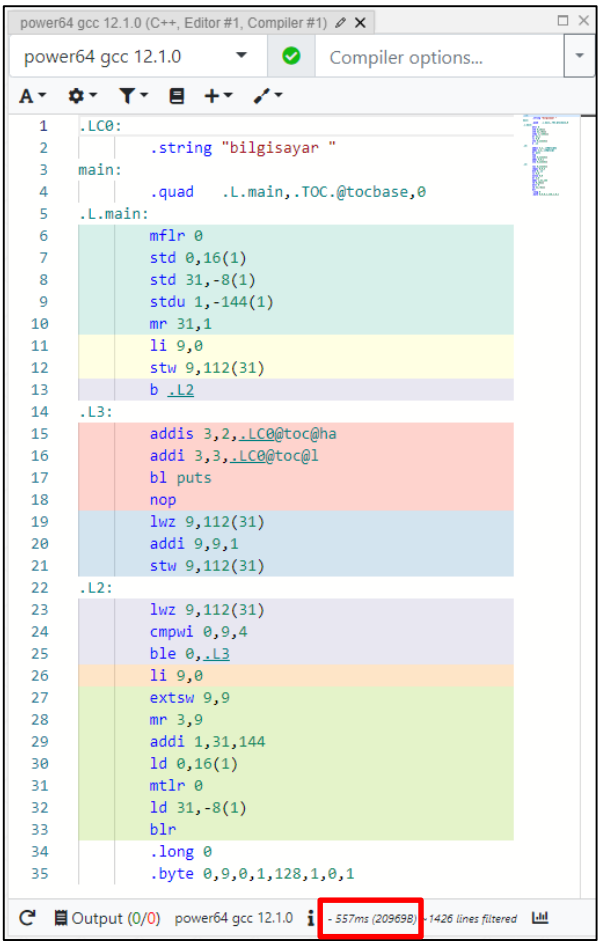


Figure 5. Assembly code of the while-loop benchmark generated for the POWER64 architecture

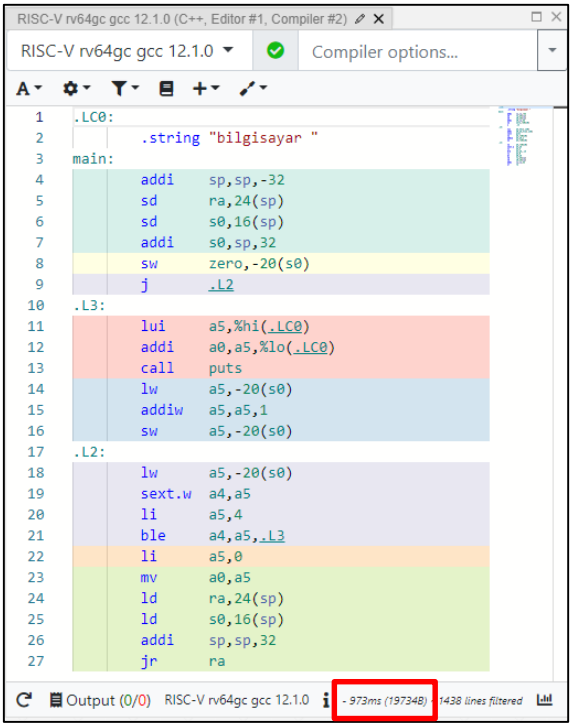


Figure 6. Assembly code of the while-loop benchmark generated for the RISC-V rv64gc architecture

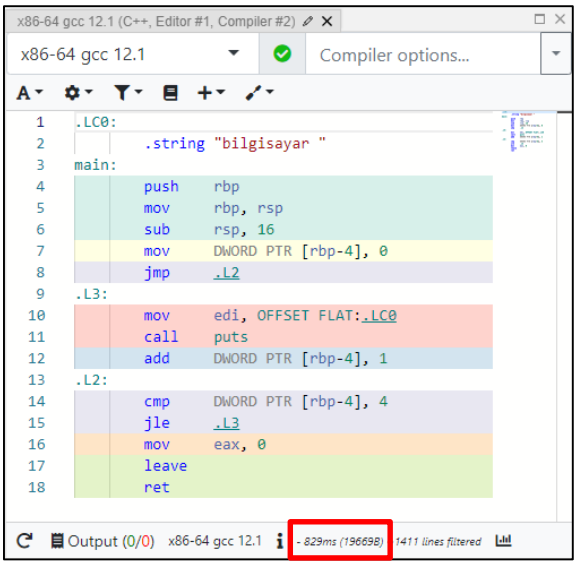


Figure 7. Assembly code of the while-loop benchmark generated for the x86_64 architecture

4. EXPERIMENTAL RESULTS

The experimental results of the study are listed in Table 1. For each benchmark, the values highlighted in bold and italic indicate the maximum memory size, while the values highlighted in only bold represent the minimum memory size in bytes.

The following four findings are extracted from this table, with each finding corresponding to a different category of benchmark subset.

(1) In the control statement-related benchmarks, x86_64 requires the least memory consumption of all architectures, while MIPS64 requires the most. In terms of the generated assembly code sizes, the x86_64 architecture contains at least 11% and at most 13% less code in bytes than the MIPS architecture.

(2) In the data structure-related benchmarks, RISC-V rv64gc requires the least amount of memory consumption among all architectures, while MIPS64 requires the most. In terms of the generated assembly code sizes, the RISC-V rv64gc architecture contains at least 11% and at most 22% less code in bytes than the MIPS64 architecture.

(3) In the function type-related benchmarks, the RISC-V rv64gc requires the least memory for the first five benchmarks in this category, while x86_64 requires the least for the remaining three benchmarks. On the other hand, MIPS64 performs the worst for all benchmarks. The x86_64 architecture contains at least 11% and at most 14% less code in bytes than MIPS64.

(4) In function overload-related benchmarks, RISC-V rv64gc consumes the least memory in one half of the benchmarks, while x86_64 consumes the least memory in the remaining half. On the other hand, MIPS64 performs the worst for all benchmarks. The RISC-V rv64gc architecture contains at least 14% and at most 15% less code in bytes than MIPS64.

In summary, the analysis shows that the x86_64 and RISC-V rv64gc architectures have the lowest memory requirements, while the MIPS64 architecture consumes the most memory in terms of assembly code for all benchmarks.

Table 1. Compiler Explorer results for all benchmarks from the dataset

Dataset	**Outputs (Byte)**					
	ARM64	*MIPS64*	*POWER64*	*s390x*	*x86_64*	*RISC-V rv64gc*
Control Statement-Related						
If-Else	21210	***22699***	21281	21242	**19996**	20015
Switch-Case	21577	***23390***	21666	21573	**20326**	20441
For Loop	21148	***22536***	21205	21187	**19903**	19949
While Loop	20912	***22294***	20969	20951	**19669**	19734
Do While Loop	20938	***22314***	20995	20977	**19693**	19760
Break	21455	***22850***	21518	21511	**20240**	20263
Continue	21439	***22834***	21502	21495	**20224**	20247
Data Structure-Related						
Static Array	21553	***22801***	21583	21845	20370	**20228**
Dynamic Array	21961	***23765***	22004	22329	20936	**20682**
STL Vector	180353	***222479***	192777	193914	179131	**176399**
Stack	258082	***325342***	275977	279080	257898	**252595**
Queue	260048	***328115***	278164	281303	259836	**254596**
Function Type-Related						
Standard Library Function	51976	***55583***	52922	52580	50531	**49018**
User Defined Function Call by Value	23427	***25443***	23427	23845	22410	**22177**
User Defined Function Call by Reference	23589	***25494***	23921	24009	22590	**22290**
Function with Argument and Return Value	22845	***25000***	23157	23153	21645	**21609**
Function with Argument but No Return Value	22681	***24848***	22975	22953	21462	**21446**
Function with No Argument but Return Value	22043	***24242***	22350	22308	**20762**	20868
Function with No Argument and No Return Value	21625	***23835***	21915	21854	**20324**	20487
Recursive Function	22689	***24880***	22974	23031	**21466**	21513
Function Overload-Related						
Overloading Using Different Data Types of Parameters	24133	***27280***	25049	24735	23104	**23091**
Overloading Using Different Number of Parameters	24015	***26868***	24608	24526	**22757**	22848
Overloading Using Different Sequence of Parameters	24224	***27228***	25058	24803	23229	**23164**
Overloading Using Different Return Types & Data Types of Parameters	23829	***26677***	24560	24372	**22729**	22738

5. CONCLUSION

In this study, the memory sizes of the assembly codes generated by the GCC g++ compiler for the ARM64, MIPS64, POWER64, s390x, x86_64 and RISC-V rv64gc processor architectures were investigated. Unlike other studies in the literature, this study is the first one to compare the sizes of the assembly codes generated by the same compiler for different processor architectures.

The study reveals three important findings: (1) The x86_64 (CISC family) and RISC-V rv64gc (RISC family) processor architectures provide a more ideal memory consumption in terms of assembly code sizes compared to other architectures. (2) For the MIPS64 architecture, assembly codes take up the most memory. (3) CISC computers use fewer instructions to perform complex tasks, while RISC computers have a smaller set of simple instructions. Therefore, CISC computers generally consume less space than their RISC counterparts. On the other hand, in the context of this study, RISC-V rv64gc does a competitive job although it is a RISC computer.

The discrepancies between the processor architectures in terms of the size of the generated assembly codes might be due to the compiler's capability or the instruction set architectures of the processors in question. Running the same experiments with a different compiler and taking the compiler optimization flags into account could be one of the future directions.

REFERENCES

[1] J. von Neumann, "First Draft of a Report on the EDVAC", **IEEE Annals of the History of Computing**, 15(4), 27-75, 1993.

[2] J. V. Vijay, B. Bansode, "ARM Processor Architecture", *International Journal of Science, Engineering and Technology Research (IJSTR)*, 4(10), 2015.

[3] W. Stallings, "Reduced Instruction Set Computer Architecture", **Proceedings of the IEEE**, 76(1), 38-55, 1988.

[4] D. A. Patterson, "Reduced Instruction Set Computers", *Communications of the ACM*, 28(1), 8-21, 1985.

[5] T. Karkhanis, J. E. Moreira, **IBM Power Architecture**, IBM Research Report, 2011.

[6] H. Penner, U. Weigand, "Porting GCC to the IBM S/390 platform", **In GCC Developers Summit**, 195-212, 2003.

[7] T. Jamil, "RISC versus CISC", **IEEE Potentials**, 14(3), 13-16, 1995.

[8] Internet: M. Kerner, N. Padgett, A History of Modern 64-bit Computing, https://courses.cs.washington.edu/courses/csep590/06au/projects/history-64-bit.pdf, 12.09.2022.

[9] A. Waterman, Y. Lee, D. Patterson, K. Asanovic, V. I. U. level Isa, A. Waterman, D. Patterson, **The RISC-V Instruction Set Manual**, User-Level ISA, 2014.

[10] Internet: Free Software Foundation, Inc. GCC, the GNU Compiler Collection, https://gcc.gnu.org/, 12.09.2022.

[11] D. Novillo, "GCC an Architectural Overview, Current Status, and Future Directions", **In Proceedings of the Linux Symposium**, 2, 185, 2006.

[12] G. Attiya and Y. Hamam, "Task Allocation for Maximizing Reliability of Distributed Systems: A Simulated Annealing Approach", *Journal of Parallel and Distributed Computing*, 66(10), 1259–1266, 2006.

[13] M. Angelova and T. Pencheva, "Tuning Genetic Algorithm Parameters to Improve Convergence Time", *International Journal of Chemical Engineering*, 2011.

[14] H. Kallwies, M. Leucker, M. Prilop, & M. Schmitz, "Optimizing Trans-Compilers in Runtime Verification Makes Sense–Sometimes", **In International Symposium on Theoretical Aspects of Software Engineering**, Springer, Cham, 197-204, 2022.

[15] J. Doerfert, A. Patel, J. Huber, S. Tian, J. M. M. Diaz, B. Chapman, & G. Georgakoudis, "Co-Designing an OpenMP GPU runtime and optimizations for near-zero overhead execution", **In 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)**, 504-514, 2022.

[16] R. D. Escobar, A. R. Angula, and M. Corsi, "Evaluation of GCC Optimization Parameters", *Revista Ingenierias USBmed*, 3(3), 31–39, 2012.

[17] A. Shivam, **A Multiple Compiler Approach for Improved Performance and Efficiency**, Doctoral Dissertation, University of California, Irvine, 2021.

[18] R. C. O. Rocha, P. Petoumenos, Z. Wang, M. Cole, and H. Leather, "Function Merging by Sequence Alignment", **In Proceedings of the 2019 IEEE/ACM International Symposium on Code Generation and Optimization, ser. CGO 2019**, Piscataway, NJ, 149–163, 2019.

[19] T. Damasio, V. Pacheco, F. Goes, F. Pereira, and R. Rocha, "Inlining for Code Size Reduction", **In 25th Brazilian Symposium on Programming Languages, ser. SBLP'21**, New York, NY, 17–24, 2021.

[20] S. K. Debray, W. Evans, R. Muth, and B. De Sutter, "Compiler Techniques for Code Compaction", *ACM Trans. Program. Lang. Syst.*, 22, 378–415, 2000.

[21] S. Jain, Y. Andaluri, S. VenkataKeerthy, & R. Upadrasta, "POSET-RL: Phase ordering for Optimizing Size and Execution Time using Reinforcement Learning", **In 2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)**, 121-131, 2022.

[22] R. Kennedy, S. Chan, S. M. Liu, R. Lo, P. Tu, and F. Chow, "Partial Redundancy Elimination in SSA Form", *ACM Trans. Program. Lang. Syst.*, 21, 627–676, 1999.

[23] L. Torczon and K. Cooper, **Engineering A Compiler**, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[24] Internet: M. Godbolt, Compiler Explorer, https://godbolt.org/, 12.09.2022.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

Text Preprocessing Steps for Patent Documents: A Reconsideration of Standardized Approaches

Furkan İNCEDAYI [1], Hakan ÇERÇİOĞLU [2]

[1] Graduate School of Informatics, Gazi University, Ankara, Turkey
[2] Department of Industrial Engineering, Gazi University, Ankara, Turkey
furkan.incedayi@gazi.edu.tr, cercioglu@gazi.edu.tr

***Abstract*—** Text preprocessing techniques may have a large impact on a model that attempts to provide accurate text classification. This study focuses on the reconsideration of text preprocessing techniques prior to applying a topic modeling approach, such as latent dirichlet allocation (LDA), to unstructured patent abstracts. LDA is useful for inferring topics hidden in the patent abstracts to monitor the trends in a certain domain. Nevertheless, the differences between text preprocessing steps regarding patent-related texts have not been put forward. Our experiments applied on a dataset retrieved from Turkish Patent and Trademark Office show that the stop-word removal method along with whether the unigram tokens of a document collection should be stemmed or lemmatized have a non-negligible impact on the resulting set of topics. We present the finding that lemmatization, instead of stemming, does not only improve the semantic coherence of topics, it also prevents the human interpreters from making wrong assumptions about the original meaning of a corresponding term. Lemmas, as opposed to stems, draw a much clearer path for the topic labeling in exchange for computational intensity as the main trade-off. Stop-word removal, a common method to eliminate non-informative terms, is one of the standardized steps of text preprocessing in natural language processing applications. However, in the patent domain, we propose an approach that building an additional in-context stop-word dictionary results in a more distinctive model output in terms of face validity. Thus, practitioners are likely to acquire a more distinguishable set of topics when corpus-specific stop words of the technical patent language, in addition to generic stop-words, are filtered out.

***Keywords*—** text preprocessing, topic modeling, LDA, patents

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

An IaaS Instance Selection Decision Model for Cloud Computing

Onur KOŞAR [1], Üzeyir FİDAN [2], Mehmet ATAK [3]

[1]Management Information Systems Department, Gazi University, Ankara, Türkiye
[2]Distance Education Vocational School, Uşak University, Uşak, Türkiye
[3] Industrial Engineering, Gazi University, Ankara, Türkiye
onur.kosar@gazi.edu.tr, uzeyir.fidan@usak.edu.tr, matak@gazi.edu.tr

***Abstract*—** In the scope of Infrastructure as a Service(IaaS) model of cloud computing; virtual servers, disk storage, network bandwidth and physical hardware can be rented and easily scaled with the variety of customer needs. Virtual server instance rental is one of the basic services offered by cloud computing service providers. An instance basically consists of virtual CPU, memory, disk and dedicated or shared network bandwidth. Moreover, there are pre-prepared virtual server products with different configurations and located in different locations available in service providers' product range. Selecting the appropriate instance for intended workload is a difficult task when the product option list is large and when there are multiple criteria to be considered. In this study in order to solve the IaaS instance selection problem, an Entropy-based VIKOR Multi-Criteria Decision Making model was proposed and the model was tested with different customer request scenarios on the Amazon EC2 product database.

***Keywords*—** cloud computing, IaaS instance selection, Amazon EC2, Multi Criteria Decision Making

1. INTRODUCTION

Cloud Computing provides great convenience in accessing and using information resources. It is a model where configurable and scalable computing resources can be accessed over computer networks with minimal administrative effort and interaction. It has three main service models; Software as a Service, Platform as a Service, Infrastructure as a Service [1]. IaaS, the lowest level cloud computing service, is a structure that allows customers to use services such as virtual servers, virtual storage and virtual computer networks as local computing resources [2]. In the IaaS model, resources are prepared in advance according to customer needs and billed as much as they are used [3]. Customers can rent cloud computing services over the internet with the option to pay as they go. Cloud service providers(CSPs) bill their virtual server products, called instances according to the amount of resource reserved and duration that it is rented [4]. Virtualization technology is the foundation of cloud computing and enables shared use of server computers hardware, disk space and network resources between customers [5-6]. With the help of virtualization, users are isolated from each other, resources can be shared and computing capacity can be dynamically scaled [7]. Server virtualization aims to make efficient use of resources by offering processors, memory and other server resources to users with the facilities of different operating systems [8].

Cloud virtual machine instances are the basic product of IaaS service of the cloud computing provider companies. A virtual machine generally consists of virtual CPU, memory, disk space and virtual network interfaces. CSPs offer virtual server products to their customers as ready-made packages configured considering the user workloads and needs, or as packages created at the time of customer request is taken. Cloud service providers differ from each other by the locations where they provide their products, types of virtual servers and different pricing policies [9].

Since many service providers offer a large number of products in the IaaS tier, choosing among them is a very difficult process and becomes a multi-criteria decision making problem since it requires the evaluation of many criteria together [10].

Objective of this study is to propose a method to help decision makers that need cloud computing virtual servers to find the products and alternatives suitable for their needs in large-scale product repositories. In order to solve the IaaS instance selection problem, the Entropy-based VIKOR Multi-Criteria Decision Making model is proposed and the model is tested with different customer request scenarios in the Amazon EC2 product lists [11,12]. The proposed model is tested with processor and storage-prioritized customer requests and use case scenarios.

2. RELATED WORK

In cloud computing, when the features of the product are evaluated together with its offerings, the selection of a suitable product turns into a decision process in which many criteria are evaluated together [13].

Hussain et al. [14] proposed cloud service selection framework as a service method using Triangular Fuzzy Numbers [15] in determination of criteria weights to make decision makers judgment more precise. In this work, high CPU requested computing product selection and IaaS Service selection performance were compared to other MCDM models.

In the study by Gerasimou and et al. [16], quality of service based cloud service ranking solution by the use of Satisfiability Modulo Convex (SMO) Optimization [17] for ranking service oriented computing offerings suggested. They used SMO as a specification language and execution time of a customer request was taken for ranking the available computing services. Fast Cloud Service Ranking System is proposed in [18] for ranking cloud services. In this study numerous cloud services with multiple attributes were eliminated according to binary encoded user requests within intervals and therefore limited number of services according to user requests were compared to each other.

Generally, studies that deal with IaaS instance selection problem, instance attributes like vCPU, Ram, Disk, Bandwidth and Price are taken into consideration for selection process. On the other hand, cloud service providers have many options in their IaaS product lists and that the prices of the same products vary in different data center locations makes the selection problem difficult for the customer. AHP [19], one of the main MCDM methods, is a widely used in service and instance selection problem of cloud computing. Pareto Dominance and AHP was used in [20] for service selection with datacenter location consideration as sub criteria. Different cloud vendor IaaS options were tested by synthetic workloads and performance results were compared by AHP is another MCDM model in [21]. TOPSIS and VIKOR techniques were used in [22] for the IaaS service selection problem. Alternatives distance to the ideal solution were calculated by TOPSIS and for determination of outranking candidates VIKOR is used.

Fuzzy AHP method was used in [23] to service selection problems of IaaS products. Expert opinions were taken for pairwise comparison of alternatives and these comparisons were presented as fuzzy numbers in order to reduce human ambiguity in the decision process. In calculating the weights, weights were calculated from fuzzy numbers of pairwise comparison by the method of Fuzzy Extend Analysis [24].

3. METHOD

In this part of the study, the methods used for the cloud virtual machine product selection problem are presented. In this context, the MCDM method applied for obtaining data and applying data preprocessing steps, determining criteria and calculating criteria weights, and finally ranking the alternatives are emphasized.

3.1. Data Acquisition and Data Preprocessing

Test data was gathered from Amazon EC2 configuration lists and price calculator applications [11,12]. It was imported into PostgreSQL relational database as a table in the format of 'product id', 'location id', 'product name', 'vCPU', 'Ram', 'Disk', 'Bandwidth' and 'Price'.

Since some products require usage of disk space that can be allocated according to customer demand over computer networks, the disk sizes of these products are included as '0' in the data set. These product groups were excluded from the dataset as they would make it difficult to compare with the product configurations that contain all attributes of an instance. In total 7654 rows of product data, 4315 product configurations with '0' disk space were eliminated and a total of 3339 rows of table was used. The dataset contains 1497 different price information of 275 different products in 65 different locations. Sample scenarios were applied to the model with data from 7 different views created by SQL queries. Sample product and location data are shown in Table 1-2.

