# MAPiS 2019 - First MAP-i Seminar

# Proceedings

Edited by:
Rui Rua, Vanessa Silva, Shamsuddeen Muhammad, Fernando Duarte

January 31, 2019

Aveiro, Portugal

Universidade do Minho universidade de aveiro U.PORTO UNIVERSIDADE DO PORTO MAPi DOCTORAL PROGRAMME IN COMPUTER SCIENCE

MAP-i: Doctoral Programme in Computer Science
MAP-i: Programa Doutoral em Engenharia Informática

# MAPiS 2019 - First MAP-i Seminar

# Proceedings

Edited by:
Rui Rua, Vanessa Silva, Shamsuddeen Muhammad, Fernando Duarte

January 31, 2019

Aveiro, Portugal

MAP-i: Doctoral Programme in Computer Science
MAP-i: Programa Doutoral em Engenharia Informática

**Title:** MAPiS 2019 - First MAP-i Seminar, Proceedings
**Editors:** Rui Rua, Vanessa Silva, Shamsuddeen Muhammad, Fernando Duarte
**Patrons:** MAP-i: Doctoral Programme in Computer Science
**Publisher:** UA Editora, Universidade de Aveiro
**ISBN:** 978-972-789-590-8

# Welcome Message

This book contains a selection of Informatics papers accepted for presentation and discussion at "MAPiS 2019 - First MAP-i Seminar", held in Aveiro, Portugal, January 31, 2019. MAPiS is the first conference organized by the MAP-i first year students, in the context of the Seminar course. The MAP-i Doctoral Programme in Computer Science is a joint Doctoral Programme in Computer Science of the University of Minho, the University of Aveiro and the University of Porto. This programme aims to form highly-qualified professionals, fostering their capacity and knowledge to the research area.

This Conference was organized by the first grade students attending the Seminar Course. The aim of the course was to introduce concepts which are complementary to scientific and technological education, but fundamental to both completing a PhD successfully and entailing a career on scientific research. The students had contact with the typical procedures and difficulties of organizing and participate in such a complex event. These students were in charge of the organization and management of all the aspects of the event, such as the accommodation of participants or revision of the papers. The works presented in the Conference and the papers submitted were also developed by these students, fomenting their enthusiasm regarding the investigation in the Informatics area.

MAPiS 2019 intends to be the first of many others, and in its first edition, the topics focused in several areas of Computer Science and Engineering. Several of these topics were influenced by the most actual and hot topics boarded by the main Research Institutes of the three Universities that compose the joint Doctoral Programme and the scientific community.

MAPiS 2019 featured two Keynote talks by:

- Liliana Ferreira- Director of the Fraunhofer Research Center for Assistive Information and Communication Solutions (AICOS), Faculty of Engineering of the University of Porto, Portugal
- Pedro Miguel Neves- R&D at Altice Labs in Lisbon Aveiro, Portugal

MAPiS featured 4 special sessions, composed by four slots in the Conference Program. The sessions followed logic associations between the topics of the works presented. The first session was composed by 2 paper presentations under the topic of 'Model and Simulation'. The second session was composed by 3 paper presentations under the topic of 'Deep Learning and Machine Learning'. The third session was composed by 4 paper presentations under the topic of 'Networks and Software'. The last session was composed by 3 paper presentations under the topic of 'Big Data Management'.

We would like to thank all organizers for their hard work inviting all the academic community, organizing the papers review process, and helping to promote the MAPiS 2019 Conference. This acknowledgment goes especially to the members of the Program Committee and Reviewers for the hard work required to prepare this volume as they were crucial for ensuring the high scientific quality of the event and to all the authors and delegates whose research work and participation made this event a success. The work of both the Local Organizing Committee and the Technical Program Committee was also crucial to produce the MAPiS Conference Program and this book.