Table 1. CSP product sample data

Product Name	Specifications					
	location id	*vCPU*	*Ram (GB)*	*Disk (GB)*	*BW (Gbps)*	*Price ($)*
R3.large	1	2	15	32	3	121.26
C5d.2xlarge	8	8	16	200	10	350.63
I3large	3	2	15	475	10	132.21

Table 2. CSP location sample data

CSP	Specifications	
	location id	*Location Name*
Amazon	1	N. Virginia (useast)
Amazon	2	Atlanta (useast)
Amazon	33	London
Amazon	37	Bahrain
Amazon	38	Saopaulo

3.2. Calculation of Criteria Weights with Entropy Method

In the decision problem, Entropy, an objective method, was used to calculate the weights of the criteria [25]. The smaller the Entropy value obtained, the higher the information contained in the criteria, assuming the higher the weight of the criteria, the calculation steps are given in Eq. 1-3.

$$p_{ij} = \frac{x_{ij}}{\sum_{j=1}^{n} x_{ij}} \quad (1)$$

The p_{ij} values in Eq. 1 show the standardized version of the i index in the j.th example. Thus, the Entropy value E_i is calculated as shown in Eq. 2.

$$E_i = -\frac{\sum_{j=1}^{n} p_{ij} . ln p_{ij}}{ln ln\, n} \quad (2)$$

Since the information obtained with the entropy values is inversely proportional, the criteria weights (w_i) are calculated with the help of Eq. 3.

$$w_i = \frac{1-E_i}{\sum_{i=1}^{m} (1-E_i)} \quad (3)$$

3.3. Ranking of Alternatives Using the VIKOR Method

VIKOR is a MCDM method that aims to determine the closest alternative to the ideal solution in decision problems where criteria with different characteristics are handled together [26]. The steps of the VIKOR method are given in Eq. 4-8.

First, the best (f_j^+) and worst (f_j^-) criteria values are determined by Eq. 4.

$$f_j^+ = x_{ij}\ ,\ f_j^- = x_{ij} \ \text{(j. criteria benefit-based)}$$
$$f_j^+ = x_{ij}\ ,\ f_j^- = x_{ij} \ \text{(j. criteria cost-based)} \quad (4)$$

The data normalized with the f_j^+ and f_j^- values obtained here are multiplied by the criterion weights and the S_j and R_j values are calculated with the help of Eq. 5-6.

$$S_j = \sum_{j=1}^{n} w_j(f_j^+ - x_{ij})/(f_j^+ - f_j^-) \quad (5)$$

$$R_j = [w_j(f_j^+ - x_{ij})/(f_j^+ - f_j^-)] \quad (6)$$

Finally, the method ends with calculating the Q_j values and ranking the alternatives.

$$Q_j = q.\frac{S_j - S^*}{S^- - S^*} + (1-q).\frac{R_j - R^*}{R^- - R^*} \quad (7)$$

In Eq. 7, the S^* value is S_j , the S^- value is S_j , the R^* value is R_j , and the R^- value is R_j . In addition, the q value shows the maximum group benefit, and the $1-q$ value shows the minimum regret of the dissidents. The solution list is obtained by ordering the alternatives from smallest to largest according to Q_j values. Although the VIKOR solution includes final rankings of alternatives, acceptable advantage and acceptable stability conditions are expected to be tested.

The Acceptable Advantage: When the alternatives are ordered from smallest to largest according to Q_j values, the first alternative is A^1 and the second alternative is A^2,

$$Q(A^2) - Q(A^1) \geq \frac{1}{m-1} \quad (8)$$

The m value in Eq. 8 shows the number of alternatives in the decision problem. If the acceptable advantage condition is not met, it is checked for all Q_j values, respectively, to find the value where the condition is satisfied. Thus, all alternatives up to the value where the condition is satisfied are accepted as compromise solutions.

The Acceptable Stability: When sorted from smallest to largest according to Q_j values, if the first alternative A^1 is also in the first place in the list created according to S or R values, then A^1 alternative is accepted as a compromise solution. If the condition is not met, both A^1 and A^2 alternatives are accepted as compromise solutions.

3.4. Hierarchical Structure of the Decision Model

The model that was created for determining the decision problem (purpose) and determining the criteria to be considered for solving the problem in order to rank the decision alternatives in MCDM methods is presented in Fig. 1.

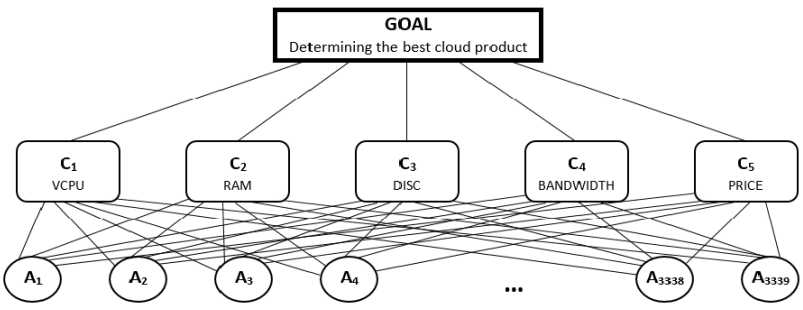


Figure 1. Hierarchical structure of the decision model

In Figure 1, there are 5 criteria (vCPU, Ram, Disk, Bandwidth, Price) and 3339 alternatives in the hierarchical model created to determine the best cloud virtual machine product. Alternatives were formed from products that differ in their criteria features or that are located in different locations with similar criteria features.

3.5. Test Scenarios

Within the scope of the study, 7 different sample scenarios were tested. These scenarios are, in order,

Scenario 1: When no constraints are specified,
Scenario 2: When the number of CPUs is 8 or more,
Scenario 3: When the RAM size is 16 GB or more,
Scenario 4: When the hard disk size is 1000 GB or more,
Scenario 5: When the data transfer rate is 20 Gbps or more,
Scenario 6: When the monthly product price is $100 or less,
Scenario 7: The number of CPUs is 16 or more, the RAM size is 32 GB or more, the hard disk size is 4000 GB or more, the data transfer rate is 20 Gbps or more, and the monthly product price is $2500 or less.

According to the decision maker's preference, 797040 different scenarios can be created for all possible configurations except the price variable. With the addition of the price variable, the number of possible scenarios exceeds 1 billion.

4. RESULTS

In this study, Entropy-based VIKOR method is proposed for the selection of the best suitable cloud virtual machine product in the Amazon EC2 product database. Amazon EC2 database contains 3339 products in 65 different locations. 5 criteria of the products in the decision-making process; number of processors (vCPU), memory size (Ram), hard disk size (Disk), data transfer rate (Bandwidth) and price (Price) were taken into consideration. Summary information about these criteria is presented in Table 3.

Table 3. Summary statistics of criteria

	Specifications				
	vCPU	*Ram (GB)*	*Disk (GB)*	*BW (Gbps)*	*Price ($)*
Min	1	2	4	1	17.89
Max	192	4096	336000	400	37999
Types (n)	15	41	81	16	1497

The objective weights of the criteria were calculated by the Entropy method and the results are given in Table 4.

Table 4. The weights of the criteria

	Criteria				
	vCPU	*Ram*	*Disk*	*BW*	*Price*
W	0.1201	0.2272	0.3344	0.1074	0.2109

The objective weights obtained by the Entropy method were used to calculate the S_j and R_j values. Thus, Q_j value was calculated by using Eq. 7 with S_j and R_j values. The calculated S_j, R_j and Q_j values as a result were presented in Table 5.

Table 5. Ranking of alternatives according to Q_j

VIKOR S,R,Q values						
S_j	R_j	Q_j *(q=0)*	Q_j *(q=0, 25)*	Q_j *(q=0, 50)*	Q_j *(q=0,75)*	Q_j *(q=1)*
A_{257}	A_{257}	A_{257}	A_{257}	A_{257}	A_{257}	A_{257}
A_{960}	A_{688}	A_{688}	A_{960}	A_{960}	A_{960}	A_{960}
A_{688}	A_{960}	A_{960}	A_{688}	A_{688}	A_{688}	A_{688}
A_{2708}	A_{1695}	A_{1695}	A_{2708}	A_{2708}	A_{2708}	A_{2708}
A_{1695}	A_{1889}	A_{1889}	A_{1695}	A_{1695}	A_{1695}	A_{1695}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
A_{3276}	A_{50}	A_{50}	A_{50}	A_{50}	A_{3276}	A_{3276}
A_{3287}	A_{753}	A_{753}	A_{753}	A_{753}	A_{3287}	A_{3287}
A_{3288}	A_{2503}	A_{2503}	A_{2503}	A_{2503}	A_{3288}	A_{3288}

Table 5 shows the rankings that occur when the q parameter takes different values. The Q_j list obtained regarding the alternatives can be considered as the final decision list, but it is expected that the decision will be acceptable and stable by testing the conditions presented by the VIKOR method. Accordingly, the DQ value for acceptable advantage is calculated as $1/(m-1) = 1/3338 \cong 0.0003$. In addition, the rankings in Table 3 were compared for acceptable stability. For $q = 0.5$, it is seen that both the acceptable advantage and stability conditions are met. In this direction, the values obtained with $= 0.5$ consensus value were accepted as the solution. The results obtained show the resulting calculations without any constraints. The resulting product list summary is shown in Table 6.

Table 6. Summary of overall results

Alternatives					
A	*vCPU*	*Ram (GB)*	*Disk (GB)*	*BW (Gbps)*	*Price ($)*
A_{257}	1	2	160	**1**	32.14
A_{960}	1	2	160	1	32.14
A_{688}	1	2	160	1	34.33
A_{2708}	1	2	160	1	34.33
A_{1695}	1	2	160	1	42.37
⋮	⋮	⋮	⋮	⋮	⋮
A_{50}	48	192	336000	75	4608.34
A_{753}	48	192	336000	75	4608.34
A_{2503}	48	192	336000	75	5622.25

As seen in Table 6, A_{257}, A_{960} alternatives constitute the best choice in the first two rows. Although all the features of these products are the same, they are presented to the decision maker as two options due to the location difference. All the results according to the determined scenarios are shown in Table 7.

5. CONCLUSIONS

Within the scope of this study, it is aimed to present a model that will facilitate the selection of a product that can meet the needs of the decision maker parallel with the increasing diversity in virtual server product packages. Due to the timeliness of the problem and the increasing variety

of products, model studies on virtual server selection continue to increase day by day in the literature. In this study, the server configuration that can meet the expectations of the decision maker were examined with 5 different criteria and 7 different scenarios. The problem was considered as a Multi-Criteria Decision Making problem and the selection was made among 3339 alternative products. Criteria weights were calculated with Entropy, an objective weighting method. The results were calculated with the VIKOR method and the acceptable advantage, acceptable stability conditions of the results were tested. As a result, in the selection problem with no constraints, a cloud server product with 1 CPU, 2 GB of RAM, 160 GB of Hard Disk and 1 Gbps data transfer rate was recommended considering the criteria weights and Amazon product database data variation. In addition, according to the special scenario created for all criteria, the cloud server product with 16 processors (CPU), 64 GB of memory (RAM), 7500 GB of hard disk and 25 Gbps data transfer rate was selected.

Table 7. Scenario results

Alternatives						
Scenario	*A*	*vCPU*	*Ram (GB)*	*Disk (GB)*	*BW (Gbps)*	*Price ($)*
without constraints	A_{257} A_{960}	1	2	160	1	32,14
vCPU >= 8	A_{189} A_{717}	8	15	160	5	306,80
RAM >= 16 GB	A_{232} A_{857}	2	17	420	3	178,97
DISK >= 1000 GB	A_{150} A_{514}	4	15	1680	5	255,67
BW>= 20 Gbps	A_{67} A_{186} A_{371}	2	8	75	25	99,35
PRICE <= 100$	A_{20} A_{69}	1	4	4	3	48,94
with all constraints	A_4 A_8	16	64	7500	25	1063,00

The study differs from other studies in the literature in that 797040 different scenarios can be tested. On the other hand, the study cannot be called as a Decision Support System since the user interface design is not yet complete. In future studies, it is planned to design a web-based Decision Support System by completing the user interface design. Another limitation is the variety of alternatives considered in the study. Although the number of alternatives (product variety) is high, they consist of products of only one product provider that is Amazon EC2. In the planned web-based decision support system design, it is aimed to evaluate a high number of alternatives by increasing the variety of cloud service providers. Moreover, proposed model for instance selection from cloud service providers with fewer product options may not be sufficiently effective.

REFERENCES

[1] H. Cloud, **The nist definition of cloud computing**, National Institute of Science and Technology, Special Publication, *800*, 2011.

[2] L. Qing, Y. Jiang, and M. Yang, "Evaluating open iaas cloud platforms based upon nist cloud computing reference model**", 2014 17th IEEE International Conference on Computational Science and Engineering**, Chengdu, China, 1909-1914, December, 2014.

[3] J. Teixeira, C. E. Salgado, and R. J. Machado. ,"Modeling an IaaS broker based on two cloud computing reference models", **2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)**, Berlin, Germany, 166-171, April, 2016.

[4] W. Ming, Y. Jian, and R. Yongyi., "Dynamic instance provisioning strategy in an iaas cloud", **Proceedings of the IEEE 32nd Chinese Control Conference**, Xi'an, China, 6670-6675, July, 2013.

[5] N. Jain, and S. Choudhary, "Overview of virtualization in cloud computing", **IEEE 2016 Symposium on Colossal Data Analysis and Networking (CDAN)**, Indore, India, ,1-4, March, 2016.

[6] W. M. Ding, B. Ghansah, and Y. Wu, "Research on the virtualization technology in cloud computing environment", *International Journal of Engineering Research in Africa*, 21, 2016.

[7] L. Malhotra, D. Agarwal, and A. Jaiswal, "Virtualization in cloud computing", *J. Inform. Tech. Softw. Eng* 4.2, (1-3), 2014.

[8] A. Rashid, and A. Chaturvedi, "Virtualization and its role in cloud computing environment", *International Journal of Computer Sciences and Engineering*, 7.4, 1131-1136, 2019.

[9] J. Zhang, H. Huang, & X. Wang, "Resource provision algorithms in cloud computing: A survey", *Journal of network and computer applications*, *64*, 23-42, 2016.

[10] G. Manish, and S. Mulik, "An approach for selecting software-as-a-service (SaaS) product", **2009 IEEE International Conference on Cloud Computing**, Banglore, India, 155-158, September, 2009.

[11] Internet: Simple Montly Calculator, https://calculator.s3.amazonaws.com/index.html, 5.09.2022.

[12] Internet: Amazon EC2 On-Demand Prices, https://aws.amazon.com/tr/ec2/pricing/on-demand, 5.09.2022.

[13] M. O. Ogbole, E. A. Ogbole, and A. Olagesin, "Cloud Systems and Applications: A Review", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* ,142-149, 2021.

[14] A. Hussain, J. Chun, and M. Khan, "A novel framework towards viable cloud service selection as a service (cssaas) under a fuzzy environment", *Future Generation Computer Systems*, 104, 74-91, 2020.

[15] J. Dong, S. Wan, and S. Chen, "Fuzzy best-worst method based on triangular fuzzy numbers for multi-criteria decision-making", *Information Sciences*, 547, 1080-1104, 2021.

[16] Gerasimou, Simos, Radu Calinescu, and Giordano Tamburrelli, "Synthesis of probabilistic models for quality-of-service software engineering", *Automated Software Engineering*, 25.4, 785-831, 2018.

[17] Y. Shoukry, et al., "SMC: Satisfiability modulo convex optimization.", **International Conference on Hybrid Systems: Computation and Control**, 20, 19-28, 2017.

[18] M. H. Nejat, et al., "Efficient cloud service ranking based on uncertain user requirements", *Cluster Computing*, 25.1, 485-502, 2022.

[19] O. S. Vaidya, and S. Kumar, "Analytic hierarchy process: An overview of applications", *European Journal of operational research*, 169.1, 1-29, 2006.

[20] C. L. Belusso, et al., "Selecting services in the cloud: a decision support methodology focused on infrastructure-as-a-service context", *The Journal of Supercomputing*, 78.6, 7825-7860, 2022.

[21] P. A. Bommannavar, et al., "Analyzing Performance Parameters for Cloud Service Selection using AHP**", 2021 6th International Conference for Convergence in Technology (I2CT)**, Maharashtra, India, 1-7, April, 2021.

[22] D. Rai, and P. Kumar, "Instance based multi criteria decision model for cloud service selection using TOPSIS and VIKOR", *International Journal of Computer Engineering and Technology*, 7.1, 78-87, 2016.

[23] N. Tanoumand, et al., "Selecting cloud computing service provider with fuzzy AHP", **2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)**, Naples, Italy, 1-5, July, 2017.

[24] D. Y. Chang, "Applications of the extent analysis method on fuzzy AHP", *European Journal of Operational Research*, 95, 649-655, 1996.

[25] A. Abdelmuttlib I. Abdalla, et al., "Entropy-based fuzzy AHP model for trustworthy service provider selection in Internet of Things", **2018 IEEE 43rd Conference on Local Computer Networks (LCN)**, Chicago, IL, USA,606-613, October, 2018.

[26] S. Opricovic, & G. H. Tzeng, "Extended VIKOR method in comparison with outranking methods", *European journal of operational research*, *178*(2), 514-529, 2007.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

Systematic Literature Review: Improving IT Helpdesk Services

Melek BASMACI TOP[1], Aslıhan TÜFEKCI[2]

[1] Management Information Systems, Gazi University
[2] Graduate School of Informatics, Gazi University
melektop@gmail.com, asli@gazi.edu.tr

***Abstract*—** The aim of this study is to examine the studies on helpdesks, to reveal what kind of studies have been done to improve their services and to identify the deficiencies in this field in the literature. In the study, first of all, basic information about IT helpdesks will be given, and then the findings of systematic literature review will be included. Finally, the conclusions from the research and suggestions for future studies will be stated.

***Keywords*—** service desk, helpdesk, help desk, user support

1. INTRODUCTION

The purpose of this study is to conduct a systematic literature review and to shape the framework for improving IT helpdesk services through the studies reviewed to help understand the research trend in the IT helpdesk service field and identify research gaps.

2. IT HELPDESKS

Although IT helpdesks are named differently according to different sectors, fields or standards, their common task is to provide information and necessary support to the end user. Helpdesk, service desk or user support unit is the unit or system that provides end-user support, information and support about products or services [1]. The helpdesk handles service requests and incidents and provides timely appropriate assistance to those who need it. The purpose of the IT helpdesk is usually to troubleshoot IT problems and can also provide guidance on hardware and software. Helpdesks can provide support to users through various channels such as telephone, call system, e-mail or face to face. As the helpdesk is at the forefront of user services and thus becomes the face of the organization, improving the helpdesk staff and the infrastructures used is vital for productivity and overall user satisfaction [2].

3. METHOD

The systematic literature study method to be applied in the research was adapted from the theoretical study of Denyer and Tranfield (2009) [3]. Accordingly, the basic steps of the systematic literature approach are determination of research questions, selection of keywords and databases, selection criteria and selection of the sources, and search for answers to the research questions by examining the obtained sources in detail.

For the purpose of the research, four research questions were determined:

- Question 1: In which years were the studies on IT helpdesks made and published by which publishers?
- Question 2: In which areas/sectors have the IT helpdesks been studied?
- Question 3: What issues are focused on in the IT helpdesk studies?
- Question 4: What kind of work has been done on improving IT helpdesk services?

The keywords to be used in the research were chosen according to the research questions. The basic concepts that make up the scope of the research were determined as "help desk", "user support" and "service desk". The query used for the search is as follows: “service desk” or “helpdesk” or “help desk” or “user support”. By using the specified query, Middle East Technical University

(METU) Library's Integrated Search (METUnique Search) method was used, so that all databases that METU Library is a member of were searched. METUnique Search provides access to the entire METU library collection from a single entry point. Some of the databases accessible by this method are: ACM Digital Library, IEEEXplore, ScienceDirect, Emerald, Taylor & Francis Online, Springer.

In the METUnique Search, firstly, the query sentence was written directly without any criteria and the keywords were searched. A year filter was added between 2000-2022 in order to narrow the results and increase the focus on current studies, since the search gave very comprehensive results, and a title filter was added to the query in the next step. Especially using the title filter increased the focus on the research topic and made it easier to reach the articles related to the researched field. Then, within the scope of the source type filter, the article filter was selected in order to reach the research articles written about the research topic in the literature. This filter was chosen to focus on research in the field, and book and book reviews, standards, magazines and news were excluded.

Since there are many different types of the researched concept in the literature, it is aimed to filter the articles according to their topics and thus to find the articles closest to the research topic. In this context, topics that may be related to the research topic were selected according to the topics listed in the Integrated Search below, and the following numbers of articles were found from the selected topics, shown in the graph in Figure 1.

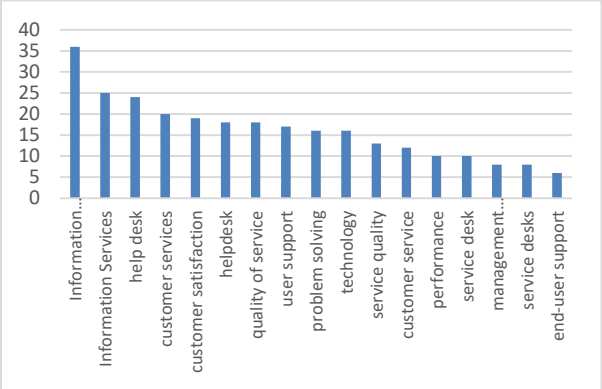


Figure 1. Distribution of studies by subject in source selection

The resources obtained after all filtering were exported from Integrated Search in BibTeX reference format, then imported into JabRef, an open source, cross-platform citation and reference management tool that uses BibTeX as the file format. 24 duplicate resources were removed using JabRef's features. After this control, the remaining 188 resources were included in the title and summary evaluation manually. With this evaluation, 84 sources written in different languages other than English and Turkish and which did not meet the target of the research questions or were not related to the research focus were excluded from the scope. After all the selections, the selected articles were re-examined in detail in order to answer the research questions. The number of articles obtained during the entire resource selection process is shown in Table 1.

Table 1. Number of articles reached step by step in source selection

Source Selection	Article Number
First query on keywords	62877
“2000-2022” year filter	60602
Title filter	9559
"Academic paper" resource type filter	882
Subject filter	212
JabRef repetitive article cleaning	188
Detailed article analysis	104

4. RESULTS

Selected sources were examined and analyzed according to the research questions. Results are presented below.

4.1. Distribution of Studies by Year and Publishers

Figure 2 shows the distribution of resources by years. While the number of studies and the popularity of the subject were higher in the early 2000s, the number of studies has varied over the years.

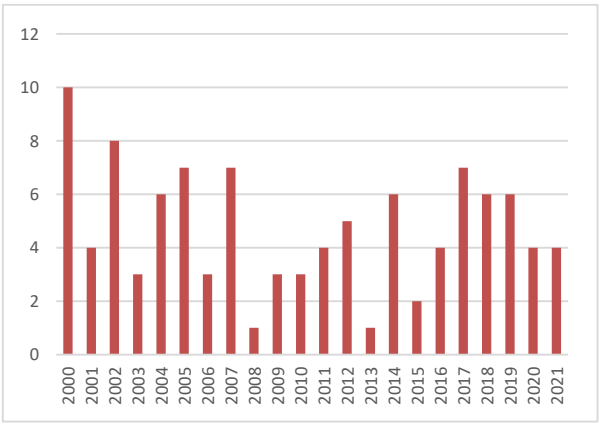


Figure 2. Distribution of selected studies by years

The distribution of publishers is shown in Figure 3. 12 publishers with only one publication in the selected sources are not indicated in the chart.

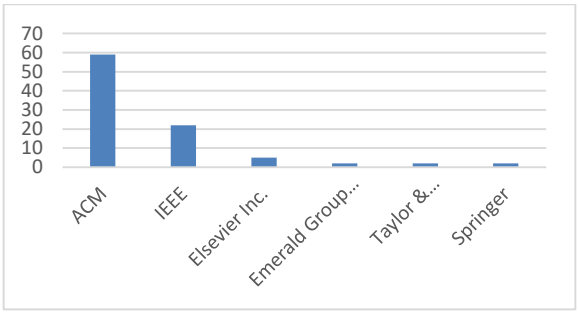


Figure 3. Distribution of selected studies by publishers

4.2. Distribution of Studies by Sector/Fields and Subjects

Figure 4 shows the distribution of the selected studies according to the sector they focus on.

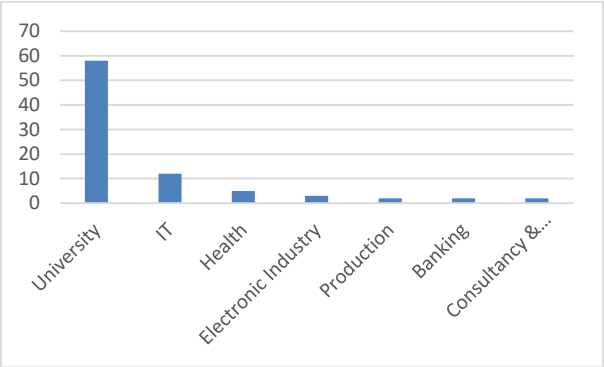


Figure 4. Distribution of selected studies by sectors/fields

When the selected studies were examined, it was possible to group them under certain topics. The most researched subject distributions are shown in Figure 5. It has been observed that researches on helpdesk working system development and helpdesk software are done more than other subjects.

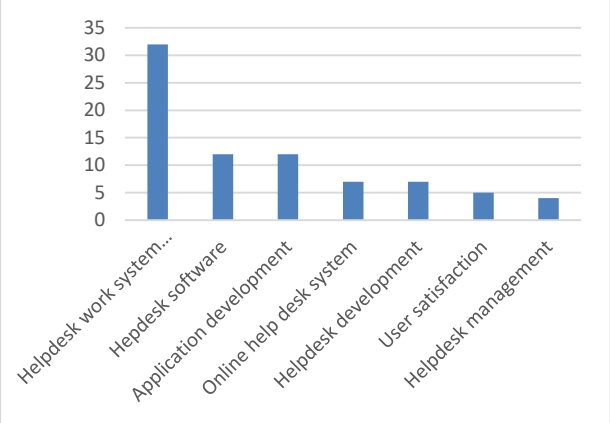


Figure 5. Distribution of selected studies by topic

The most used methods used in the studies examined are shown in Figure 6. Case studies were used as a method in most of the studies, followed by software/application, method and model developed studies.

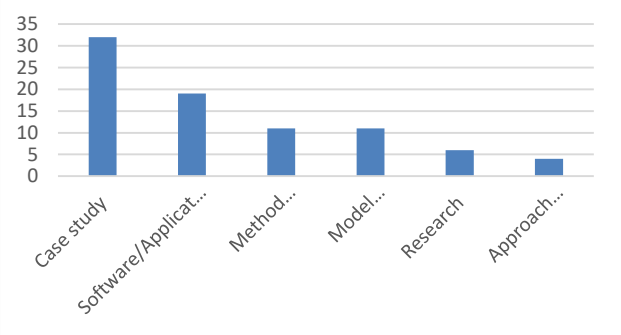


Figure 6. Distribution of selected studies by method

4.3. Studies on Improving IT Helpdesk Services

There are many studies on IT helpdesks in the literature. Different studies have been carried out to improve the services of helpdesks used in many different sectors. Methods have been produced and different tools, models or methods have been developed to improve the helpdesk working system, which is the most studied subject. The studies reviewed revealed that helpdesks are an area worth exploring because of the benefits derived from practical outputs.

While some of the researches in the literature describe what has been done on the establishment of helpdesks, some of them describe the projects produced to improve the existing helpdesks. These projects include establishing and centralizing helpdesks, developing online systems/software, developing/improving existing systems, implementing applications such as ticket systems, remote assistance, knowledge-based systems and unified helpdesks [4-19].

In some of the studies, open source systems developed in-house and the cost and advantages of being developed specifically for the institution are mentioned [4, 20]. The benefits and difficulties of the ready-made systems, which are not developed in-house and are mentioned in many stages from the selection process to user satisfaction, are also mentioned in the literature. It has been explained that such systems used can provide a more efficient operation in terms of employees and users by making IT operations and support process more structured and transparent, but they can also force organizations in terms of cost. In the case studies within this scope, the steps in the application life cycle such as defining the organization's requirements, product selection, commissioning and adoption are also explained [7, 21-24].

While the general aim of the studies is to increase service quality and user satisfaction, this issue has come to the fore in some studies. In these studies, the factors affecting user satisfaction were examined, and it was stated that when service quality measurements and user satisfaction measurements were combined, a richer understanding of end-user support would be achieved [1, 25, 26]. Communication at the helpdesk is one of the most critical issues, and the success of the process depends on open communication with all stakeholders. Listening to the user and asking the necessary questions until the problem is clarified and documented is one of the most important tasks that increase user satisfaction for all helpdesk personnel [27]. It is possible to increase overall satisfaction by establishing a positive, emotional bond with the user [28].

The increase in the complexity of the work in the field of IT requires a change and development from the past to the present in the working systems of such teams that support the users. It is necessary to create solutions to the difficulties encountered. These challenges include technical infrastructure inadequacies, limited resources,

lack of collaboration between teams, and change management challenges. Developing viable and creative solutions to such challenges will transform the average helpdesk into a great one [29-33].

When the help desk is managed consciously, a team or unit emerges that can provide invaluable opportunities for success to the organization [28]. Planning, choosing the right human resources, training new employees, creating a dynamic and fun workplace and team spirit, performance evaluation and low staff turnover will help the team to improve their skills and their responses to users [34-38]. It is also included in the literature that staff motivation and participation can be increased by optimizing the use and adherence to IT best practices recommended by ITIL [39].

Employees in worldwide helpdesk organizations typically use long-distance phone calls to provide user support, which is inefficient and costly, requires long service cycles, and results in poor service quality. For these reasons, Web-based helpdesk systems have emerged and have eliminated many of the associated disadvantages of traditional helpdesks [40, 41]. Integrating human assistants into web systems has also been shown to provide efficient user support [42]. Semantic helpdesk systems are also recommended to provide support to users over the web [43, 44]. Self-service IT or attempts to solve the user's technological problem without first resorting to the help of helpdesk personnel is an area that has been witnessed for some time and is open to improvement [45].

Gonzalez et al. (2001) hypothesized that information management systems would improve the operational performance of the call center [46]. The same hypothesis can be assumed for a service desk. A knowledge management system that collects organizational information will enable each staff member to benefit from the organization's knowledge and ultimately improve the overall service and performance of the service desk.

There are studies in the literature that provide guidelines for improving the services of help desks and designing online help systems for institutions [47]. Despite the wide variety of studies on help desks, there are few models or standard-based methodologies for problem detection and resolution in articles.

5. CONCLUSION

Helpdesks are critical in today's world as people need a point of contact for IT support. The goals of the help desks, in addition to being a single point of contact for all users, are to increase the satisfaction of the end users by shortening the problem resolution times and at the same time to provide quality service to the users.

In this study, a systematic literature review has been conducted to reveal the work done for the development of IT helpdesk services. The study resulted in 104 references and the sources were examined and analyzed in detail. The year, publisher, sector, methods and topics of the studies were determined and the findings were arranged to answer the research questions. In particular, it was seen that 58 of the studies were carried out in Universities, the studies in this field were common in universities and the results for direct application were in the majority. Although the number of studies conducted since the beginning of the 2000s has fluctuated over the years, the fact that the studies are still continuing shows that help desks are an area worth researching and interest in every period.

There have been many and various studies on helpdesks in the literature, but some issues have been addressed less than others. One of them is that helpdesks support the user through smart systems. Software such as Frequently Asked Questions, where users can find solutions on their own and search for solutions to their problems without resorting to the help of the helpdesk personnel at first, or "Chatbot", which provides information or performs a transaction by communicating with users through text or conversation, have been observed as areas open to research in this context. Studies can be also conducted on procedure-based, knowledge-based, semantic or case-based helpdesk systems, which are rarely encountered in the researched articles, to be used in the development of helpdesks.

In addition, it was observed that defined models or methodologies for problem detection and solution were rarely used in the studies examined. It can be suggested that management methodologies and strategies such as ITIL and COBIT should be used more widely in studies for the identification, measurement, analysis, improvement, control and best quality management of the problems at the help desks. Thus, it will be possible to identify the sources of the problems more accurately, to make appropriate developments and to determine management strategies. At the same time, a management approach to be applied at IT help desks can be developed by evaluating the findings in the study and applications can be made according to the structure of the organizations.

REFERENCES

[1] I. Pogarcic, S.R. Jankovic, & R. Seturidze, “How does the Help Desk Quality Improve customer satisfaction?”, *Athens Journal of Mass Media and Communications*, *3*(4), 343–362, 2017.

[2] Internet: C. Cilli, “Improving the Service Desk by Using COBIT 5”, https://www.isaca.org/resources/news-and-trends/industry-news/2018/improving-the-service-desk-by-using-cobit-5, 24.12.2022.

[3] D. Denyer, D. Tranfield, “Producing a Systematic Review”, **The Sage Handbook of Organizational Research Methods,** Editor: D. Buchanan, A. Bryman,, SAGE Publications Ltd, 1st ed., 671-688, 2009.

[4] C. Hitlin, C. Roy, D. Tallman, “Edesk online-Internal Development and deployment of Help Desk Management software”, **28th Annual ACM SIGUCCS Conference on User Services Building the Future - SIGUCCS '00**, 339-340, 29 Oct-1 Nov, 2000.

[5] J. Saul, B. Black, E. Larsson, “Helpdesk.drew.edu: Home Growing a Helpdesk Solution Using Open-Source Technology”, **28th Annual ACM SIGUCCS Conference on User Services Building the Future - SIGUCCS '00**, 289-293, 29 Oct-1 Nov, 2000.

[6] D. Orrick, J. Bauer, E. McDuffie, “Remedial help desk 101 at Florida State University”, **28th Annual ACM SIGUCCS Conference on User Services Building the Future - SIGUCCS '00**, 222-226, 29 Oct-1 Nov, 2000.

[7] J.V. Roberts, “Track-it!: Help Desk Automation”, **29th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '01**, 164-168, 17-20 Oct, 2000.

[8] R. Kirchmeyer, “The Consolidated Help Desk”, **30th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '02,** 184-185, 20-23 Nov, 2002.

[9] J. Gormly, “Rapid Help Desk Revitalization”, 3**1st Annual ACM SIGUCCS Conference on User Services - SIGUCCS '03**, 159-162, 21-24 Sept, 2003.

[10] B. Black, E.A. Larsson, “A Case Study: Implementing Supportworks Professional Helpdesk at Drew University”, **32nd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '04,** 26-29, 10-13 Oct, 2004.

[11] R. Davis, S. Maxwell, “The comprehensive it help desk: a single point-of-contact for the university community”, **32nd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '04,** 37-39, 10-13 Oct, 2004.

[12] E.Y. Borkowski, J. Elvove, C. Higgins,S. Kueppers, “Voluntary centralization of user support in a de-centralized organization”, **32nd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '04,** 327-331, 10-13 Oct, 2004.

[13] K. Evans, W.T. Jones, “Building an IT Help Desk: From Zero To Hero”, **33rd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '05,** 68-74, 6-9 Nov, 2005.

[14] A. Padeletti, B. Coltrane, R. Kline, “Customer service: help for the help desk”, **33rd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '05**, 299-304, 6-9 Nov, 2005.

[15] M.J. Conlon, “Overhaul your helpdesk ticketing system”, **35th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '07**, 7-10 Oct, 2007.

[16] R. Cottam, J. Goff, P. Nguyen, “Extending the centralized helpdesk functionality to improve decentralized support”, **ACM SIGUCCS 40th Annual Conference on Special Interest Group on University and College Computing Services - SIGUCCS '12**, 153-155, 15-19 Oct, 2012.

[17] E.-E. Jan, K.-Y. Chen, T. Ide, “A probabilistic concept annotation for IT service desk tickets”, **7th International Workshop on Exploiting Semantic Annotations in Information Retrieval - ESAIR '14**, 21-23, 7 Nov, 2014.

[18] R.K. Chapman, “Re-inventing the Helpdesk. Again. In five weeks or less”, **2015 ACM SIGUCCS Annual Conference**, 109-112, 9-13 Nov, 2015.

[19] O. Nikiforov, L.R. Mukhametova, “Key aspects of implementing the help desk system in an educational institution”, **III International Scientific and Practical Conference**, 19-20 March, 2020.

[20] C.J. Sinnett, T. Barr, “OSU Helpdesk: A cost-effective helpdesk solution for everyone”, **32nd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '04**, 209-216, 10-13 Oct, 2004.

[21] D. Bradstreet, M. Chast, K. Ziehl, “Ursula or Ariel? Is your help desk application evil or good in the eyes of your support staff”, **35th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '07**, 21-23, 7-10 Oct, 2007.

[22] D.R. Herrick, L. Metz, A. Crane, “Effective zero-cost help desk software”, **ACM SIGUCCS 40th Annual Conference on Special Interest Group on University and College Computing Services - SIGUCCS '12**, 157-160, 15-19 Oct, 2012.

[23] S.L. de Castro Silva, S.B. de Oliveira, A.A. Gonçalves, “Service Desk Software Usability Evaluation: The Case of Brazilian National Cancer Institute”, *Procedia Computer Science*, 100, 557–564, 2016.

[24] V. Krishnaswamy, M. Sharma, “Software-as-a-Service (SaaS) IT Helpdesk at an Institute of Higher Education: Implementation Issues”, *South Asian Journal of Management*, 25(4), 173-190, 2018.

[25] N.C. Shaw, W.H. DeLone, F. Niederman, “Sources of dissatisfaction in end-user support: an empirical study”, *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 33(2), 41–56, 2002.

[26] L.S. van Velsen, M.F. Steehouder, M.D.T. de Jong, “Evaluation of user support: Factors that affect user satisfaction with helpdesks and helplines” *IEEE Transactions on Professional Communication*, 50(3), 219–231, 2007.

[27] T. Chipman, “Creating and managing a helpdesk in a multicultural environment”, **35th Annual ACM SIGUCCS Conference on User -Services - SIGUCCS '07**, 30-33, 7-10 Oct, 2007.

[28] J. Pelletier, “One little spark: Putting the Service back in to Service Desk”, **2018 ACM SIGUCCS Annual Conference**, 31-34, 7-10 Oct, 2018.

[29] A. Cruess, “Transforming a help desk from average to excellent”, **30th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '02**, 229-231, 20-23 Nov, 2002.

[30] M. Link, “Transforming support: from helpdesk to information center”, **30th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '02**, 272-274, 20-23 Nov, 2002.

[31] S. Maxwell, R. Davis, “From soup to nuts the comprehensive it help desk”, **33rd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '05**, 211-213, 6-9 Nov, 2005.

[32] K. Kirchner, “Going from 0 to 60 in 20 years”, **33rd Annual ACM SIGUCCS Conference on User Services - SIGUCCS '05**, 132-137, 6-9 Nov, 2005.

[33] D.J., Kearney, J. Davidson, “Continuous change: A help desk motto”, **ACM SIGUCCS 40th Annual Conference on Special Interest Group on University and College Computing Services - SIGUCCS '12**, 115-119, 15-19 Oct, 2012.

[34] G. Sperl, “Taming the help desk”, **34th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '06**, 381-386, 5-8 Nov, 2006.

[35] K. Evans, G. Sperl, “The proper care and feeding of your help desk”, **35th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '07**, 82-86, 7-10 Oct, 2007.

[36] K. Wainwright, C. Power, “Adventures in starting a service desk”, **2016 ACM SIGUCCS Annual Conference**, 11-14, 6-9 Nov, 2016.

[37] C. Easom, “The Superheroes of Cyberspace: Help Desk Careers”, *Career Planning and Adult Development JOURNAL, 8-11, 2017.*

[38] C.L. Miller, J.C. Grooms, H. King, “To Infinity and beyond-gamifying IT service-desk training: A case study”, *Performance Improvement Quarterly*, 31(3), 249–268, 2018.

[39] F.S. Conceicao, A.P. Silva, A.Q. Filho, R.C. Filho, “Toward a gamification model to improve IT service management quality on service desk”, **2014 9th International Conference on the Quality of Information and Communications Technology,** 255-260, 2014.

[40] S. Foo, S.C. Hui, P.C. Leong, S. Liu, “An integrated help desk support for customer services over the World Wide Web - a case study”, *Computers in Industry*, 41(2), 129–145, 2000.

[41] S. Foo, S.C. Hui, P.C. Leong, “Web-based Intelligent Helpdesk-support environment”, *International Journal of Systems Science*, 33(6), 389–402, 2002.

[42] J. Aberg, N. Shahmehri, “An empirical study of human web assistants: Implications for User Support in Web Information Systems”, **SIGCHI Conference on Human Factors in Computing Systems - CHI '01**, 404-411, 31 March-4 April, 2001.

[43] T.T. Quan, S.C. Hui, A.C.M. Fong, “Automatic fuzzy ontology generation for semantic help-desk support”, *IEEE Transactions on Industrial Informatics*, 2(3), 155–164, 2006.

[44] D. Wang, T. Li, S. Zhu, Y. Gong, “IHelp: An Intelligent Online Helpdesk System” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1), 173–182, 2011.

[45] S. Zaza, I. Junglas, D.J. Armstrong, “Who needs the help desk? tackling one's own technological problem via Self It Service”, *Information and Organization*, 31(3), 1-19, 2021.

[46] L.M. Gonzalez, I. Gonzalez, R. Giachetti, G. Ramirez, “Analysis of Applying Knowledge Management in an Information Technology Call Center”, **Twelfth Annual Conference of the Production and Operations Management Society,** 30 March-2 April, 2001.

[47] Y. Chen, “To help helpdesk: A field study of online help systems in campus context”, *Campus-Wide Information Systems*, 21(2), 81–94, 2004.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

Digital Design and FPGA Implementation that Converts an Integer to a Floating Point Number

Esin MUTLU[1]*, Serkan DERELİ[1], Asiye Rümeysa ÜNSAL[2]

[1] Computer Engineering, Sakarya University of Applied Sciences, Sakarya, Turkey
[2] Electrical and Electronics Engineering, Sakarya University of Applied Sciences, Sakarya, Turkey
esinimutluuu@gmail.com, dereli@subu.edu.tr, asiyerumeysaunsal@gmail.com

Abstract— In this study, a design on floating point numbers, which are frequently used in application-specific integrated circuit (ASIC) systems, has been carried out. In digital designs, it is seen that sometimes integers are used in the initial stage, but then the transition to floating-point numbers is made. Especially considering that floating-point numbers are hardware costly, operations are performed with integers as much as possible. However, when integers need to be converted to floating-point numbers, the conversion must be done quickly and the process must continue. In this study, integers are first converted to 32-bit and then converted to 32-bit floating point numbers. The study is analyzed at the point of hardware cost, time cost and conversion to semiconductor level production.

Keywords— FPGA design, embedded system, IP-Core, floating-point, exponential

1. INTRODUCTION

The initial challenges related to FPGAs programmability and large interconnection capacitances (poor performance, low logic density and high power dissipation) have been overcome while providing attractive low cost and flexibility [1].

The use of FPGAs in the implementation of complex applications is increasingly common but relatively new when dealing with floating-point applications ranging from scientific computing to financial or physics simulations [2,3,4]. This is a field of increasing research activity due to the performance and efficiency that FPGAs can achieve. The peak FPGA floating-point performance is growing significantly faster than the CPU counterpart [5] while their energy efficiency outperforms CPUs or GPUs [6]. Additionally, FPGA flexibility and inherent fine-grain parallelism make them ideal candidates for hardware acceleration improving CPUs capabilities for a particular set of problems with complex datapaths or control and data inter-dependencies [7]. FPGAs flexibility also allows the use of tailored precision, what can significantly improve certain applications. Furthermore, new FPGA architectures have embedded resources which can simplify the implementation of floating-point operators.

Exponential numbers, which are frequently used in solving engineering problems, have a very complex structure in digital design, although they can easily produce results in today's computers. Especially the mathematical operations performed with floating point numbers tire digital systems quite a lot. For this reason, besides ALU, there is also a unit for processing integer numbers in processors. Floating-point numbers were first adopted as the industry standard in 1985 then they were revised in 2008 and standardized as IEEE 754-2008 [8]. Thus, until today, due to its strong representation ability, it has been widely preferred in computer systems with 32-bit length single precision and 64-bit length double precision [9]. There are two possibilities for integer numbers especially for those using FPGA technology. One of them is floating point numbers while the other are fixed numbers. When the literature is examined in detail, we know that both of them are widely used in many studies [10, 11, 12].

The main reason of designing this conversion operator is almost every system uses hybrid data types, therefore it is significant to convert while transferring different type of

data to sub-systems. Especially, in a design that do operates with integer numbers, when it is necessary to use floating point numbers, created design in this study can be easily used by the researches those who works on this subject. As we can see on the Table-1, there is no supported conversion operator for integer to floating point in the library of Xilinx IP CORE [13].

Table 1. IP Core list supported by Xilinx company

IP CORE	Supported	IP CORE	Supported
Multiply	Yes	Natural logarithm	Yes
Add/subtract	Yes	Exponential	Yes
Accumulator	Yes	Conversion from floating-point to fixed-point	Yes
Divide	Yes	Conversion from fixed-point to floating-point	Yes
Square-root	Yes	Conversion between floating-point types	Yes
Comparison	Yes	Conversion between floating-point types	Yes
Absolute value	Yes	Conversion from integer to floating-point	No

2. MATERIALS AND METHOD

2.1. Floating Point Conversion and IEEE 754 Standard

A floating point number, is a positive or negative whole number with a decimal point. For example; 5.5, 0.25, and -103.342 are all floating point numbers, while 91, and 0 are not. Floating point numbers get their name from the way the decimal point can "float" to any position necessary. Due to this, in computer science, floating point numbers are often referred to as floats. Other common types of numbers in computer science are integers, short, and long. Floating point numbers are used to represent non-integer fractional numbers and are used in most engineering and technical calculations, for example; 3.256, 2.1, and 0.0036. The most commonly used floating point standard is the IEEE standard. According to this standard, floating point numbers are represented with 32 bits (single precision) or 64 bits (double precision).