January 2019

Rui Rua Vanessa Silva Shamsuddeen Muhammad Fernando Duarte

# MAPiS 2019 Committees

## Organization Committee

| | | |
|---|---|---|
| General Chair | Daniel Bradan | University of Porto |
| Program Chairs | Fernando Duarte | University of Aveiro |
| | Vanessa Silva | University of Porto |
| | Shamsuddeen Muhammad | University of Porto |
| Publication Chairs | Fernando Duarte | University of Aveiro |
| | Rui Rua | University of Minho |
| Sponsorship Chair | Amir Rastegarlari | University of Porto |
| Awards Chairs | Francisco Ribeiro | University of Minho |
| | José Macedo | University of Minho |
| Local Arrangement/Finance Chairs | José Duarte | University of Aveiro |
| | Nuno Simões | University of Aveiro |
| Publicity Chairs | Vanessa Silva | University of Porto |
| | Akilu Rilwan Muhammad | University of Porto |
| Social Media/Communication Chair | Rui Rua | University of Minho |
| Webmaster | Mohammadreza Kasaei | University of Aveiro |

## Steering Committee

| | |
|---|---|
| Antonio Teixeira | University of Aveiro, DETI/IEETA |
| Nuno Lau | University of Aveiro, DETI/IEETA |
| Joaquim Arnaldo Martins | University of Aveiro, DETI/IEETA |

## Program Committee

### Senior Committee

| | |
|---|---|
| Alexandre Madeira | University of Minho |
| Ali Shoker | HASLab, INESC TEC, University of Minho |
| Alberto Simões | 2Ai Lab – IPCA |
| André Zúquete | University of Aveiro, DETI/IEETA |
| Antonio Teixeira | University of Aveiro,DETI/IEETA |
| Joaquim Arnaldo Martins | University of Aveiro,DETI/IEETA |
| João Mendes Moreira | University of Porto |
| João Gama | University of Porto |
| Luis Barbosa | University of Minho |
| Manuel Filipe Santos | University of Minho |
| Manuel Barbosa | HASLab – INESC TEC and FCUP |
| Manuel A. Martins | University of Aveiro |
| Mário Rodrigues | ESTGA/IEETA – University |
| Nuno Lau | University of Aveiro,DETI/IEETA |
| Pedro Brandão | University of Porto |
| Rolando Martins | FCUP/CRACS-InescTec |

### Junior Committee

| | | | |
|---|---|---|---|
| Amir Rastegarlari | University of Porto | Francisco Ribeiro | University of Minho |
| Akilu Rilwan Muhammad | University of Porto | Nuno Simões | University of Aveiro |
| Daniel Bradan | University of Porto | Rui Rua | University of Minho |
| José Duarte | University of Aveiro | Mohammadreza Kasaei | University of Aveiro |
| José Macedo | University of Minho | Shamsuddeen Muhammad | University of Porto |
| Fernando Duarte | University of Aveiro | Vanessa Silva | University of Porto |

# Table of Contents

# NFV Management and Orchestration: Analysis of OSM and ONAP

Nuno Simões
*Department of Computer Science*
*Faculty of Sciences of the University of Porto*
Porto, Portugal
up201801786@fc.up.pt

***Abstract*—Today, the new generation networks associated to the concept of Software Defined Network (SDN) and Network Functions Virtualization (NFV) is very relevant, particularly for telecom operators. In this paper the topic to be considered is the orchestration of the SDN/NFV networks, that is, Management and Orchestration (MANO). Taking into account the attention given to MANO, several aspects of this segment of NFV networks were analyzed. The MANO theme is analyzed in this article in the form of analysis of two solutions associated to this theme. The focus of the paper is an introductory analysis to ONAP and OSM, as an experiment and introductory paper to OSM and ONAP. Thus, two MANO solutions were analyzed: one with many followers, the OSM, and another more recent, but with plenty of potential and with a large community of participants and, consequently, users of its solution - ONAP. We show that ONAP is a much more complex tool, but, in turn, much more complete than the OSM, because it addresses other points that the OSM does not reach. Nevertheless, both solutions have their advantages and disadvantages.**

***Index Terms*—NFV, SDN, MANO, ONAP, OSM, Orchestrator**

## I. Introduction

In order to fully understand the main theme of the article, it is important to introduce the SDN / NFV concepts. From the various topics related to telecommunications computer networks, some of the most popular are Software-Defined Networking (SDN) and Network Functions Virtualization (NFV). Regarding SDN, where there used to be only hardware and network protocols, there is now software that allows drivers to run. About NFV, this allows highly optimized packet processing of network functions. Both concepts allow the control and orchestration of architectures and are used to create, manage and scale new on-demand service platforms. These last described functions can be made fast and agilely [1]. The creation of an L3VPN service (Virtual Private Network over the network layer of the OSI model) is an example of the use of SDN/NFV[1].