The IEEE 754 single precision format is a 32-bit format. This format uses 1-bit for sign, 8-bits for exponent and 23-bits to represent the fraction as in fig. 1. The single precision floating-point number is calculated as $(-1)^S \times 1.F \times 2^{(E-127)}$. The sign bit is 0 for non-negative number and 1 for negative numbers. The exponent field can be used to represent both positive and negative exponents. To do this, a bias is added to the actual exponent. For IEEE single-precision format, this value is 127. As an example, a stored value of 147 indicates an exponent of (147-127), or 20. The mantissa (significand) is composed of an implicit leading bit and the fraction bit, and represents the precision bits of the number. Exponent values of 0xFF and 0x00 are reserved for special numbers such as zero, de-normalized numbers, infinity, and Not a Number (NaN) [14].

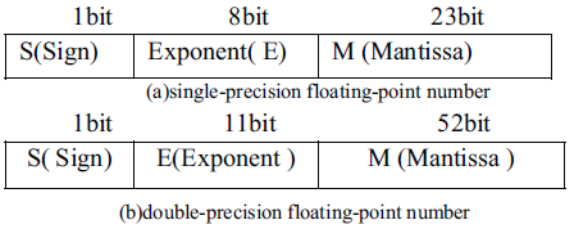


Figure 1. Floating-Point Standard

Floating point arithmetic is widely used in many areas. Multipliers play an important role in today's digital signal processing and various other applications. The way floating point operations are executed depends on the data format of the operands. IEEE standards specify a set of floating point formats with single precision and double precision. When we look at the literature, the studies on this subject are as follows;

In the study by Hilal Kaptan, Ali Tangel, and Suhap Sahin in 2021, it is shown that FFT algorithms using floating point numbers can be implemented on an FPGA. This includes addition, subtraction, multiplication, and division modules based on the 32-bit single precision IEEE 754 format [15].

In the study by Yuheng Yang, Qing Yuan, and Jian Liu in 2021, they found that they used an efficient architecture of floating-point square-root circuit with low area cost, which is in accordance with the IEEE-754 standard. We extend the principle of the standard SRT algorithm so that the latency and area cost of the proposed circuit are linear with the radix [16].

In the study by Sneha S. Jumle and M. V. Vyawhare in 2015, they present an efficient FPGA implementation of a single-precision floating point multiplier that supports the IEEE 754-2008 binary interchange format. The proposed multiplier doesn't implement rounding and presents the significant multiplication result as is (48 bits) [17].

In this study by Serkan Dereli;Mahmut Uç in 2020, a digital circuit design that computes both the integer and a floating point exponent of a 32-bit floating-point number has been realized[18].

2.2. Field Programmable Gate Arrays (FPGA)

Field Programmable Gate Arrays (FPGA) are increasingly being used to construct high-end, computationally intensive microprocessors capable of handling both fixed and floating-point mathematical computations. In terms of usage, FPGAs stand out as an empty device and they have to realize all the designs they will use in every designer application. For example, ready-made functions such as mathematical functions, analog-digital conversions found in many embedded devices (raspberry, PIC, Arm) are not available in FPGAs. This is one of the major disadvantages

of these devices and a major obstacle for first time users [19].

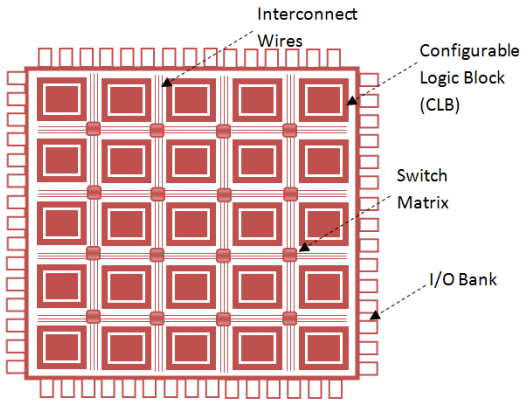


Figure 2. Architecture of FPGA

Figure 3 shows the structure of a basic FPGA and their interconnection. As it seems, an FPGA consists of three important units: CLB, IOB, SM.CLBs are the most basic unit and the designs performed are found here. It is named as configurable logical block. As can be seen from Figure 3, the more CLBs there are in an FPGA, the larger the circuits can be accommodated [20]. Already, it is the CLBs that make FPGAs stand out from other embedded systems, enabling the transactions to be hardware-based and ultra-fast. Although IOBs (Input/output Block) vary from application to application, they can also be used as output pins as input pins. These pins each of which is 1-bit, come together to form data groups in the form of 4, 8, 16, 32 bits [21].

SMs actually carry out the function of managing and routing the paths that provide the connection between input / output pins and logical blocks (CLB). Because, if a design is too large, it is placed in more than one logical block (CLB) and the communication of these logical blocks with each other is completely carried out by SMs [22].

The main difference between an FPGA and others is the level of customization and complexity. They also vary in price and ease of use. Essentially, an FPGA allows for greater customization and more complex processes, as well as retroactive changes to hardware. Table 2 contains the subject of FPGA used in some studies in the literature. Even from these studies, it can be seen that FPGAs are actually ideal for performance and real-time systems.

Table 2. Studies of some literatüre

STUDY	YEAR	SUBJECT
[18]	2015	The Objective of this to implement the 32 bit binary floating point adder with minimun time.
[19]	2017	They describe their implementation of IEEE 754 compliant single precision floatingpo adder that supports denormal inputs. Further, they compare its performance and reso utilization against the Xilinx floating-point adder IP core.
[20]	2017	They evaluates the accuracy of a fixed-point LU decomposition based on FPGA. Fixe point architecture of LU decomposition is implemented on FPGA using random matri and different word-lengths, descriptive analysis of error is performed.
[21]	2006	The single-precision floating-point IEEE-754 standard Adder/Subtractor and Multipli modules with high speed and area efficient are presented.
[22]	2021	They propose an efficient architecture of floating-point square-root circuit with low a cost, which is in accordance with the IEEE-754 standard.

2.3. Designed Proposed

First we get the integer number that exists. Then the marking is done, this part represents the leftmost 1-bit of our number, then we try to determine the number 1 that we left on the left by default. The value of the number of shifts we make here also helps us find the exponent value. When we get this value, the sum is performed with the bisa value. Finally, we place the values we found in the format we have and create the format.

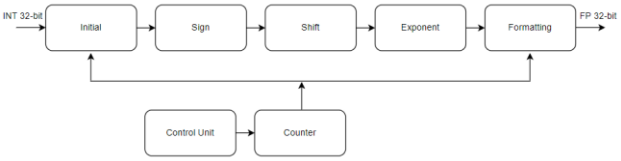


Figure 3. Finite state machine flowchart used in the study

In the numerical design performed in this study, the flowchart shown in Figure-3 was followed serially. In these five stages, the running times of the states were also realized by means of a control unit. The detailed description of these finite state machines is shown in Figure-4.

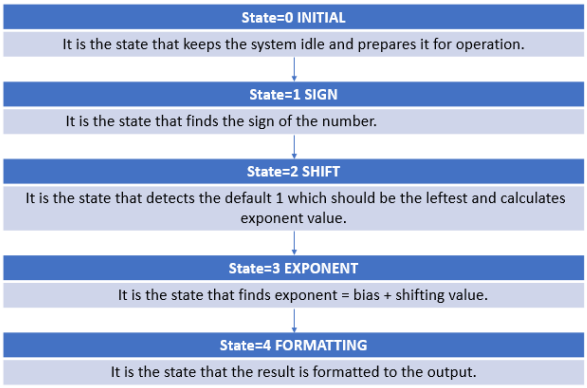


Figure 4. Finite state machine workflow used in the study

The representation of some sample numbers used for testing purposes in the design that converts a 32-bit integer to a 32-bit floating point number developed within the scope of this study is given in Table-3. In this table, the integer that comes to the system entry is first rounded up to 32 bits. For this, if the number is positive, the MSB side is filled with zeros to complete 32-bit. otherwise, the number is negative and using the 2's complement method, the positive value of the number is obtained and this value is padded with zero, resulting in a 32-bit conversion.

Figure-5 shows the step-by-step conversion of an integer to a floating point number according to the finite state diagram shown in Figure-4. In this figure, the value on the left represents the input to the finite state machine, and the value on the right represents the output of the finite state machine.

Tablo 3. Conversion of some integer numbers to 32-bit floating point numbers

Decimal	Input → Integer (Binary)	Output → Floating Point (Binary)
5	0101	01000000101000000000000000000000
-5	1111111111111011	11000000101000000000000000000000
2640	101001010000	01000101001001010000000000000000
-2640	1111010110110000	11000101001001010000000000000000
4640320	10001101100111001000000	01001010100011011001110010000000
-4640320	11111111101110010011000111000000	11001010100011011001110010000000

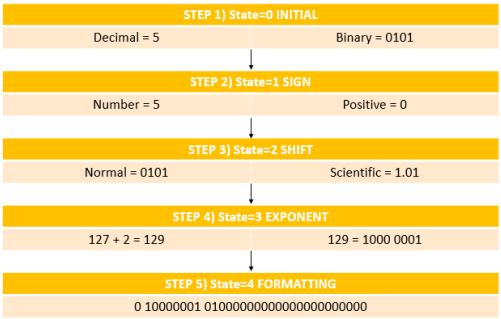


Figure 5. The conversion journey of an integer to a floating point number according to the FSM

3. RESULTS

As seen in the results of this study, the result for the number 5 and our result in the table match. The time required for this process to occur is approximately 36 clock pulses. Here, the conversion process from integer to floating point, which is not available in IP Core, has been realized. The most important condition to be able to use this work is that the integer coming to the system input is 32-bit length. In the simulation screen shown in Figure-6, the "g" signal represents the input integer value, and the "o" signal represents the output floating point number, and both signals are 32-bit in length.

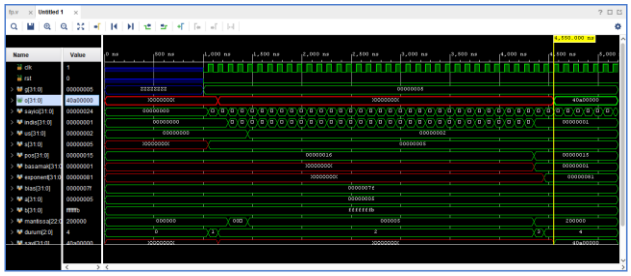


Figure 6. Simulation steps of the number "5"

As seen in our simulation outputs (Figure-6), the flexible and fast structure of FPGA was used and the correct result was obtained in the expected time interval for this process. In addition, the design realized in this study is in a synthesizable structure that can be used as IP-CORE in ASIC designs, and the hardware cost is shown in Table-4.

Tablo 4. Cost of design

Cell Number	I/O Port Number	Net Number	LUT Number	FF Number
625	66	847	255	194

Table-3 shows the cost of the design stages in terms of time. As it is clearly seen in this table, the shifting step of each bit is directly proportional to the precision of the floating point number.

Tablo 5. FSM cycle count in design

ORDER	FINITE STATE MACHINE	COUNT
1	INITIAL	1
2	SIGN	1
3	SHIFT	32
4	EXPONENT	1
5	FORMATTING	1
	TOTAL	36

CONCLUSION

In this study, the conversion of 32-bit integers to 32-bit floating point numbers has been performed. In the conversion process, first of all, the integer that will enter this process is completed to 32-bit in terms of the number of bits. This situation can be converted to a single precision floating point number by means of the system here, by normalizing those that have been somehow converted to integers from other number types in any system to 32-bit width. Since the work is designed in a synthesizable structure, it can be used in any work that will turn into semiconductor level production.

REFERENCES

[1] I. Kuon, J. Rose, "Measuring the gap between FPGAs and ASICs, computer-aided design of integrated circuits and systems", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26 (2), 203-215, 2007.

[2] R. Scrofano, M. Gokhale, F. Trouw, V. Prasanna, "Accelerating molecular Dynamics simulations with reconfigurable computers, Parallel and Distributed Systems", *IEEE Transactions on Parallel and Distributed Systems*, 19 (6), 764-778, 2008.

[3] G. Zhang, P. Leong, C. Ho, K. Tsoi, C. Cheung, D.-U. Lee, R. Cheung, W. Luk, "Reconfigurable acceleration for Monte Carlo based financial simulation", **IEEE International Conference on Field-Programmable Technology**, 215-222, 2005. doi:10.1109/FPT.2005.1568549.

[4] L. Zhuo, V. Prasanna, "High-performance designs for linear algebra operationson reconfigurable hardware, Computers, IEEE Transactions on Computers 57 (8) (2008) 1057-1071.

[5] K. Underwood, FPGAs vs. CPUs: trends in peak floating-point performance, in: FPGA '04: Proceedings of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays, ACM, New York, NY, USA, 2004, pp. 171- 180. doi:http://doi.acm.org/10.1145/968280.968305.

[6] A. George, H. Lam, G. Stitt, Novo-g: At the forefront of scalable reconfigurable supercomputing, Computing in Science Engineering 13 (1) (2011) 82-86, doi:10.1109/MCSE.2011.11.

[7] S. Che, J. Li, J.W. Sheaffer, K. Skadron, J. Lach, Accelerating compute-intensive applications with gpus and fpgas, Application Specific Processors, Symposium on (2008) 101-107. doi:http://doi.ieeecomputersociety.org/ 10.1109/SASP.2008.4570793.

[8] M.A. Çavuşlu, C. Karakuzu, S. Şahin, F. Karakaya, "Yapay sinir ağı eğitiminin IEEE 754 kayan noktalı sayı formatı ile FPGA tabanlı gerçeklenmesi" **İstanbul Teknik Üniversitesi (GOMSİS)**, 2008.

[9] L. Kamm, J. Willemson, "Secure floating point arithmetic and private satellite collision analysis" *Int. J. Inf. Secur*, 14:531-548, 2015.

[10] N.J. Higham, S. Pranesh, "Simulating low precision floating-point arithmetic", *SIAM Journal on Scientific Computing*. 41:585-602, 2019.

[11] M. Brain, C. Tinelli, P. Rümmer, T. Wahl, "An automatable formal semantics for IEEE-754 floating-point arithmetic", **IEEE 22nd Symposium on Computer Arithmetic**, 160-167, 2015.

[12] O. Catrina, "Round-efficient protocols for secure multiparty fixed-point arithmetic", **International Conference on Communications (COMM)** 2018; 431-436.

[13] Xilinx, LogiCORE IP Product Guide, Floating-Point, Operator v7.1, PG060 December 16, 2020.

[14] M.F. Tolba, A.M. AbdelAty, N.S. Soliman, L.A. Said, A.H. Madian, A.T. Azar, A.G. Radwan, "FPGA implementation of two fractional order chaotic systems", *AEU-International Journal of Electronics and Communications*, 78:162-172, 2017.

[15] H. Kaptan, A. Tangel, S. Sahin, "Fpga Implementation of Fft Algorithms Using Floating Point Numbers", 2021.

[16] Y. Yang, Q. Yuan, and J. Liu, "A Low-Cost High Radix Floating-Point Square-Root Circuit", 2021.

[17] S. S. Jumle and M. V. Vyawhare, "Development of an FPGA based high speed single precision floating point multiplier", 2015.

[18] S. Dereli, M. Uç, "Exponential Computing Digital Circuit Design Developed for FPGA-based Embedded Systems", 2020.

[19] M.A. Çavuşlu, C. Karakuzu, S. Şahin, F. Karakaya, "Yapay sinir ağı eğitiminin IEEE 754 kayan noktalı sayı formatı ile FPGA tabanlı gerçeklenmesi", **İstanbul Teknik Üniversitesi (GOMSİS)**, 2008.

[20] H. Abdelkrim, S.B. Othman, S.B. Saoud Reconfigurable SoC FPGA based: Overview and trends. **IEEE International Conference on Advanced Systems and Electric Technologies (IC_ASET)**, 2017.

[21] S. Dereli, "Micro-sized parallel system design proposal for the solution of robotics based engineering problem", *Microsystem Technologies*, 27(12), 4217-4226, 2021.

[22] S. Dereli, "Yüksek Hızlı FPGA ile Yeni Bir LFSR Tabanlı 32-Bit Kayan Noktalı Rastgele Sayı Üreteci Tasarımı", *International Journal of Advances in Engineering and Pure Sciences*, 32:219-228, 2020.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

Color-Based Image Segmentation Methods and Applications: A Review of the Last Decade

Tesnim GÜLSEN [1] *, Serkan DERELİ [2]

[1] Electrical Electronics Engineering, Sakarya University of Applied Sciences, Sakarya, Turkey
[2] Computer Technology, Sakarya University of Applied Sciences, Sakarya, Turkey
tesnim.gulsen@gmail.com, dereli@subu.edu.tr

***Abstract*—** Image segmentation is one of the most essential and sophisticated stages for object detection in image processing. Researchers have presented many techniques to reach the qualified result that is both computationally efficient and high accuracy over the years. Although, these techniques have implemented grayscale images firstly, then enhanced in color images to provide more valuable pieces of information by developing technical requirements. This study, have been evaluated color image segmentation techniques with their applications in various fields over the last few years. The major part of the paper, provides significant achieved success in techniques that are assumed traditionally. Also, have been explained the most commonly used color spaces used to represent colors. Finally, shown a table that contains the pros and cons of image segmentation techniques to compare briefly.

***Keywords*—** Color spaces, Image segmentation, Thresholding-based segmentation, Region-based segmentation, Clustering-based segmentation, Edge-based segmentation

1. INTRODUCTION

Segmentation is one of the most critical and complex operations encountered in image processing. The importance of the segmentation stage is that seriously affects the result of operations such as feature extraction or object detection [1]. Segmentation is basically the process of clustering pixels in the image together with similar characteristics. In this way, homogeneous regions are created in the image, to facilitate the analysis of the image and the detection of objects. Ultimately, according to some properties (*like density, texture, edge, or color*), obtained K number of homogeneous regions that contain N pixels in the image. The homogeneity of the regions is achieved by the property of the color channel in color images, naturally [2]. Color-based segmentation, as it provides features that are directly related to the object in the image; stands out with its various applications in many different fields, including in agriculture [3] for the detection of diseases in products, in medicine [4] with the example of pathology extraction from endoscopic images and in the industry [5] with its studies on the separation of objects on the production lines. Although, color space has a more complete and reliable aspect of image segmentation, it has been more possible for researchers to work on gray images in terms of both hardware requirements and computation times [6]. For this reason, various techniques have been developed on gray-scale images in the past three decades. However, with the increasing processing capabilities of the hardware used today and the decrease in computation time, a remarkable rise is observed in the algorithms which work on color images [7]. These algorithms and models have been presented as supervised and unsupervised methods. Although, supervised methods are at the forefront of the applied ones, has emerged in need of very large data sets for the process. Moreover, neuroscientists have demonstrated that the visual system in living things effectively processes and perceives images unattended [8].

Unsupervised methods work on the classification of pixels in the image without any training process, and it is important for real-time applications. Unsupervised methods can be grouped into thresholding-based methods, edge-based methods, clustering-based methods, region-based methods, and hybrid methods [9]. Although, none of the developed approaches is superior to the other, the most appropriate algorithm varies depends not only on the image but also the application [10]. It should also be emphasized that since the image segmentation problem is mainly related to psycho-physical perception, it can be not

possible to solve completely with any mathematical model [11]. In this study, the methods suggested in the literature in terms of the segmentation of color images have been explained. The next part of the study is organized as follows: In the second part, there are short pieces of information about the color spaces with their characteristics that are frequently used in color image processing. The third part contains information about segmentation methods and current applications in color images. The fourth part is the conclusion section of this paper.

2. COLOR SPACES

There are several color spaces to define the color and deciding a color space belongs to its characteristics that bring advantages to operation. But none of them can surpass the others for all types of color images. Deciding the color space which will be used to process is one of the challenges in color image segmentation due to each color representation having its pros and cons. For instance, the RGB space is useful for color display, it is largely utilized in photography, animation, and the film industry [12]. One other color space HSV is useful for face detection and color spaces. CIELab and CIELuv are very impressive for image segmentation [13]. In the literature, there are frequently used color spaces like that RGB (*red, green, blue*), HSV (*hue, saturation, value*), HSI (*hue, saturation, intensity*), CIEL*a*b, CIEL*u*v, YUV and YCbCr [12].

2.1. RGB Space

RGB color space represents the image as an MxN by tri chromatic formed that contain values of the red, green, and blue color channels. Representation of the RGB is standardized by performing color matching tests using the primary colors of red (*700 nm*), green (*546 nm*), and blue (*435 nm*) by the Commission Internationale d'Eclairage (*CIE*) in the 1930s [14]. RGB color space is fundamentally used on digital platforms; computer graphics, scanners and image storage. The RGB space's shape is an ordinary cube that coordinates account for the three primary colors; red, green and blue. Each channel value is in the range [0, 255]. The vertices value of [0, 0, 0] and [1, 1, 1] are special because they represent white (*the three main colors mixed up together in full amount*) and black (*none of any of the three basic colors*) [15].

2.2. CIE HSV Space

The HSV space can separate the color information from the brightness and represents the color with the composition of hue (h), saturation (s) and value (v). Hue the purity of the color in the range of [0-360] as a chromatic feature, saturation describes the measurement of the purity of the color. Value controls the brightness of the color; ranging from black at the bottom with lightness, to white at the top with lightness [16]. The value and the saturation are both placed on a scale of 0 to 100 percent. If brightness is zero, pure black is visible while a color with 100 percent brightness has no amount of black mixed into the color. While unsaturated colors are gray, the fully saturated form of color means that color has no white component [17].

2.3. CIE HSI Space

HSI is another hue-based color space similar to the HSV for hue and saturation parameters. Each pixel has three pieces of information: hue and saturation which come from color, and intensity which describes the brightness in HSV. It is knowns similarity to the human visual system and for this reason frequently used in visible-light image processing [18].

2.4. CIE Lab Space

The schematic diagram of CIELab space is very similar to the RGB, however, the coordinates of color are different. L* symbolizes the intensity, a* and b* values come from the color information. Part of from +a to -a corresponds from red color to green, and b signifies between the yellow (+b) and blue (-b) axis [12]. CIELab space can deal with lighting changes due to having information from the lightness of the color. One of the reasons to surpass other spaces is that Lab has every shade of every color that humans can see or not [19].

2.5. CIE Luv Space

In the CIELuv space, color distance can be evaluated linearly like the CIELab, distance between two points approximately gives how to change the color information with luminance, chroma, and hue. Parameters differ from CIELab and have another range in coordinates. Generally, u* and v* change in the [-175,175] range and the L* varies from 0 to 100 [12].

3. METHODS AND APPLICATIONS

3.1. Supervised Techniques

With the improvement of machine learning techniques, effective systems that are learning-based have been proposed after rule-based methods used frequently [20]. After the learning, the original algorithm can adapt to different images, however, the development is limited by the essential limitation of the algorithm. One of the key points that limits researchers is selecting a fit algorithm to implement in the images. The guidance of a human is often needed to choose the algorithm to be used also in most practical image segmentation problems. In addition, there is no accepted comprehensive method either algorithm or mathematical formulation for image segmentation that works automatically [21]. Lately, even though there are great improvements in supervised learning using deep learning methods in image segmentation like CNN, RNN [22], and U-net [23], unsupervised methods seem a difficult task. However, these methods are still an up-and-coming research area due to no need for a huge variety of labeled samples. Wang and others [24], explained a new

method by combining the mean clustering algorithm and SVM. Firstly, the clustering algorithm is used to get samples of training, then they extracted the features of color and the texture with mean-clustering, and then trained the images by SVM. Hajabdollahi and his friends, for reducing the network complexity in feature extracting, proposed a pruning framework that works selection of the color channels that gives more information in the dermoscopy images. In this way, CNN with the chosen channel has %8 better accuracy than without a channel selection stage [25]. Libouga and Laboratory of Parasitology & Ecology team members, despite the input image condition that has to be a gray-scale in U-Net, to take advantage of color information have used the three color components of the microscopic image as input image [26]. And the network shows, a performance of 99.8% accuracy for segmentation of the 4 types of intestinal parasites from the background of the color microscopic image. Mohammadi and others have proposed a composing technique for crops and weeds segmentation in color images. After the thresholding, segmentation is applied by using U-Net then crops and weeds are subtracted by following K-Means clustering in the algorithm [27]. When the composed method compares with K-Means clustering and superpixels algorithms, the maximum accuracy of equivalent to 99.19%.

3.2. Unsupervised Techniques

3.2.1. Thresholding-Based Techniques

Thresholding algorithms can be explained like this briefly; when an image is separated into two classes according to a specified threshold value, it is called bi-thresholding, however, multiple thresholding values are determined and the image is split into many various parts, it is called multilevel thresholding. In addition, given color image has several color components; there is a tough process that implements multilevel thresholding to segment this image. Finding fit thresholding values (*when implementing multilevel thresholding segmentation to the image*) can be called an optimization problem. Over the years, by coping with the computational challenges, swarm intelligence (SI) algorithms have been used for multilevel thresholding segmentation to find proper thresholding values. Nobuyuki Otsu's method [28], Kapur's entropy-based approach [29] and Tsallis's moment-preserving entropy principle [30] are well-known thresholding techniques that are joined with several swarm intelligence algorithms that can be used to multilevel thresholding segmentation over the years. He and Huang presented an Impressive Krill Herd (EKH) algorithm to overcome the problem of finding optimal thresholding values and Otsu's, Kapur's and Tsallis's methods were used as objective functions in that study [31]. Dutta, Talukdar and Bora, used color vegetative indices and Otsu's thresholding on aerial RGB images for the segmentation of unhealthy leaves to support an early disease detection system [32]. Xing, has worked with the Improved Emperor Penguin Optimization (IEPO) algorithm developed from the EPO algorithm by using three strategic stages. And then, has shared its experimental results that IEPO has an excellent effect than other color image segmentation methods as higher accuracy, and less CPU time for satellite images that have complex backgrounds and plant canopy images [33]. Baz and her team have changed the red threshold for segmentation in HSV and RGB spaces by combining the two segmentations. Results show this study improved the precision of the red segmentation to detect traffic signs in a different sample of illumination and weather cases (*foggy, sunny, and rainy*) [34]. Yang and al. proposed a study using the color space of YCbCr with the fixed threshold that has more information by Cb and Cr to deal with the problem of image segmentation in the underground cable equipment that is affected by the intensity of the illumination [35].

3.2.2. Clustering-Based Techniques

Clustering-based image segmentation methods use the classified pixels to generate classes of prototypes as similar as possible. Some of them use the neighborhood of pixels that used the local features of the image for segmentation, while others process the general features of the image. The K-means algorithm which is used widely due to being understandable easily and implementable simply uses the entire image features in the segmentation process. The working principle of K-means is based on randomly assigning the central points the according to the pre-assigned centroids, which represent a predefined number of clusters K with the help of a decider, and then using a predetermined metric to gradually reduce the error. The metric (*the Euclidean, the City Block, the Mahalanobis, etc.*) works on minimizing the distance between the data points and the central point in the same cluster and then maximizing the distance between centroids of all clusters [36]. In 1973 [37], J.C. Dunn developed Fuzzy C-Means, another well-known algorithm in the literature, by working on a new one of fuzzy clustering methods (*also referred to as soft clustering*). Fuzzy clustering differs in that each point can take place in more than one cluster, from non-fuzzy methods by assigning a membership matrix (μf) to every pixel in the image. The method proposed in [38] separates extracted mean feature vectors from RGB images into homogeneous regions with a Bayesian framework and creates adaptive trees and hierarchical clusters. The proposed ABT (*Adaptive Bayesian Trees*) approach that expanded the form of the Mean Shift has reached the value of 0.8148 in the Rand index when employed in the Berkeley segmentation dataset. Zhi Li *et al.* proposed a variational model based on a convex K-means approach with a smoothing and thresholding (SaT) strategy for color image segmentation [39]. And they tested this novel method on six real-world images to show comparable segmentation results to the Convex K-means Clustering (CKC) without complex calculation. Results show that CKC takes the lead in many images in terms of segmentation accuracy, however, it gets behind the FRC (*Fuzzy Region Competition*) framework in terms of CPU time-in. A 3-step segmentation for color micro-tool images is proposed in [40] for the observation of wear detection of the micro-milling tool, which is frequently needed in the

industry. Fuzzy C-Means which was highlighted when compared to other segmentation methods were employed for the ROI extracted from the high-resolution 24-bit color image first. After the clustering step for the segmentation, images morphologically improved with fill hole and closing operations in the post-process stage. These operations were applied only to the clusters that have the minimum pixel area. While K-means reach 90%, the accuracy of RGB thresholding varies seriously from 53 % to 74 % but fuzzy c-means has the highest record with 95 % and min of 92 %. Clustering-based color image segmentation way that has a novel line to improve the weighting proses by implementing group-local feature weighting take part in [41]. One of the remarkable parts of this approach is the cluster weighting plan to get a lower true-positive rate in the selection of initial centers. The extracted sub-features have dissimilar weights that relate their importance in the clusters different from methods of global weighting and local weighting. And they used the ICA optimization algorithm to calculate all weights automatically. Experiment results were obtained with the aerial dataset, skin lesion images, and white blood cell dataset. One of the hill climbing algorithms Mean-shift very effective in image segmentation but can not give a fit bandwidth for the initial image, to cope with this challenge scholars introduced the Neutrosophic Set (NS) theory, and each color channel converted in a neutrosophic set in [42]. Thus, results are improved using the adaptive mean shift clustering in low-contrast images and natural images.

3.2.3. Region-Based Techniques

Region-based segmentation is the application of growing, merging, and splitting operations or their combinations, starting from the initially selected seed region to reach homogeneous regions. Candidate pixels be selected with respect to predefined similarity criteria and considering the adjacency spatial relationships between pixels. Therefore, understanding the relationships among pixels and ranking them according to their qualification most essential part of that method. Some region-based approaches when having use the integration of both color and texture features but some of them to reach low computational cost, and high segmentation accuracy have considered color-alone or texture-alone, separately [43]. In region-based segmentation, the most widely used techniques are watershed algorithm that starts from the local minimums for segmenting the interior area like in MR images and histogram thresholding to overcome the distortion problems [44]. For getting lower the system complexity by use of color-alone features a novel color-based reduced and conditional ordering approach have presented in [43]. In this study, HSI color distance has been used for defining orders of color vectors. Then, proposed an adaptive merging algorithm, which involves quartile analysis, to specify the threshold automatically. According to results metrics, the performance of CRC ordering is moderate but over-segmentation was rare. Moth Flame Optimization (MFO), is one of the bio-inspired algorithms, employed for color image segmentation in [45]. The MFO inspired by insects rotated in a spiral path around light moths and the flame's position gives the modeled algorithm [46]. a* ve b* color feature was used that takes from the color image used in the region-based segmentation. According to observations, MFO is better than PSO (Particle Swarm Optimization) and GA (*Genetic Algorithm*) in terms of computation times for all the images.

3.2.4. Edge-Based Techniques

Edge-based methods have to need to detect meaningful brightness discontinuities in color images in the segmentation process. Due to other factors such as shadows, reflections, and noise can also be perceived as edges, the images have smooths with filters (*to reduce noise*) first and then have eliminates irrelevant edges from selected edges and grouping finally [12]. Processes generally are based on gradient calculations and the derivative of the image function that traditional edge detectors mostly used as Robert, Prewitt, Sobel, Canny, etc [47]. The development of learning-based techniques has turned edge detectors into learning-based ones. The method proposed in [48] is based on edge features and color indices of an image using CIELAB color space to automate the segmentation process in crops. A horizontal window has used to find the edge pixels by convolving. When the edge pixel is detected, have been computed its neighbors to find if any one or both of them contains an object or part of the object in L and A channels.

3.2.5. Hybrid Techniques

Hybrid-based methods are formed by combining two or more methods to improve image segmentation. Some hybrid methods use region growing and splitting while some of them combine thresholding-based and region-based methods. Researchers proposed a study by combining the edge-based level set method and region-based fuzzy C-Means clustering aim to achieve fast segmentation in [49]. Methods have lower computation time on segmentation in aerial images dataset. Hybrid methods have been improved in recent years with several studies by integration of neural-like computing. One of the most notable of them is Pulse-coupled neural networks that are modeled by getting inspired by a cat's visual cortex and these two-dimensional neural networks have been developed for high-performance biomimetic image processing [50]. Following some examples of operating PCNN neurons on gray-scale images with region-based methods [51], Xi et al. proposed a method that combined PCNN with region growing method to deal with color image segmentation [52]. This study enabled full access to color information by using a linking control unit (LCU). Firstly, the dataset images have been converted to CIELAB space that gives color information distance-based more efficiently from RGB space. After then, linking corresponding neurons depends on the situation that the distance of two pixels is lower than the prespecified threshold value. Results that have comparisons of performance metrics have been gained with the study on the Berkeley segmentation image dataset. Based on Spiking neural P systems, a kind of membrane system,

dynamic threshold neural P systems have modeled the neuron's firing [53]. DTNP-based hybrid color segmentation approach that 3 steps be included also the pre-process for denoising and the post-process to merge small regions proposed in [54]. Experiment results were obtained by a study on Berkeley segmentation datasets and compared against 17 state-of-the-art methods.

4. CONCLUSION

This study, have been surveyed fundamental color spaces and image segmentation techniques which are supervised and unsupervised as threshold-based, region-based, clustering-based, edge-based, and hybrid techniques with their applications in different areas. The table shown in Table 1 contains unsupervised techniques' pros and cons to compare briefly.

1. Unsupervised Color Image Segmentation Techniques Comparison Table

Image Segmentation Techniques Comparisons	*Pros*	*Cons*
Thresholding-Based	Easy to implement	Need predefined threshold value
Clustering-Based	Suitable and useful for both low- and high-dimensional data	Computational cost is quite high for large dataset
Region-Based	Considers the color information and spatial details at the same time	Quite expensive, high computational time and memory
Edge-Based	Fast and no require prior knowledge of the image.	More sensitive to noise
Hybrid Techniques	Quality information	Complex to implement

REFERENCES

[1] H.D. Cheng, X.H. Jiang, Y. Sun, Jingli Wang, "Color image segmentation: advances and prospects", *Pattern Recognition*, 34(12), 2259-2281, 2001.

[2] Y.N. Deng, B.S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8),800 - 810, 2001.

[3] A. Sriwastwa, S. Prakash, Mrinalini, S. Swarit, K. Kumari and S. S. Sahu, "Detection of Pests Using Color Based Image Segmentation," **2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT),**1393-1396, 2018.

[4] P. Szczypiński, A. Klepaczko, M. Pazurek, P. Daniel, "Texture and color based image segmentation and pathology detection in capsule endoscopy videos", *Comput Methods Programs Biomed*,113(1),396-411, 2014.

[5] Ö. Şengül, S. Öztürk, M. Kuncan," Color Based Object Separation in Conveyor Belt Using PLC", *Avrupa Bilim ve Teknoloji Dergisi*, (18), 401-412, 2020.

[6] S. Mousavi, B. Soleymani, N. Razmjooy, "Color image segmentation using neuro-fuzzy system in a novel optimized color space",*Neural Computing and Applications*, 23(5), 1513-1520, 2013.

[7] H. Hazlyna, N. Mashor, M. Y., Mokhtar, N. R., A. A. Salihah, R. Hassan, R. A. A. Raof, & M. K. Osman, "Comparison of acute leukemia Image segmentation using HSI and RGB color space", **In 10th International Conference on Information Science**, **Signal Processing and their Applications**,749-752, 2010.

[8] Jia, A. D., Li, B. Z., & Zhang, C. C., "Detection of cervical cancer cells based on strong feature CNN-SVM network", *Neurocomputing*, 411, 112-127, 2020.

[9] M. Belgiu, L. Drăguţ, "Comparing supervised and unsupervised multiresolution segmentation approaches for extracting buildings from very high resolution imagery", *ISPRS Journal of Photogrammetry and Remote Sensing*, 96, 67-75, 2014.

[10] Y. Cai, S. Mi, J. Yan, H. Peng, X. Luo, Q.Yang, J. Wang, "An unsupervised segmentation method based on dynamic threshold neural P systems for color images", I*nformation Sciences*, 587, 473-484, 2022.

[11] K.S. Fu, J.K. Mui, "A survey on image segmentation", *Pattern Recognition*, 13(1), 3-16, 1981.

[12] F. Garcia-Lamont, J. Cervantes, A. López, L. Rodriguez, "Segmentation of images by color features: A survey", *Neurocomputing*, 292, 1-27, 2018.

[13] P. Ganesan, B. S. Sathish, K. Vasanth, V. G. Sivakumar, M. Vadivel and C. N. Ravi, "A Comprehensive Review of the Impact of Color Space on Image Segmentation", **5th International Conference on Advanced Computing & Communication Systems (ICACCS)**, 962-967, 2019.

[14] R. Szeliski, **Computer vision algorithms and applications**, 2011.

[15] E. R. Marsel, Master's Thesis, Department of Electrical and Computer Engineering, "**Color Image Segmentation Using spectral clustering**", 2017.

[16] N. Khediri, M. B. Ammar and M. Kherallah, "Comparison of Image Segmentation using Different Color Spaces", **IEEE 21st International Conference on Communication Technology (ICCT)**, 1188-1192, 2021.

[17] K. Erdoğan, N. Yılmaz, "Shifting Colors to Overcome not Realizing Objects Problem due to Color Vision Deficiency", 2014.

[18] S. Chen and C. Liu, "Fast eye detection using different color spaces", **/IEEE International Conference on Systems Man and Cybernetics**, 521-526, 2011.

[19] Internet: Anonym, Peppermint, www.printpeppermint.com ,10.01.2022.

[20] A.M. Nazif, M.D. Levine, "Low level image segmentation: an expert system", **IEEE Trans. Pattern Anal. Machine Intell**., 6 (5), 555-577, 1984.

[21] Li, X., Wang, K., Li, Z., "A rule based image segmentation system (RBISS)", **IEEE International Conference on Systems, Man, and Cybernetics**, 1, Beijing, China, 652–653, 1988.

[22] S.Zheng,S.Javasumana,B.Romera-Paredes, V. Vineet, Su Z., Du D., Huang C., P.H.S. Torr, "Conditional random fields as recurrent neural networks", **Proceedings of the IEEE Computer Society Conference on Computer Vision**,1529- 1537, Santiago, Chile, 2015.

[23] R. Olaf, F. Philipp, B. Thomas, "U-Net: convolutional networks for biomedical image segmentation", **Proceedings of the Medical Image Computing and Computer-Assisted Intervention**, 234-241, Munich, Germany, 2015.

[24] Y. Wang, Y. Lu and Y. Li, "A New Image Segmentation Method Based on Support Vector Machine", **2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)**,177-181, 2019.

[25] M. Hajabdollahi, R. Esfandiarpoor, P. Khadivi, S.M.R. Soroushmehr, N. Karimi, S. Samavi, "Simplification of neural networks for skin lesion image segmentation using color channel pruning", *Computerized Medical Imaging and Graphics*, 82, 2020.

[26] I. Libouga, L. Bitjoka, D. Libouga Li Gwet, O.Boukar, A. Nlôga, "A supervised U-Net based color image semantic segmentation for detection & classification of human intestinal parasites", *Advances in Electrical Engineering, Electronics and Energy*, 2, 2022.

[27] S. Sodjinou, V. Mohammadi, A. Mahama, P. Gouton, "A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images", *Information Processing in Agriculture*, 9(3), 355-364, 2022.

[28] N. Otsu, "A threshold selection method from gray-level histograms", **IEEE Trans. Syst. Man Cybern.,** 9, 62-66, 1979.

[29] J.N. Kapur , P.K. Sahoo, A.K. Wong "A new method for gray-level picture thresholding using the entropy of the histogram" *Comput. Vis. Graph. Image Process.*, 29 (3) , 273-285, 1985.

[30] W.H. Tsai, "Moment-preserving thresholding: A new approach" *Comput. Vis. Graph. Image Process.*, 29 (3), 377-393, 1985

[31] L. He, S. Huang, "An efficient krill herd algorithm for color image multilevel thresholding segmentation problem", *Applied Soft Computing*, 89, 2020.

[32] K. Dutta, D. Talukdar, S.S. Bora, "Segmentation of unhealthy leaves in cruciferous crops for early disease detection using vegetative indices and Otsu thresholding of aerial images", *Measurement*,Volume 189, 2022.

[33] Z. Xing, "An improved emperor penguin optimization based multilevel thresholding for color image segmentation", *Knowledge-Based Systems*, 194, 2020.

[34] M. El Baz, T. Zaki and H. Douzi, "An Improved Method for Red Segmentation Based Traffic Sign Detection", **IEEE 9th International Conference on Information, Communication and Networks (ICICN)**, 2021.

[35] G. Yang, M. Han, Y. Wang, Z. Wang and G. Wei, "YCbCr Color Space Based Image Segmentation Algorithm with Feedback in the Underground Cable Tunnel", **Chinese Automation Congress (CAC)**, 692-696, 2019.

[36] S.I.R. Rodríguez, F. de A.T. de Carvalho,"Fuzzy clustering algorithms with distance metric learning and entropy regularization", *Applied Soft Computing*, 113 (A), 2021.

[37] J. C. Dunn , "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics*, 3(3), 32-57, 1973.

[38] G. Schu, J. Scharcanski,"A new method for building adaptive Bayesian trees and its application in color image segmentation", *Expert Systems with Applications*, 98, 57-71, 2018.

[39] T. Wu , X. Gu, J. Shao, R. Zhou and Z. Li, "Colour image segmentation based on a convex K-means approach", *IET Image Process.* 15,1596-1606, 2021.

[40] J. Malhotra, S. Jha, "Fuzzy c-means clustering based colour image segmentation for tool wear monitoring in micro-milling", *Precision Engineering*, 72, 690-705, 2021.

[41] A. G. Oskouei, M. Hashemzadeh, "CGFFCM: A color image segmentation method based on cluster-weight and feature-weight learning", *Software Impacts*, 11, 2022.

[42] Y. Guo, A. Şengür, Y. Akbulut, A. Shipley, "An effective color image segmentation approach using neutrosophic adaptive mean shift clustering", *Measurement*, 119, 28-40, ISSN 0263-2241,2018.

[43] H. Shih, E. Liu, "New quartile-based region merging algorithm for unsupervised image segmentation using color-alone feature", *Information Sciences*, 342, 24-36, 2016.

[44] M. Zawish, A. A. Siyal, S. H. Shahani, A. Z. Junejo, A. Khalil, "Brain Tumor Segmentation through Region-based, Supervised and Unsupervised Learning Methods: A Literature Survey: Brain Tumor Segmentation through Image Processing Methods: A Literature Survey". *British Journal of Healthcare and Medical Research*, *6*(2), 08–26, 2019.

[45] V. Jaiswal, V. Sharma, S. Varma, "MMFO: modified moth flame optimization algorithm for region based RGB color image segmentation", *International Journal of Electrical and Computer Engineering (IJECE)*, 2020.

[46] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm",*Knowledge-Based Systems*, 89, 228-249, 2015.

[47] R. Priyadharsini, T. Sree Sharmila, "Object Detection In Underwater Acoustic Images Using Edge Based Segmentation Method", *Procedia Computer Science*, 165, 759-765, 2019.

[48] C. Pal, S. Pratihar, S. Chatterji and I. Mukherjee, "Automatic Rice Crop Extraction using Edge based Color Features and Color Indices", **2nd Asian Conference on Innovation in Technology**, 1-8, 2022.

[49] Z. Ye, H. Yin and Y. Ye, "Aerial Scene Classification and Information Retrieval via Fast Kernel Based Fuzzy C-Means Clustering", **Information Management and Big Data: Communications in Computer and Information Science Book Series**, Springer International Publishing, 898, 98-111, January 2019.

[50] K. Zhan, J. Shi, H. Wang, Y. Xie, Q. Li, "Computational mechanisms of pulse-coupled neural networks: A comprehensive review", *Archives of Computational Methods in Engineering*, 2017.

[51] R.D. Stewart, I. Fermin, M. Opper, “Region growing with pulse-coupled neural networks: an alternative to seeded region growing”, *IEEE Trans. Neural Netw.* 13, 1557–1662, 2002.

[52] G. Xu, X. Li, B. Lei, K. Lv, “Unsupervised color image segmentation with the color-alone feature using region growing pulse coupled neural network”, *Neurocomputing*, 306, 1-16, 2018.

[53] H. Peng, J. Wang, M. J. Pérez-Jiménez, A. Riscos-Núñez, “Dynamic threshold neural P systems”, *Knowledge-Based Systems*, 163, 875-884, 2019.

[54] Y.Cai, S. Mi, J. Yan, H. Peng, X. Luo, Q. Yang, J. Wang, “An unsupervised segmentation method based on dynamic threshold neural P systems for color images”, *Information Sciences*, 587, 473-484, 2022.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

What Word Embeddings Tell Us About Turkish Coreference Resolution

Seniz DEMİR

Department of Computer Engineering, MEF University, Istanbul, Turkey
demirse@mef.edu.tr

***Abstract*—** Coreference resolution, which aims to resolve all mentions in a text that refer to the same entity, is a fundamental task in natural language understanding. Encoding semantic information into mention span representations is of great importance in identifying coreference mentions. Word embeddings have been utilized as sources of semantic knowledge in various tasks but their expressive power in Turkish coreference resolution have not been studied yet. In this work, we examined how different word embeddings perform in measuring the relatedness of coreference mention spans in Turkish. We experimented with character, subword, morphological, and word embeddings retrieved from pretrained language models in two different settings. First, we measured to what extent different embeddings reflect the semantic closeness of mentions in the same coreference chains using well-known similarity measures. Second, we performed clustering of mention representations in order to assess the discriminatory power of different word embeddings. Our results revealed several insights about the potential use of embeddings in Turkish coreference resolution.

***Keywords*—** word embedding, coreference resolution, Turkish

1. INTRODUCTION

Coreference resolution deals with automatically grouping mentions in a text that refer to the same real world entity where a mention (e.g., the mention of a person name or a location) might take different forms such as a noun phrase, a named entity, or a pronoun. This challenging task, which plays an important role in discourse analysis, has attracted considerable attention in several language processing applications from question answering [1] to machine translation [2]. Previous research explored various rule-based [3] and learning-based solutions [4] to resolve coreferring entities in a text. These studies formulated the task in different models [5] such as mention-pair model where the task is to determine whether two mentions are coreferent or not, and the mention-ranking model that aims to select the correct antecedent for a given mention from a group of antecedent mentions.

Unsupervised and supervised approaches have benefited from several sources of information in order to connect mentions to their antecedents such as the lexical relations (e.g., synonym and hyponym) between mentions, the number, gender, and part-of-speech (POS) tag features of mentions, and the distance of mentions within sentences. Incorporating world knowledge as features [6] or benefiting from semantic features of mentions [7] have also been shown to be effective in resolving coreference mentions. Recent years have witnessed the increasing popularity of word embeddings as high-quality vectoral representations of semantics and their major impact in achieving state-of-the-art results on this task.

Word embeddings (dense vectoral representations) are effective in capturing syntactic and semantic features of words in the discourse and hence words with similar meaning get closer in the defined vector space. Early approaches to word embedding generation have learned static vectors from large text collections (e.g., Word2Vec [8] and Glove [9]). In these approaches, a single context-independent vectoral representation is produced for all senses of a word. Moreover, no representation can be obtained for out-of-vocabulary (OOV) words and multi-word expressions cannot be handled. Neural language models have undoubtedly contributed to the development of more efficient ways of producing word embeddings. Approaches that utilize language models can generate context-dependent word vectors such that a word might be associated with different representations according to its context (surrounding words). Moreover, these approaches better handle rare and out-of-vocabulary words since language models can be trained with words or word constituents such as characters, character sequences, and morphemes. In cases where word constituents are learned by the model, their vectoral representations are often

summed [10], averaged [11], or fed to another neural model [12] in order to obtain the final word embedding.

Word embeddings are typically learned in an unsupervised fashion and this semantic source can be obtained at a small cost in cases where an annotated corpus for the downstream task does not exist or is limited. These representations once employed to supervised approaches as semantic features could address multiple issues including data sparsity in the input. Turkish is a morphologically-rich language and one of the resource-poor languages in coreference resolution. There are only a few research studies mostly on pronoun resolution [13, 14] and one publicly available coreference corpus [15] of limited size. To our best knowledge, the potential of word embeddings in reflecting semantic features of Turkish coreference mentions has not been studied yet. Moreover, the empirical utility of using different word embeddings in combination with supervised approaches has not been studied in preliminary Turkish coreference works.

This work is the first study that explored how effective word embeddings are in capturing the semantic relatedness of mention pairs. For this purpose, we represented coreference mentions in the same clusters using different embedding forms and computed their closeness using well-known similarity measures. We built mention span representations using character, subword, morphological, and word embeddings obtained from pretrained language models, and compared their individual and combined performances in reflecting already determined coreferent relations. In addition, we measured how beneficial embeddings are in resolving relations between coreferring entities in an unsupervised setting. To achieve this, we clustered mention span representations using K-Means and Mini Batch K-Means methods and assessed the discriminative power of different embedding forms. In all experiments, we used the same dataset and reported interesting insights that we obtained during the study.

2. EXPERIMENTAL SETUP

In order to explore how reliable word embeddings are as a semantic resource in relating Turkish coreference mentions, we conducted a series of experiments. The first set of experiments aimed to assess the degree of similarity between mentions by applying two distance measures to their embeddings. Clustering mention spans based on their embeddings and hence evaluating whether an unsupervised solution with no feature engineering was effective enough in identifying coreferring entities was the goal of the second set of experiments.

2.1. Dataset

We used Marmara Turkish Coreference Corpus [15] which is the only publicly available corpus in the literature. The corpus contains manually annotated mentions and coreference chains of 33 documents retrieved from METU-Sabancı Turkish Treebank [16]. In the corpus, each noun phrase, pronoun, or nominalized adjective that corresponds to a specific entity is identified as a mention and the largest possible token span that describes the entity is considered for annotation. The coreference chains are formed by clustering coreference mentions that refer to the same entity in the discourse. The corpus consists of 5170 mentions and 944 coreference chains where the number of mentions in a chain ranges from 2 to 66, with an average of 5.48. The following is a representative coreference chain from the corpus where coreferring entities are underlined:

- Gel bak, sana evi göstereyim
  Come look, I'll show you the house
- Gecenin hangi saatinde eve dönsem de, yakındaki açık gece mağazasından yiyecek birşeyler alabilirdim
  No matter what time of the night I returned home, I could buy something to eat at the nearby open night store
- Yarın akşam, bana burada kaldığım evi bulan - ev de tanıdığım bir profesörün ya - Madam Kuve'ye yemeğe çağrılıyım
  *Tomorrow evening, I am invited to dinner at Madam Kuve who found me the house where I am staying (*the house *is of a professor I know)*

2.2. Word Embeddings

Neural language models with limited vocabulary often suffer from data sparsity problems in morphologically-rich languages where the number of possible word forms is very high. Moreover, it is not practical to extend the model vocabulary by considering all derivational and inflectional suffixes since this significantly increases the number of parameters that need to be learned during training. One particular solution that has been heavily used for addressing lexical coverage is to augment word embeddings with the embeddings of word constituents at different granularities. For instance, previous studies on morphologically-rich languages (e.g., named entity recognition [17] and part-of-speech tagging [18]) have concatenated character or morphological embeddings with word embeddings in order to obtain final word representations [19]. Here, we experimented with publicly available pretrained Turkish language models that provide character, subword, morphological, and word embeddings (shown in Table 1), and compared their individual and combined performances in assessing the semantic relatedness of coreference mentions.

A language model based on characters has a small vocabulary even for morphologically-rich languages, and alleviates rare and OOV word problems. However, individual characters might not reflect syntactic or morphological constituents of a word and the differences in surface forms of semantically similar words due to the use of allomorphs (different forms of a morpheme) cannot be learned [20]. In this study, we used a RoBERT-a based

Turkish language model [21] (CM)[1] which was trained on the Turkish split of the OSCAR corpus (~27GB) [22]. For each word, the mean of character embeddings were taken as the final embedding.

Table 1. Language models

Model	*Type*	*Architecture*	*Tokenizer*	*#Layer*	*#Head*	*Dimension*	*Vocabulary*	*Corpus*
CM	Character	RoBERTa	ByT5	8	8	512	384	Oscar
SM1	Subword	RoBERTa	BPE	8	8	512	44.5 K	Oscar
SM2	Subword	RoBERTa	WordPiece	8	8	512	44.5 K	Oscar
SM3	Subword	GloVe	BPE	-	-	300	50 K	Wikipedia
MM	Morphological	RoBERTa	Zemberek	8	8	512	44.5 K	Oscar
WM1	Word	BERT	WordPiece	12	12	768	128 K	Wikipedia, Oscar, Opus
WM2	Word	RoBERTa	WhiteSpace	8	8	512	44.5 K	Oscar

Developing a language model with decent vocabulary is also possible by splitting words into their subwords (character sequences). The intuition behind subword based models is to keep a frequently used word as is while splitting a rare or unknown word into smaller commonly used chunks in the hope that the word meaning can be reconstructed from these chunks. Moreover, these language models raise the possibility of OOV words being decomposed into subwords that already appear in the model vocabulary. Earlier approaches have utilized different strategies to encode a word as a sequence of subwords [23] such as the Byte Pair Encoding (BPE) [24], Word-Piece [25], Unigram [26], and SentencePiece [27]. In our work, we experimented with two RoBERT-a based Turkish language models [21] that utilize BPE (SM1)[2] and WordPiece (SM2)[3] encodings. In addition, we used pretrained subword embeddings based on BPE encoding (SM3) [28]. These context-independent embeddings were trained on Turkish Wikipedia articles using GloVe [9]. In all models, the final word embedding was obtained by averaging the embeddings of its subwords.

For morphologically-rich languages, one logical word decomposition strategy is to consider the morphemes of a word as its meaning bearing units [29]. The lemma, the sequence of suffixes, and the part-of-speech tag are some morphological features that have been used while encoding the semantic properties of a word [30, 31]. Here, we retrieved morphological embeddings from a RoBERT-a based language model [21] (MM)[4] trained on a corpus that was tokenized by the Zemberek morphological parser[5]. All morphemes produced by the parser were considered as distinct tokens and the average of their embeddings was used as the final word embedding.

We benefitted from two different language models for word embeddings. The first BERT-based model (WM1)[6] was trained on a 35 GB text collected from multiple corpora. One corpus from that collection was used to train the second RoBERT-a based model (WM2)[7] which is small as compared to the first model. In both models, we also experimented with different vector combinations as the final word embedding: i) the last hidden layer (+LL), ii) the second to the last hidden layer (+SLL), and iii) the sum of the last four hidden layers (+SFL).

Lastly, we obtained mention span representations in three different ways. First, we considered the dependency relations between all words within a span and used the embedding of the head word as the span embedding (+HW). Second, we obtained the span embedding by averaging only the embeddings of the first and last words of the span (+FLW). Third, we used the mean of all word embeddings as the span embedding (+AW).

2.3. Evaluation Measures

We considered all possible embedding configurations by selecting at most one model for each type (a single model or a combination of multiple models from Table 1). In a configuration with multiple models, the final mention span embedding was obtained by concatenating model embeddings. Therefore, the size of span embeddings ranged from 300 (SM3) to 2304 (CM⊕SM1 or SM2⊕ MM⊕WM1). We performed the same evaluations for each embedding configuration and used the coreference chains that were already given in our corpus as the ground truth.

In our first set of experiments, we computed pairwise cosine similarities of mention spans that appear in the same coreference chain and used the average of these scores as the chain similarity score (Intra-Sim). Additionally, we quantified how dissimilar coreference chains are in an embedding configuration. For each chain, we computed the average similarity (Inter-Sim) of all mention spans in the chain to mention spans in the remaining chains. In order to minimize the computation overhead, we randomly selected only one mention span from the remaining coreference chains. Finally, we computed the configuration cosine similarities (ConfIntra-Sim, ConfInter-Sim) by averaging the related scores of all coreference chains as follows:

(i) $$\text{Intra-Sim}_z=\frac{1}{T}\sum_{t=1}^{T}\text{CosineSim}(\text{pair}_t) : \text{pair}_t=(m_z^r, m_z^s),\ m_z^r, m_z^s \in \text{Chain}_z, r \neq s$$

[1] https://huggingface.co/ctoraman/RoBERTa-TR-medium-char

[2] https://huggingface.co/ctoraman/RoBERTa-TR-medium-bpe-44k

[3] https://huggingface.co/ctoraman/RoBERTa-TR-medium-wp-44k

[4] https://huggingface.co/ctoraman/RoBERTa-TR-medium-morph-44k

[5] https://github.com/ahmetaa/zemberek-nlp

[6] https://huggingface.co/dbmdz/bert-base-turkish-128k-cased

[7] https://huggingface.co/ctoraman/RoBERTa-TR-medium-word-44k

(ii) $\text{Inter-Sim}_z = \frac{1}{KR} \sum_{k=1}^{K} \sum_{r=1}^{R} \text{CosineSim}(m_z^r, m_k^s)$: $m_z^r \in \text{Chain}_z, m_k^s \in \text{Chain}_k, k \neq z$

(iii) $\text{ConfIntra-Sim} = \frac{1}{K} \sum_{k=1}^{K} \text{Intra-Sim}_k$

(iv) $\text{ConfInter-Sim} = \frac{1}{K} \sum_{k=1}^{K} \text{Inter-Sim}_k$

Table 2. Configuration cosine similarity scores according to embedding types

Type	**ConfIntra-Sim**				**ConfInter-Sim**			
	Max		**Min**		**Max**		**Min**	
	Config.	*Score*	*Config.*	*Score*	*Config.*	*Score*	*Config.*	*Score*
Character	CM+AW	0.667	CM+HW	0.660	CM+AW	0.441	CM+HW	0.378
Subword	SM2+AW	0.606	SM3+HW	0.467	SM2+AW	0.419	SM3+HW	0.034
Morphological	MM+AW	0.561	MM+HW	0.543	MM+AW	0.344	MM+HW	0.304
Word	WM2+AW	0.598	WM1+HW	0.554	WM2+AW	0.431	WM1+HW	0.292

Table 3. Configuration euclidean distance scores according to embedding types

Type	**ConfIntra-Euc**				**ConfInter-Euc**			
	Max		**Min**		**Max**		**Min**	
	Config.	*Score*	*Config.*	*Score*	*Config.*	*Score*	*Config.*	*Score*
Character	CM+HW	12.202	CM+AW	11.696	CM+HW	17.525	CM+AW	15.821
Subword	SM1+HW	19.877	SM3+AW	3.869	SM1+HW	24.902	SM3+AW	5.819
Morphological	MM+HW	19.114	MM+AW	18.088	MM+HW	24.042	MM+AW	22.486
Word	WM1+HW	23.066	WM2+AW	17.546	WM1+HW	29.612	WM2+AW	21.380

where K is the number of chains, R and T are the number of total mentions and the number of unique mention pairs in the coreference chain z, m_z^r is the embedding of the r^{th} mention in the chain z, and m_k^s is the embedding of the randomly chosen mention from the chain k. Additionally, for each configuration, we computed the average Euclidean distance scores of coreference chains (Intra-Euc), how far away the chains are in a configuration (Inter-Euc), and the configuration distance scores (ConfIntra-Euc, ConfInter-Euc) by replacing the cosine similarity measure with Euclidean distance measure in the above formulations.

In our second set of experiments, we clustered mention span embeddings using K-Means and Mini Batch K-Means methods[8]. We considered each coreference chain as a distinct cluster and hence used 944 as the number of clusters that should be formed by these methods. We measured the degree of agreement between gold clusters and those identified by a clustering approach using Adjusted random scores and Silhouette scores. We computed clustering scores for all embedding configurations used in the first experiment set.

3. RESULTS

3.1. Cosine Similarity and Euclidean Distance Scores

We first examined the configurations where only one type of embedding is used. The cosine similarity results given in Table 2 revealed that the highest cosine similarity score between the mentions of the same chain (ConfIntra-Sim) was achieved by using character embeddings only (0.667) whereas the use of context-independent subword embeddings obtained by BPE encoding received the lowest similarity score (0.467). As anticipated, the use of character embeddings also increased the similarity between the mentions of different coreference chains (0.378-0.441). The range of measured similarities between coreference mentions (ConfIntra-Sim) in configurations with context-dependent embeddings (excluding the SM3 model) was observed to be larger for subword embeddings (0.538-0.606) as compared to other types. However, the largest range was observed between the mentions of different chains (ConfInter-Sim) for word embeddings (0.292-0.431).

We experimented with three different methods of building mention span embeddings. Our results showed that forming span embeddings from the embeddings of all words in the span (+AW) yielded the highest similarity scores for all embedding types. However, considering only head word embeddings (+HW) received the lowest scores for all cases. It is particularly worth mentioning that the similarity between the mentions of different chains (ConfInter-Sim) increased once all words are considered for building span representations.

One of our observations about the word based language models was that the smaller RoBERTa-based model (WM2) received higher similarity scores (ConfIntra-Sim, ConfInter-Sim) than the larger BERT-based model (WM1) for all span representation forms (+HW, +FLW, +AW). However, we did not observe any significant difference between the similarity scores according to the hidden layers considered for obtaining word embeddings (+LL, +SLL, +SFL). In configurations with only subword embeddings, the context-independent language model (SM3) achieved the lowest similarity scores once compared to other models (SM1 and SM2) and the scores were significantly low for mentions from different chains (ConfInter-Sim). Moreover, the results demonstrated that WordPiece encoding improved the scores as compared to the BPE encoding for all span representation forms.

The Euclidean distance scores that we obtained from our experiments (shown in Table 3) supported most of our

[8] https://scikit-learn.org/stable/modules/clustering.html

findings from cosine similarity evaluations. Using context-independent embeddings with comparably smaller dimension (SM3) resulted in significantly low distance scores (ConfIntra-Euc, ConfInter-Euc) and hindered a clear separation between different coreference chains. Our analysis showed that from among context-dependent language models, using the character-based one (CM) achieved the lowest average distance between mentions of the same (ConfIntra-Euc) and different chains (ConfInter-Euc). The measured configuration distances for character and morphological embeddings had a smaller range than those of subword and word embeddings.

The results highlighted that using all words in a mention span while building its representation decreased the distance between mentions for all cases (ConfIntra-Euc and ConfInter-Euc). As seen in cosine similarity evaluations, the smaller language model (WM2) was found to be more successful than the larger model (WM1) in gathering coreference mentions together. Finally, we observed that BPE encoding returned better results than WordPiece encoding in subword configurations.

As our final analysis, we examined the configurations with multiple language models. We observed that all configurations with multiple models received lower cosine similarity scores than the configuration with character embeddings only and we attributed this to the fact that the span embeddings in such configurations have high dimensions. However, our in-depth analysis revealed several fruitful insights about the performance of individual models once combined with other language models. We particularly focused on the additive performance of character, subword, and morphological embeddings once used to augment word embeddings. The results demonstrated that higher similarity scores can be achieved by concatenating character and subword embeddings to word embeddings. Moreover, the contribution of character embeddings to the similarity score was shown to be higher than that of subword embedding. On the other hand, we observed that the incorporation of morphological embeddings negatively impacted the similarity scores and the lowest scores were achieved if morphological embeddings were combined with subword embeddings where BPE encoding was used. Table 4 presents some embedding configurations that reflect the general tendency in measured similarity scores for cases with multiple language models.

Table 4. Cosine similarity scores for different configurations

Configuration	*ConfIntra-Sim*	*ConfInter-Sim*
WM2+AW	0.598	0.431
WM2+CM+AW	0.621	0.433
WM2+SM2+AW	0.602	0.426
WM2+MM+AW	0.577	0.387
WM2+MM+SM1+AW	0.569	0.372
WM2+MM+SM2+AW	0.586	0.397
WM2+SM2+CM+AW	0.616	0.428
WM2+MM+SM2+CM+AW	0.598	0.403

In Euclidean distance score evaluations, we observed that concatenating character, subword, and morphological embeddings to word embeddings resulted in higher distance values due to increased span embedding dimensions. However, the amount of measured increase for character and subword embeddings was less than that for morphological embeddings. In addition, the configurations where the word embedding was combined with only one other embedding type received lower distance scores than those with multiple embeddings.

3.2. Clustering Scores

In our first analysis, we examined the silhouette and adjusted random scores that the configurations with a single embedding received. As shown in Table 5, the measured scores were in the range of 0.0677 to 0.1949 for K-Means and -0.0349 to 0.1052 for Mini Batch K-Means. Unfortunately, these scores were close to 0 and hence demonstrated that the distances between identified clusters are not significant enough. From among all types, character embeddings and word embeddings were shown to be the most and least effective types in clustering. As given in Table 6, the adjusted random scores also showed that representing mention pairs using a particular embedding type resulted in pretty low level of agreement between the identified and gold clusters. Nonetheless, the configuration where only character embeddings are used reached the highest agreement level both in K-Means and Mini Batch K-Means approaches. Our second analysis revealed that configurations where multiple types of embeddings are concatenated did not reach the scores obtained by character embeddings only. Our experiments demonstrated that using any form of embedding in a straightforward unsupervised clustering setting cannot cluster coreference mentions at an acceptable confidence level.

Table 5. Silhouette scores

	K-Means		**Mini Batch K-Means**	
Type	*Max*	*Min*	*Max*	*Min*
Character	0.1949	0.1588	0.1052	0.0701
Subword	0.1115	0.0906	0.0190	-0.0177
Morphological	0.0900	0.0789	-0.0142	-0.0177
Word	0.0890	0.0677	-0.0194	-0.0349

Table 6. Adjusted random scores

	K-Means		**Mini Batch K-Means**	
Type	*Max*	*Min*	*Max*	*Min*
Character	0.1843	0.1382	0.1542	0.1082
Subword	0.1277	0.1004	0.0753	0.0620
Morphological	0.1124	0.0927	0.0884	0.0752
Word	0.1547	0.0492	0.0994	0.0412

4. CONCLUSION

Coreference resolution plays a vital role in several language understanding tasks. Unfortunately, there is a handful of Turkish coreference resolution studies in the literature. To the best of our knowledge, this work is the first study that explored whether and to what extent dense

vectoral representations can be used in capturing semantic information of coreference mentions in Turkish. Our work benefited from publicly available pretrained language models in order to retrieve character, subword, morphological, and word embeddings. Using cosine similarity and Euclidean distance measures, our work examined the semantic relatedness of mention spans whose representations were obtained by obtaining the final embeddings via three different strategies. Moreover, these embeddings were used to cluster mention spans in an unsupervised setting via K-Means and Mini Batch K-Means approaches, and their discriminative performances were evaluated. As future work, we are planning to build a larger Turkish coreference dataset and develop a neural model in order to resolve mention-pair problem in Turkish using the insights that we obtained from this study.

REFERENCES

[1] S. Bhattacharjee, R. Haque, G. M. de Buy Wenniger, A. Way, "Investigating Query Expansion and Coreference Resolution in Question Answering on BERT", **2020 Natural Language Processing and Information Systems Conference, 47–59**, 2020.

[2] L. M. Werlen, A. Popescu-Belis, "Using Coreference Links to Improve Spanish-to-English Machine Translation", **2017 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON),** Valencia, Spain, 30-40, 2017.

[3] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, D. Jurafsky, "Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task", **2011 15th Conference on Computational Natural Language Learning: Shared Task**, 28-34, 2011.

[4] N. Stylianou, I. Vlahava, "A neural Entity Coreference Resolution review", *Expert Systems with Applications*, 168, 2021.

[5] A. Ferreira Cruz, G. Rocha, H. Lopes Cardoso, "Coreference Resolution: Toward End-to-End and Cross-Lingual Systems", *Information*, 11(74), 2020.

[6] A. Rahman, V. Ng, "Coreference Resolution with World Knowledge", **2011 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**, Oregon, USA, 814-824, 2011.

[7] K. Lee, L. He, L. Zettlemoyer, "Higher-Order Coreference Resolution with Coarse-to-Fine Inference", **2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, New Orleans, Louisiana, 687-692, 2018.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, "Distributed representations of words and phrases and their compositionality", *In Advances in Neural Information Processing Systems*, 3111-3119, 2013.

[9] J. Pennington, R. Socher, C. Manning, "GloVe: Global Vectors for Word Representation", **2014 Conference on Empirical Methods in Natural Language Processing**, Doha, Qatar, 1532-1543, 2014

[10] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, "Enriching Word Vectors with Subword Information", *Transactions of the Association for Computational Linguistics*, 5, 135-146, 2017.

[11] J. Zhao, S. Mudgal, Y. Liang, "Generalizing Word Embeddings using Bag of Subwords", **2018 Conference on Empirical Methods in Natural Language Processing**, Brussels, Belgium, 601-606, 2018.

[12] G. Aras, D. Makaroğlu, S. Demir, Altan Cakir, "An evaluation of recent neural sequence tagging models in Turkish named entity recognition", *Expert Systems with Applications*, 182, 2021.

[13] H. Abaci, M. Eminagaoglu, I. Gor, Y. Kılıcaslan, "Pronoun Resolution for Tweets in Turkish", *International Journal of Intelligent Systems and Applications in Engineering*, 10(2), 207–215. 2022.

[14] Y. Kılıçaslan, E. S. Güner, S. Yıldırım, "Learning-Based Pronoun Resolution for Turkish with a Comparative Evaluation", *Computer Speech and Language*, 23(3), 311–331, 2009.

[15] P. Schüller, K. Cıngıllı, F. Tunçer, B. G. Sürmeli, A. Pekel, A. H. Karatay, H. E. Karakaş, **Marmara Turkish Coreference Corpus and Coreference Resolution Baseline**, 2017.

[16] B. Say, D. Zeyrek, K.l Oflazer, U. Özge, "Development of a corpus and a treebank for present-day written Turkish", **2002 11th International Conference of Turkish Linguistics (ICTL)**, 183-192, 2002.

[17] O. Güngör, T. Güngör, and S. Üsküdarlı. "The effect of morphology in named entity recognition with sequence tagging", *Natural Language Engineering* , 25(1), 147–169, 2019.

[18] A. Saimaiti, L. Wang, T. Yibulayin, "Learning Subword Embedding to Improve Uyghur Named-Entity Recognition", *Information*,10(4), 2019.

[19] M. Labeau, A. Allauzen, "Character and Subword-Based Word Representation for Neural Language Modeling Prediction", **2017 1st Workshop on Subword and Character Level Models in NLP**, Copenhagen, Denmark, 1-13, 2017.

[20] A. Üstün, M. Kurfalı, B. Can, "Characters or Morphemes: How to Represent Words? ", **2018 3rd Workshop on Representation Learning for NLP**, Melbourne, Australia, 144-153, 2018.

[21] C. Toraman, E. H. Y., F. Şahinuç, O. Ozcelik, **Impact of Tokenization on Language Models: An Analysis for Turkish,** arXiv, 2022.

[22] P. J. O. Suárez, B. Sagot, L. Romary, "Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures", **2019 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7),** Cardiff, United Kingdom, 2019.

[23] K. Bostrom, G. Durrett, "Byte Pair Encoding is Suboptimal for Language Model Pretraining", **2020 Findings of the Association for Computational Linguistics: EMNLP**, 4617-4624, 2020.

[24] R. Sennrich, B. Haddow, A. Birch, "Neural Machine Translation of Rare Words with Subword Units", **2016 54th Annual Meeting of the Association for Computational Linguistics**, Berlin, Germany, 1715-1725, 2016.

[25] M. Schuster, K. Nakajima, "Japanese and Korean voice search", **2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**, 5149-5152, 2012.

[26] T. Kudo, "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates", **2018 56th Annual Meeting of the Association for Computational Linguistics,** Melbourne, Australia, 66-75, 2018.

[27] T. Kudo, J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing", **2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, Brussels, Belgium, 66-71, 2018.

[28] B, Heinzerling, M. Strube, "BPEmb: Tokenization-free Pretrained Subword Embeddings in 275 Languages", **2018 11th International Conference on Language Resources and Evaluation**, Miyazaki, Japan, 2018.

[29] T. Luong, R. Socher, C. Manning, "Better Word Representations with Recursive Neural Networks for Morphology", **2013 17th Conference on Computational Natural Language Learning**, Sofia, Bulgaria, 104-113, 2013.

[30] O. Avraham, Y. Goldberg, "The Interplay of Semantics and Morphology in Word Embeddings", **2017 15th Conference of the European Chapter of the Association for Computational Linguistics**, Valencia, Spain, 422-426, 2017.

[31] R. A. Salama, A. Youssef, A. Fahmy, "Morphological Word Embedding for Arabic", *Procedia Computer Science*, 142, 83–93, 2018.

.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

A System for Optimization of Workloads of Test Units and Monitoring of Their Processes

Pınar BÜLBÜL UĞUREL[1], Hasan Şakir BİLGE[2]

[1]Computer Science, Institute of Informatics, Gazi University, Ankara, Turkey
[2] Electrical and Electronics Engineering, Faculty of Engineering, Gazi University, Ankara, Turkey
pinar.ugurel@gazi.edu.tr, bilge@gazi.edu.tr

***Abstract*—** Software occupies a huge place in our daily life. With the increase in the expectations of customers and users from the software, the size and complexity of the software have started to increase. It is inevitable that there will be errors in the software development process. Minimizing these errors before the software goes live is only possible by testing. Although testing processes are so important, it is seen that the necessary time is not allocated for this phase in software development projects. This is due to the fact that more time was spent than planned especially in the previous stages of software development processes and tight deadline pressures. Especially in today's internet world, it has become very important to enter the market quickly and to be a pioneer. However, launching faulty software to users can have huge consequences. In this study, a system has been developed that will operate in a transparent way where a test unit can manage requests for different test types from many different sources, plan in an objective manner, monitor processes and their performance criteria, and report when necessary. Two different objectives are studied according to the purposes of meeting the largest number of requests and prioritizing the most important requests according to their importance level. In addition to the constraints of the Multi-dimensional Knapsack problem, the proposed planning model considers the prerequisite condition to be considered in the completion and planning of requests. This prerequisite situation is when some requests have another request that must be completed before they can be completed. Traceability through the system will provide transparency to requesters, team members, and managers. In this study, the test unit is considered, but job planning and tracking of processes are also valid concerns for other business units. The proposed system can also be used for other business units.

***Keywords*—** job selection, project selection, resource allocation, job planning, web application, process tracking

1. INTRODUCTION

The stages of software development can be divided into 4 main parts as analysis, design, coding, and testing [1]. It can be verified by testing that the software meets the requirements specified during the analysis and design phases, responds correctly to all relevant inputs, achieves the overall results desired by the stakeholders, and operates within an acceptable period of time [2]. Errors are inevitable in the software development process. These errors can be caused by reasons such as unclear requirements in the analysis, misunderstandings in interpersonal communication, mistakes that can be made in the design, and incorrect coding.

Software testing is the activity of executing software from the point of view of finding errors [3]. With the increase in customer and user demands, the size and complexity of the software have started to increase, and it became difficult for the software to be of the desired quality [4]. Before it is launched to the customer, tests must be carried out and necessary actions must be taken according to the results of the tests in order to ensure the quality of the software, reduce the costs of re-correction and development, and increase customer satisfaction. The consequences of launching a faulty product can be enormous, for example, a small mistake in a financial application can result in a huge loss of money for the company, as well as great damage to the company's credibility. For this reason, special attention should be given to the testing stage and enough time should be allocated.

Although testing is so important, the necessary time is not allocated for this stage in software development projects.

This is due to the fact that more time was spent than planned especially in the previous stages of software development processes and tight deadline pressures. Especially in today's internet world, it has become very important to enter the market quickly and to be a pioneer. Managers, who are aware of the importance of these, want to launch the product without being sufficiently tested as the development of the software is completed.

There are many different testing methods that can be used while testing software: Functional tests, Black box tests, White box tests, Gray box tests, Unit tests, Integration tests, Regression tests, End-to-end tests, User acceptance tests, Requirements tests, Validation tests, User interface tests, System tests, Alpha tests, Beta tests, Security tests, Penetration tests, Fuzz test, Brute force attack tests, Performance tests, Load tests, Stress tests, Scalability tests, etc. Some of these are more generic naming, meaning that they include other types of tests. For example, tests such as unit, integration, and user acceptance tests can be grouped under the heading of functional tests. Load, stress, and scalability tests are tests to measure the performance of the software.

In this study, three main test types are used as Hooda and Chhillar [3]. These are functional, performance, and security tests. Functional tests are the tests that verify whether each function of the software is working according to the specified requirements. Performance tests are the tests performed to determine how fast the software performs under a given workload. Security tests, on the other hand, are the tests carried out to detect problems that may occur due to malicious use of the software. Executing different test types require different expertise and knowledge. The methods and tools used for the analysis, creation, and execution of test cases vary. Different test types of software cannot be expected to be performed by a single tester. For this reason, test teams should consist of people with different knowledge and expertise.

The importance of the testing and the tight deadline schedules put pressure on the test teams, and tests are tried to be made without good planning. In addition, the work of test teams is not seen enough by other teams and management. In this study, a system has been developed which will operate in a transparent manner, where a test unit can manage the requests received, make their plans, monitor their processes and their performance criteria, and report easily when necessary. Test units receive test requests from many different teams. The importance values of these requests and the times required to perform these tests are different. With the developed application, the test unit will be able to evaluate the requests received and enter the necessary workload information into the system. In addition, the test type to be performed and the required expertise are different for each request. Due to the limited workforce resources of the test unit, it is not always possible to meet all demands. Also, for some requests, there may be another request that must be completed beforehand. In this study, two different objectives are studied according to the purposes of meeting the largest number of requests and prioritizing the most important requests according to their importance level. The system will create plans according to two different objectives, the requests to be scheduled that the algorithm suggests and the performance criteria calculated according to the requests to be scheduled will be displayed on the application. In addition to the constraints of the Multi-dimensional Knapsack problem, the proposed planning model considers the prerequisite to be considered in the completion and planning of requests. The supervisor or manager of the test unit will preview this information on the application and decide which plan to choose. After the planning periods are over, the determined performance criteria will be traceable on the system. Traceability through the system will provide transparency to requesters, team members, and managers. Today, it is very important not only to do the job but also to do the job efficiently and to be able to report the work done. Especially transparency is becoming more and more important. In this study, the test unit has been considered, but job planning and tracking of processes are also valid concerns for other business units. The proposed system can also be used for other business units.

2. DEFINITION OF THE PROBLEM

A web application and scheduling algorithm are developed by considering the following assumptions and requirements.

- The relevant test unit has different and limited resources for functional, performance, and security tests. The availability of resources may vary for each period, for example, there may be people who will be on leave within that period from the resources who can do the security tests. For this reason, the resource constraint in each period will be adjustable in the application to be developed. It is assumed that the number of resources available for the planning period is known before planning is made.
- Requests will be collected before each planning period. Requests will be created at the application by the project managers. The following information will be requested during the creation of requests:
    - Short Name of the Request
    - Job Description
    - Importance Level
    - Test Type (functional, performance, or security tests)
    - Request Creator (This field will be filled automatically as the person logged into the application.)
    - Request Creation Date (This field will be filled automatically as the date when the request is created.)

- Prerequisite (This field will be selected if there is another request that needs to be completed before the test request.)
- Document if any

- If the request has just been created and no action has been taken, it can be deleted. Requests that have been evaluated by the administrator but not planned can be canceled. A request that has been taken into the planning once cannot be deleted or canceled.
- The request can be deleted or canceled by only the person who created it.
- Planning periods may differ as resource constraints for that period will be dynamic; such as monthly or weekly. However, in this application, the period is considered fixed as two weeks, so that project managers can predict when the planned work will be finished at the latest.
- After the requests are collected, the requests will be evaluated by the test unit manager or supervisor, and if necessary, the manager will be able to reject the request. If the request is approved, the estimated unit time (day, hour, etc.) for the completion of the request will be provided.
- After the evaluation of the requests, new planning can be started by the test unit manager.
- In the new plan screen, the number of resources of different test types for the period will be provided into the system as unit time.
- For some requests, there may be another request that must be completed beforehand. For example, before the performance tests of an application, it may be desirable to test the changes made in the application. If the prerequisite request has been completed in the previous planning, it does not need to be taken into account for this planning, but if it is a request that can be taken into the corresponding planning period, the request will not be included in the planning unless its prerequisite request is included in the planning.
- The test unit manager will run the algorithm for both objective functions through the application after entering the required information. Two different planning results for the purposes of meeting the largest number of requests or prioritizing the most important requests according to their importance level will be proposed according to the number of available resources provided. The test unit manager will be able to select one and start the planning period.
- The following performance criteria will be calculated together with the proposed solution:
  - Percentage of requests met based on the number of requests
  - Percentage of importance level of requests are met on the basis of importance levels
  - Estimated usage time of resources
- After the manager chooses a solution and clicks apply, the relevant planning period will be started. If there is an unfinished planning period, a new planning period cannot be started.
- When the testers enter the application, they will be able to list the requests in the current period, and change the following information on the requests:
  - Remaining Workload
- The continued planning period will be terminated by the test unit manager. After the end of the period, the determined performance criteria will be calculated:
  - Percentage of completion of planned requests based on the number of requests
  - Percentage of completion of planned requests based on importance levels
  - Actual usage time of resources
- Past plans and performance criteria of these plans can be followed through the application.
- Planned but not completed requests will be directly included in the next planning period as far as the unfinished part.

C#, Html, and Javascript are used for web application development. In addition, ASP.NET 6 MVC, Bootstrap, and jQuery frameworks are used. Entity Framework Core is used for Object-Relational Mapping (ORM) and Code First approach is used for database operations. The project structure is created with the Clean Architecture approach. PostgreSQL is used as the database.

3. METHOD

After the evaluation of the incoming requests, the planning algorithm will be run according to the appropriate resource amounts and prerequisites. At first, our planning problem can be thought of as a Knapsack Problem. The classic 0-1 Knapsack problem can be expressed as determining the items to be included in the collection so that the highest possible profit is obtained, and the total weight value is not exceeded, given a set of items, each of which has a weight and utility greater than zero [5]. Different specialists will work for different types of tests in the environment considered. Therefore, our problem can be thought of as the Multi-dimensional Knapsack Problem. The multidimensional knapsack problem will find a subset of the resource consumption such that it satisfies the capacity constraints of each resource and yields maximum profit [6].

The formulation of Multi-dimensional Knapsack problem is as follows:

$$maximize \sum_{i=1}^{n} p_i \, x_i$$

$$subject\ to \sum_{i=1}^{n} w_{ji} \, x_i \leq c_j \quad j = 1,2,\ldots,m$$

$$x_i \in \{0,1\}, \quad i = 1,\ldots,n$$

n : $total\ number\ of\ items$

p_i : $profit/benefit\ of\ item\ i$

w_i : $weight/volume\ of\ item\ i$

m : $total\ number\ of\ different\ types\ of\ resources$

c_j : $capacity\ of\ resource\ j$

$$x_i = \begin{cases} 1 & if\ item\ i\ is\ selected \\ 0 & otherwise \end{cases}$$

In this formulation, the amount of time required to complete the test request can be considered as the weight of the item. If the goal is to schedule the most request, profit will equal 1 for each request. But if the goal is to prioritize the tasks of greatest importance, the profit of each request will be the provided importance level.

The Knapsack problem is an NP-hard (nondeterministic polynomial-hard) problem that selects the optimal set of items that maximizes a goal under resource constraints and is frequently encountered in the real world [7]. This situation creates a problem in terms of solution time when the size of the problem grows, but heuristic approaches are used for this problem as well. The Multi-dimensional Knapsack problem can be applied to many different areas such as container loading [8], freight transport [9], project selection [10, 11], resource allocation [12], and capital budgeting [13]. Our problem can also be thought of as project selection or resource allocation.

In this study, in addition to the constraints of the Multi-dimensional Knapsack problem, we have added the prerequisite condition to be considered in the completion and planning of requests. For some requests, there may be another request that must be completed beforehand. For example, before the performance tests of an application, it may be desirable to test the changes made in the application. If the prerequisite request has been completed in the previous plan periods, it does not need to be taken into account for this period, but if it is a request that can be taken into the corresponding planning period, the request will not be included in the planning unless its prerequisite request is included in the planning. For this condition, the following constraint is added to the formulation.

$$x_i \leq x_j, \quad x_j \ in \ PR_{x_i}$$

PR_{x_i} : $Prerequisite\ list\ of\ item\ i$

With this constraint, our problem differs from the Multi-dimensional Knapsack problem.

3.1. Proposed Approach

In the planning phase, the algorithm is run according to two different objectives for the selection of the requests. The algorithm is based on ranking the requests according to their Contribution (Profit / Weight) values. The calculation of the Contribution value differs according to the objective. When the goal is to meet the largest number of requests, the Contribution value is calculated as 1 / time required to get the test done. When the goal is to meet the highest importance level according to the importance levels of requests, the Contribution value is calculated as the importance level of the request / the time required to get the test done.

The steps of the proposed algorithm are as follows:

1. The planning objective type and the Resource list for the available number of resources for each test type are provided as parameters. The Resource list consists of the test type, available resource amount, and planned resource amount fields. The planned resource amount fields are initially zero for all test types.
2. The Plan Requests list is initialized.
3. Uncompleted requests from the previous plan period are found and added to the Plan Requests list.
4. The sums of remaining workloads according to the test types of these uncompleted requests added to the Plan Requests list are calculated and written to the planned resource amounts columns in the resource list.
5. The list of requests that can be planned, the Can Planned Requests list is initialized.
6. Requests with approved status are found.
7. The for each loop is initialized for requests with approved status.
    a. If requests do not have a prerequisite, then it is added to the Can Planned Requests list by calculating their contribution values.
    b. Else the status of the prerequisite request,
        i. If New, Rejected, or Updated;

            The corresponding request is not included in the Plan Requests list, nor added to the Can Planned Requests list.

        ii. If Completed, Incomplete, or Canceled;

            The contribution value of the request is calculated, and the test request is added to the

Can Planned Requests list by deleting the prerequisite information.

iii. If Approved;

The contribution value of the request is calculated, and the test request is added to the Can Planned Requests list with the prerequisite information.

8. The Can Planned Requests list is divided into two separate lists as those with and without prerequisites: Requests with No Prerequisite and Requests with Prerequisite.
9. Requests with No Prerequisite list are sorted in descending order of their contribution values.
10. The requests in the Requests with the Prerequisite list are sorted by increasing their contribution values.
11. The Requests with No Prerequisite list is copied to a new list, the Suitable Requests list.
12. The for loop iterates over the sorted Requests with Prerequisite list, said request i,
    a. Another for loop iterates over the Suitable Requests list in a reverse manner said, request j.
        i. If the contribution value of the request j is greater than the contribution value of the request i,
        ii. Or if the Id of the request j is equal to the Id of the prerequisite of request i,

            The request i is added to the Suitable Requests list after the request j.
13. The for loop iterates over the Suitable Requests list.
    a. If there is a prerequisite request of the request and this prerequisite is not included in the planning, this request is not included in the planning and the loop is continued. Since the jobs are ordered according to the contribution information and prerequisite requests, it is guaranteed that the prerequisite request is on the list before it, and it is decided whether it is selected or not.
    b. In the resource list, the row which is the test type of the request is found.
    c. If the sum of the time required to complete the request and the planned resource amount of the test type is still less than or equal to the available resource amount of the test type,

        The request is scheduled and added to the Plan Requests list.

        The planned resource quantity of the relevant test type in the resource list is updated.
14. The Plan Requests list is returned.

4. CONCLUSION

In this study, a system has been developed in which a test unit can manage the requests received, make their plans easily and objectively, easily collect and trace the realization information of their planning, and track their processes. Two different purposes were studied in order to make planning. A differentiated Multi-dimensional Knapsack algorithm has been used to determine the requests to be scheduled in the planning. Performance criteria determined can be monitored and the success of the test unit can be traceable easily and transparently by everyone. It is aimed to make the work of the test unit objective and provide transparency. The subject of this study is a test unit, but the developed application and planning algorithm is a system that can be used by other business units in the real world with some minor changes.

In this study, we have considered the prerequisite condition that must be completed beforehand, which is common in the real world. However, our application currently allows only one request to be a prerequisite. As a future job, the system can be modified so that requests can have multiple prerequisites. In addition, in the current system, the number of resources is considered as a total on the basis of test type. That is, even if there are 2 employees who can work on functional tests, we enter a single value for the functional test resource into the system. As a future job, the system can be modified so that individual-based assignments can be made instead of test-type-based resource assignments.

REFERENCES

[1] D. Yıldız, "Bilişim Sistemi Yazılım Geliştirme Test Aktivitelerinde Kritik Başarı Faktörleri". *Journal of Information Systems and Management Research*, 3(2), 15-25, 2021.

[2] Internet: Yazılım Testi, https://tr.wikipedia.org/wiki/Yazılım_testi, 28.08.2022.

[3] I. Hooda, R. S. Chhillar, "Software test process, testing types and techniques", *International Journal of Computer Applications*, 111(13), 2015.

[4] F. Yücalar, E. Borandağ, "Yazılım Projelerinde Kalitenin Arttırılması: TMMi", AURUM Journal of Engineering Systems and Architecture, 3(2), 143-152, 2019.

[5] M. E. Berberler, A. Güler, U. Nuriyev, "A new genetic algorithm for the 0-1 knapsack problem", *Academic Platform-Journal of Engineering and Science*, 4(3), 2016.

[6] M. Abdel-Basset, R. Mohamed, K. M. Sallam, R. K. Chakrabortty, M. J. Ryan, "BSMA: A novel metaheuristic algorithm for multi-dimensional knapsack problems: Method and comprehensive analysis", *Computers & Industrial Engineering*, 159, 107469, 2021.

[7] D. Li, J. Liu, D. Lee, A. Seyedmazloom, G. Kaushik, K. Lee, N. Park, "A Novel Method to Solve Neural Knapsack Problems", **International Conference on Machine Learning**, 6414-6424 July, 2021.

[8] M. Eley, "A bottleneck assignment approach to the multiple container loading problem", *OR Spectrum*, 25(1), 45-60, 2003.

[9] G. I. Sampurno, E. Sugiharti, A. Alamsyah, "Comparison of Dynamic Programming Algorithm and Greedy Algorithm on Integer Knapsack Problem in Freight Transportation", *Scientific Journal of Informatics*, 5(1), 49, 2018.

[10] M. Tavana, K. Khalili-Damghani, A. R. Abtahi, "A fuzzy multidimensional multiple-choice knapsack model for project portfolio selection using an evolutionary algorithm", *Annals of Operations Research*, 206(1), 449-483, 2013.

[11] A. N. Hasoon, "Projects selection in Knapsack problem by using artificial bee colony algorithm", *Tikrit Journal of Pure Science*, 23(2), 173-142, 2018.

[12] F. Aisopos, K. Tserpes, T. Varvarigou, "Resource management in software as a service using the knapsack problem model", *International Journal of Production Economics*, 141(2), 465-477, 2013.

[13] P. C. Pendharkar, J. A. Rodger, J. A. "Information technology capital budgeting using a knapsack problem", *International Transactions in Operational Research*, 13(4), 333-351, 2006.

ICI Computer Science 2022
by Informatics Institute in Gazi University
ONLINE CONFERENCE
December 1-2, 2022

National Qualifications Frameworks and Qualification Databases: Turkish Example

Ekrem YAZICI, Aslıhan TÜFEKCİ

Informatics Institute, Gazi University, Ankara, Turkey
ekremyazici@gmail.com, asli@gazi.edu.tr

***Abstract*—** As developing technologies affect, change, transform and digitize everything, the systematization brought by this technological change is striking in various fields in the education and training community. One of these is the tool called the Qualifications Framework, which is used to manage both national and international information resources. The most important component of this tool is to define the learning outcomes and documents such as diplomas, certificates, course completion certificates, and vocational qualification certificates if they have not been defined yet.

Turkish Qualifications Framework, which covers around 30,000 qualifications in Turkey, has been created and the Turkish Qualifications Database is used as the information management system for this framework. In this study, in which the Turkish Qualifications Framework, which started to work in 2010 and started to include qualifications in the framework as of 2020, is presented, the components, basis, objectives and administrative structure of the Turkish Qualifications Framework are revealed. It is aimed to introduce this tool to the scientific world, which aims to facilitate national and international horizontal and vertical mobility in education and to prepare an easy ground for recognition processes.

***Keywords*—** qualification, qualifications framework, Turkish Qualifications Framework, Turkish Qualifications Database

1. INTRODUCTION

It has always been a valid policy to improve and develop education and training systems and to put the developments of the modern world into practice. States, countries or administrations always try to get the maximum benefit from the innovations brought by the age by adapting the changes they can for the education and training systems within their sphere of influence to the systems, and try to put into effect the practices that will make the society they live in the most qualified [1].

The interaction in the world is spreading to all areas with an increasing trend. Education and training systems have also come under the influence of this spread, and perhaps the approaches that are widely used in all processes of education affect the systems.

Simultaneously, this emerging globalization has increased employee and student mobility while revealing system interactions. The formation of this mobility has led to the necessity of comparing the education and qualification systems between the countries where there is mobility and interaction, and the fact that this information traffic can be managed with information systems, and a sub-title is the need for information systems in education.

As a result of this need, qualification databases have been created in order to manage the qualification frameworks and the information resources that these qualification frameworks concretize. In the study, these two tools and their related components were examined and information about their details was compiled.

2. NATIONAL QUALIFICATIONS FRAMEWORKS (NQF)

2.1. Qualification

In order to have an idea about what the national qualification framework means, first of all, it is necessary to provide clarity on what the term "qualification" should mean. Qualification according to TDK;

- The state of being competent, competence.
- Special knowledge, competence, competence that provides the power to do a job.
- Power to fulfill one's duty, adequacy, ability.

has three different meanings. In the European Qualifications Framework Recommendation [2] revised in 2017, it is “the official result of the evaluation and verification process, obtained if the competent institution decides that the individual has achieved learning outcomes according to certain criteria.”[3] is located as.

From this point of view, the word qualification that expresses the state of being competent in Turkish, like included in the resources of many institutions such as the European Commission, CEDEFOP, OECD, World Bank, and many other institutions. Besides qualification, term used within the scope of the Turkish Qualifications Framework expresses the official document issued by the responsible institution as a result of an assessment and evaluation process.

In this context, when there is an expression with the expression “qualification”, learning outcomes are determined and all documents prepared by the responsible institutions as a result of measurement and evaluation are expressed. This includes all diplomas, certificates, course completion certificates and international documents etc.

2.2. National Qualification Framework

With the European Commission's 2008 European Qualifications Framework (EQF) for Lifelong Learning Recommendation, national qualifications frameworks, which were seen as a little more phenomenal [4], became more functional and official, and with the places they acquired in the systems, with the support of legal regulations in each country, recognized, international they began to become tools that began to be valid.

Based on the expression of qualification, the qualification framework can be evaluated as a platform or a tool where documents are issued to successful individuals at the end of the process by passing through a measurement and evaluation process such as diplomas, certificates, course completion certificates, international documents that meet certain conditions.

Although it is defined by institutions such as ILO, OECD, UNESCO and the like, the definition in the current Terminology Dictionary [5] published by CEDEFOP is as follows.

“It is a tool for the development and classification of qualifications (e.g. at national or sectoral level) according to a set of criteria (e.g. using descriptors) applicable to identified learning outcomes.”

Looking at the definitions; By assuring the quality of qualifications, it can be defined as the tools in which the documents obtained through both formal and non-formal education and informal learning are defined and classified with learning outcomes, mostly to the levels that are evident with the learning outcomes of the formal education levels. Although these tools cannot fulfill all their functions in practice from time to time, they fulfill the function of defining, classifying and comparing qualifications.

2.2.1. Qualification Framework Types

There are various types of Qualification Frameworks according to their scope, region and sector. In addition, it is also possible to create qualification frameworks for only certain level or levels, where there are qualifications belonging to one or more levels. The most common of these are regional qualification frameworks, national qualification frameworks and sectoral qualification frameworks.

2.2.2 Regional Qualifications Frameworks

Regional qualifications frameworks are qualifications frameworks that include qualifications belonging to a particular region. By bringing together the qualifications that are organized on a certain geographical area, it can be considered that the areas with economic mobility also constitute the scope of the qualification framework [6].

2.2.3 National Qualifications Frameworks

National qualifications frameworks, on the other hand, are tools that include defined and nationally accredited qualifications given at certain levels. It shows the interrelationships of qualifications and how individuals can progress from one level to the another level.

2.2.4 Sectoral Qualifications Frameworks

Another type of Qualification Frameworks is sectoral qualifications frameworks. Sectoral qualification frameworks can be regulated for a specific occupational group, business area, qualification group or qualification level/levels.

Frameworks organized in this way are called sectoral qualifications frameworks. The Turkish Higher Education Qualifications Framework (TYYÇ), which was created for Higher Education in Türkiye, is an example of a sectoral framework [7].

2.3. European Qualifications Framework

Regional framework put into effect across Europe “European Qualifications Framework for Lifelong Learning (EQF), which was accepted as a product of the European Union's policy cooperation framework in education and training with the recommendation decision numbered 2008/C 111/01 on 23 April 2008 by the European Commission and Parliament. is”. (European Commission, 2017) The EQF is built on developments in the Copenhagen and Bologna Processes.

The EQF acts as a super-framework to ensure the transparency of qualifications in Europe, to encourage the

transfer of qualifications between different countries, to recognize qualifications and to make comparisons between different NQFs.

EQF, which is accepted as a comparison tool, means that although its name is a framework of qualifications, it does not consist of qualifications; on the contrary, it consists of levels and level descriptors used to decide on common levels for various qualifications in Europe. The basis of the EQF consists of eight levels where the minimum common knowledge, skills and competencies required to be possessed at each level are defined.

3. TURKISH QUALIFICATIONS FRAMEWORK (TQF)

The Vocational Qualifications Authority (MYK) was established with the enactment of Law No. 5544 in 2006 in order to establish and operate the national vocational qualification system in Turkey. With the institution gaining functionality and operating, studies for the preparation of the National Qualifications Framework have also been initiated.

During the preparatory work, meetings of the established working groups and commission, working group meetings, consultation and evaluation activities, workshops and consultation meetings were held. As a result of these studies, a document called TQF document was created and announced to all national and international interlocutors and the public in 2013 with the International Conference.

The regulation on TQF was prepared in cooperation with VQA, MoNE and CoHE, and then the "Regulation on the Procedures and Principles Regarding the Implementation of the Turkish Qualifications Framework" was published in the Official Gazette with the decision of the Council of Ministers [8] Since this date, secondary legislation has been completed and qualifications have been started to be placed in the Turkish Qualifications Framework.

According to the regulation that entered into force, the Turkish Qualifications Framework consists of eight levels, from simple to complex; The simplest level is defined as the first and the most complex level as the eighth level.

It was emphasized that all qualifications offered in Turkey are included in this scope and that information on qualifications defined by TQF components should be made available to the public in the Turkish Qualifications Database. The next section discusses what these components are.

3.1. Components of the Turkish Qualifications Framework

The main starting point of national qualifications frameworks and the component that can be evaluated in the innermost layer is the “Learning Outcomes” approach. It has become more well-known with the widespread use of learning outcomes. Learning outcomes that are better known by the actors within the education and training sector are now in the interest of almost all learners, employees and employers.

Learning outcomes or statements of what a learner is expected to know, being able to do and understand at the end of a learning activity, play an increasingly important role in efforts to improve the quality and relevance of education and training in Europe and around the world [9].

The focus of the “learning outcomes” approach shifts from the duration of the learning and the institution where it takes place, to the knowledge, skills and competencies that have been and should be acquired in the active learning and learning process. Although the “learning outcomes” approach is considered to be a fairly new approach, it is applied for many purposes in many countries and sectors [10].

3.2. Levels and Level Descriptors

Level is a rung used to group qualifications broadly considered equivalent in a hierarchical system, as defined as the level of attainment[11] in formal education and training recognized in a qualification system, a qualification framework, at the level of qualification acquired in education and training, work experience, or non-formal/in-formal learning environments is also defined as the 'reference level'. [12] Turkish Qualifications Framework consists of eight levels from simple to complex. One denotes the simplest level while eight denotes the most complex.

The minimum common knowledge, skills and competence expressions that each level should have are called level descriptors. It is possible to determine how much the qualifications meet the expressions in the level descriptors at which level of the qualifications framework should be. More precisely, the learning outcomes in a qualification are compared with the learning outcomes listed in the level descriptors.

While this may seem like a relatively simple process, practice shows that this is not the case. In the interpretation and application of level descriptors within the framework of qualifications, an approach that is expected to provide benefits to the user rather than a 'mere academic frame of reference' approach is expected. [13]

For this reason, the comparison of qualifications with level descriptors and other documents is not made superficially. While the current level descriptors are referenced with regional frameworks such as the European Qualifications Framework, it is decided whether the National Qualifications Framework and the Regional Qualifications Frameworks are compatible by comparing the qualifications of these regional frameworks. From this point of view, levels and level descriptors are components of primary importance.

3.3. Qualification Types and Qualifications Type Descriptors

Qualification types are grouped according to their minimum common definitions and Qualification Type Designators are presented. With the identification of these types, a sub-level classification of qualification frameworks is formed after the levels [14].

The sine qua non of a national qualifications framework are qualifications. The primary function of the qualifications framework is to classify and compare qualifications and to determine their level. In order to achieve this, first the levels are created, then the qualification types of all the qualifications in the system are determined [15].

Table 1. The qualification types in Turkey

TQF Level	Qualification types awarded in formal education and training system	NVQ	EQF Level
8	*Doctoral Diploma (PhD, proficiency in arts, specialty in medicine, dentistry, pharmacy, veterinary)*	-	*8*
7	*Master's diploma (with thesis) Master's diploma (without thesis)*	-	*7*
6	*Bachelor's diploma*	*Level 6 vocational qualification certificate*	*6*
5	*Associate diploma (General) Associate diploma (Vocational)*	*Level 5 vocational qualification certificate*	*5*
4	*High school diploma High school diploma (VET, fine arts and sports) Mastership certificate Level 4 course completion certificate*	*Level 4 vocational qualification certificate*	*4*
3	*Level 3 course completion certificate*	*Level 3 vocational qualification certificate*	*3*
2	*Level2 adult learning certificate Level 2 course completion certificate*	*Level 2 vocational qualification certificate*	*2*
1	*Literacy certificate*		*1*

With this grouping, an evaluation is made by taking into account all the qualifications within the scope of the qualification framework and the qualification types of all the qualifications included in the education and training systems are determined.

Documents containing the minimum common information of all these qualification types regarding the relevant types are called qualification type specifiers. Through these forms, the qualification type specifier is created by the responsible institution offering the qualification for each qualification type. It is then decided through these forms whether a qualification should be evaluated under this type or not.

3.4. National Qualifications Framework Diagrams

A summary representation of the national qualifications framework; The levels of qualifications are provided by the names of qualifications and the diagrams created by bringing them together in order to understand their relations with each other. An example of the Irish qualifications framework and the Turkish Qualifications framework is shown below.

The innermost digits represent the levels of the 8-level frame. Red areas indicate levels 5-8 and higher education qualifications for these levels. In the yellow part, it shows the names of the Ministry of National Education qualifications between 1-4. In the outermost area, there are qualification types and levels belonging to the Vocational Qualifications Authority.

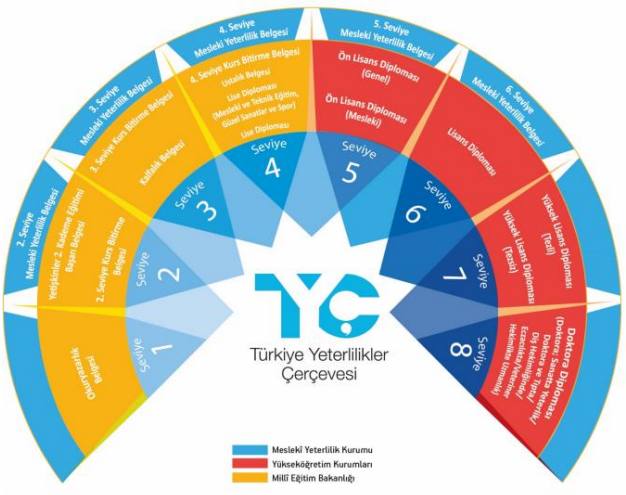


Figure 1: Turkish qualifications framework diagram [16]

3.5. Qualification Databases

The tools in which national qualification frameworks are visualized and can be seen concretely by the end user are qualification framework diagrams. However, since there are only qualification types and the levels of these qualification types in the diagrams and no qualifications can be shown, the concept of national qualifications frameworks and the information resources created by this tool cannot be fully clarified in the minds.

National qualification databases (used as national qualification records in some countries.) are information management systems that are within the scope of national qualifications frameworks and contain qualifications that meet the necessary criteria and can show the qualifications contained in a national qualifications framework. In these systems, many data related to the qualification can be

found, and all the content of the qualification can be compared and filtered in detail [17].

Comparison, which is one of the functions of qualification frameworks, fulfills the benchmarking function in the most effective way, especially when the countries that refer or want to reference the qualifications framework with the European Qualifications Framework, create these databases with the common format determined by the European Commission. National qualification databases also make access to information on qualifications efficient by identifying all qualifications placed in a qualification framework together and with similar definitions.

In many country examples, qualifications are included in qualification databases if they successfully complete all processes after their application for placement in the qualifications framework. If a qualification does not meet the required criteria, it is not included in the qualification database.

Although it is not widely used, as an alternative application like the example of Turkey, all of the qualifications can be included in the qualification database before the placement in order to bring together the qualifications within the scope of the qualification database, and after the placement process is concluded, the qualifications that are entitled to be placed in the qualification framework by successfully completing the process can be displayed separately on the database pages. http://portal.tyc.gov.tr/

With this method, how many qualifications are organized in a country or region, all of them are located on the same platform, and the comparison and comparison of the highest possible number of qualifications with each other can be performed.

Mandatory information that must be found in the databases where these data are also compiled are generally collected under five headings. Apart from this information, all detailed information about qualifications can be defined in these systems and presented to the national and international public.

Name of Qualification

The qualification name written on the qualification must be specified in this field.

Level of Qualification

It is the field where the qualification corresponds to which level within the framework of national qualification. The level of the national qualifications framework can be specified, or if it is associated with a regional framework, this level should also be specified.

Responsible Institution

It is the institution that regulates the qualification, to which the responsibility of the relevant processes belongs. Name of this institution, contact information, website, etc. There is information such as

Learning Outcomes

Learning outcomes, which are the cornerstones of qualification frameworks, are also one of the most necessary information to be found in the pages of qualification databases where information about qualification is given. Under this heading, information about the knowledge, skills and competencies of an individual with the relevant qualification is defined.

Quality Assurance

Information on the quality assurance of qualifications that can be seen as the guarantor of qualifications frameworks is specified in this field, and information on the criteria that a qualification with a quality element should have is given.

Other information

Besides all these important information; Information in the fields of Thematic Area, Country Code, Source of Information, Diploma Supplement, Qualification URL, Qualification Language Entry Requirements, Validity Period, Qualification Requirements are optionally provided in some databases and mandatory in some databases.

On the other hand, by observing the difference between the qualifications that cannot be placed in the framework and the qualifications that are placed, the opportunity to better identify and develop the shortcomings of the qualifications that do not meet all the criteria can be provided.

Qualifications that can be defined with common titles in this way at the national level with the common format of the European Commission can be found in national and international platforms thanks to these data. In platforms such as ESCO, media knowledge, skills and competency statements can be created with the data obtained from these databases. [18]

In this international platform created, as countries integrate their data with this system, they can compare their qualifications with these countries and present relevant information at the international level as well as at the national level. At the date of the research [19], 9653 qualifications, 13890 learning outcomes, 3008 occupational definitions were created with the experts of the participating countries and the data of these countries. [20] It advances this information according to the relevant directive of the European Commission [21].

4. CONCLUSION

It is seen that it creates many sources of information with the documents created by the National Qualifications Frameworks and the Turkish Qualifications Database, which is the information system it reveals. In the absence of such a database, a figure of around 30,000 was not predictable considering all the qualifications from the first level to the eighth level within the scope of TQF. At this stage, all this information becomes observable and manageable with the components of TQF, namely levels, level descriptors, types of qualification types, type specifiers and Turkey Qualifications Database. In this database, which is accessible to all national and international public, information is provided on all topics that the European Commission attaches importance to. The presentation of this information, which is provided by TQF, which is an international common identification and classification tool, with the Turkish Qualifications Database provides convenience in international worker and learner mobility in education, training and business life. In the case of any horizontal-vertical transition, a student in education can easily show what knowledge, skills and competencies a qualification provides, while it can provide information to both the employee and the employer about the talents of the candidate during a recruitment. Considering this information, it can be said that the benefits provided and can be provided by the Turkish Qualifications Database, which constitutes an information management with the priorities of TQF, are undeniable. However, increasing its awareness in terms of the audience it addresses, activating the inoperative and faulty parts of the Turkish Qualifications Database, which is presented as an information management system, and improving the features that can be developed will also increase the efficiency of this tool and the systems connected to the tool.

REFERENCES

[1] OECD, **Education at a Glance 2022: OECD Indicators**, OECD Publishing, Paris, 2022.

[2] European Commission, **Council Recommendation of 22 May 2017 on the European Qualifications Framework for lifelong learning and repealing the Recommendation of the European Parliament and of the Council of 23 April 2008 on the establishment of the European Qualifications Framework for lifelong learning**, Brussels, 2017.

[3] İnternet: Vocational Qualifications Authority Website, https://www.myk.gov.tr/images/articles/TYC/Tyc_bilgi_merkezi/mevzuat_duzenlemeleri/AYC_Tavsiye_Karari_2017_TR_Resmi_Gazete070819.pdf**,** 10.11. 2022.

[4] M. Young, **National Qualifications Frameworks as a Global Phenomenon: A comparative perspective**, Journal of Education and Work, 16:3, 223-237, 2003.

[5] Cedefop, **Terminology of European education and training policy: A selection of 130 key terms,** Publications office of the European Union, Luxembourg, 2014

[6] Etf, Cedefop, Unesco, **Global inventory of regional and national qualifications frameworks 2017: volume I,** Publications office of the European Union, Luxembourg, 2017

[7] Internet: Turkish Higher Education Qualifications Framework, http://tyyc.yok.gov.tr, 10.11.2022.

[8] Internet: Turkish Qualifications Framework Legistlation, https://www.myk.gov.tr/images/articles/mevzuat/TYC_Yonetmeligi.pdf, 10.11.2022.

[9] Cedefop, **Defining, writing and applying learning outcomes: a European handbook,** Publications office of the European Union, Luxembourg, 2017.

[10] Cedefop, **Using Learning Outcomes,** Publications office of the European Union, Luxembourg, 2011.

[11] Cedefop, **Terminology of European education and training policy: A selection of 130 key terms,** Publications office of the European Union, Luxembourg, 2014.

[12] R. Tuck, **An Introductory Guide to National Qualifications Frameworks: Conceptual and Practical Issues for Policy Makers,** International Labour Office. Skills and Employability Department, Switzerland, 2007.

[13] M. Cosser, **Towards the Development of Level Descriptors in the NQF: A SAQA discussion document,** South Africa, 2000.

[14] K. James, C. Borhene, **Level-setting and recognition of learning outcomes: the use of level descriptors in the twenty-first century,** UNESCO, 2015.

[15] Cedefop, **National qualifications framework developments in European countries,** Luxembourg: Publications Office. Cedefop research report, 2018.

[16] Internet: Turkish Qualifications Framework, http://www.tyc.gov.tr/, 11.11.2022.

[17] European Commission, **Council Recommendation of 22 May 2017 on the European Qualifications Framework for lifelong learning and repealing the Recommendation of the European Parliament and of the Council of 23 April 2008 on the establishment of the European Qualifications Framework for lifelong learning Annex-IV**, Brussels, 2017.

[18] European Commission, **A common format for Qualifications in EQF databases and in ESCO,** Luxembourg: Publications Office, 2015.

[19] Internet: ESCO Portal, https://ec.europa.eu/esco/portal/, 12.11.2022.

[20] European Commission, **ESCO Handbook: European skills, competences, qualifications and occupations,** Publications Office, 2017.

[21] Internet: ESCO Portal, https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32013L0037&from=EN , 12.11.2022.


GAZİ ÜNİVERSİTESİ
Gazi
1926