The SDN had as starting point the academia and has been developed by researchers and architects of data centers. The NFV was created from a consortium of Service Providers. The concepts of SDN/NFV are related to the management of software networks and creation of network functions in a more agile and fast way. Thus, the operators, above all, can more quickly meet the pretensions of the customers. The concept of SDN is based mainly on two concepts: Data plane and control plane. This way a control layer is added to the network management [2]. Comparing one concept with another, while the SDN aims to centralize network management, the NFV focuses on virtualizing network functions. This virtualisation aims to reduce physical equipment whenever possible, with its replacement being done for example by virtual machines (VM) [3].

Characteristics and themes associated with these two concepts began to emerge. Some of them are the virtualization of networks, the virtualization of some services, SDN controllers, NFV management and orchestration (MANO) (that will be detailed in the next section), reduction of physical space and reducing electrical energy [3].

The purpose of this paper is to analyze two MANO solutions: OSM and ONAP. This paper try to be an introduction to the theme of orchestration in NFV and, more specifically, an introduction to the two referred tools. Both tools come to try to help in the optimization and automation of tasks with regard to the management of networks through software. In particular, the tools can deal with the orchestration, policing and automation of physical and virtual infrastructures of computer networks.

The remainder of this paper follows the following organization. In the next section it will be presented the state of the art with regarding NFV Management and Orchestration (MANO). In the third section, two examples of MANO (the OSM, Open Source MANO, and ONAP) are presented, as well as a small comparison between each one. In the final section we will conclude the analysis carried out, as well as the presentation of some comparative points between both MANO solution.

## II. State of the art of NFV Management and Orchestration

As the meaning of the acronym MANO indicates, the NFV Management Orchestration is the NFV manager. NFV MANO is a working group of the *European Telecommunications Standards Institute Industry Specification Group*, commonly known as ETSI ISG NFV. NFV MANO manages and orchestrates all the resources in the cloud data center. This includes computing, networking, storage, and virtual machine resources

[1] *ATT and Ericsson presenting SDN-based L3VPN solution for Telco NFV needs*: https://cloudblog.ericsson.com/digital-services/att-and-ericsson-presenting-sdn-based-l3vpn-solution-for-telco-nfv-needs

[4]. MANO is a critical point in the proper functioning of NFV Infrastructure (NFVI) and Virtual Network Functions (VNFs) and provides the requisite requirements for provisioning and configuration of VNFs [5].

The NFV MANO is divided into three functional blocks [4] [5]:

- NFV Orchestrator (NFVO): combines several functions to create end-to-end services;
- VNF Manager (VNFM): is responsible for the life cycle of network virtualization functions (VNF). Can manage only one or multiple instances of VNFs;
- Virtualized Infrastructure Manager (VIM): manages and controls NFV physical and virtual infrastructure resources across a single domain.

Along to the blocks, NFV MANO has four types of data repositories. These repositories are databases that have several NFV MANO information. The repositories are [5]:

- Network Services (NS) Catalog: are a set of predefined templates that indicate how services should be created and developed;
- VNF Catalog (VNFM): it is a set of templates that describe characteristics of development and operationalization of VNFs;
- NFV instances: are where all the information related to functions and services is stored;
- NFVI resources: are where all information related to NFVI is stored.

The use of the NFV concept foresees that its application can be gradual and with the existing equipment.

MANO is mostly used in aspects related to virtualization mechanisms, while network management is more directed to network services composed of VNFs and physical nodes. This network management also includes Operation System Support (OSS) and Business System Support (BSS), among others.

In Figure 1 we can observe the general architecture of NFV MANO [6]. In this Figure we can see the functional blocks and repositories already mentioned in this paper. With this image we can get a more visual idea of the MANO [6]. The EMS is the Element Management System.

To conclude the section, are presented some projects related to NFV MANO.

These examples can be [5]:

- CloudNFV: is an open platform for NFV implementation. This solution, based on cloud computing and SDN, is composed of three elements: Active virtualization, which is a data model; NFV Orchestrator, which has policing rules, for example; NFV management, which uses Management Information Bases (MIBs) that is a database to manage entities in a network communication, using Simple Network Management Protocol (SNMP);
- OpenMANO: is an open source project, led by the spanish operator *Telefónica* and aims to implement a MANO framework. The architecture of this solution is based on three main components: openmano, based on REST and that serves for the management of VNF; openvim, which is an implementation of VIM; GUI, which is the component of the Graphical User Interface.

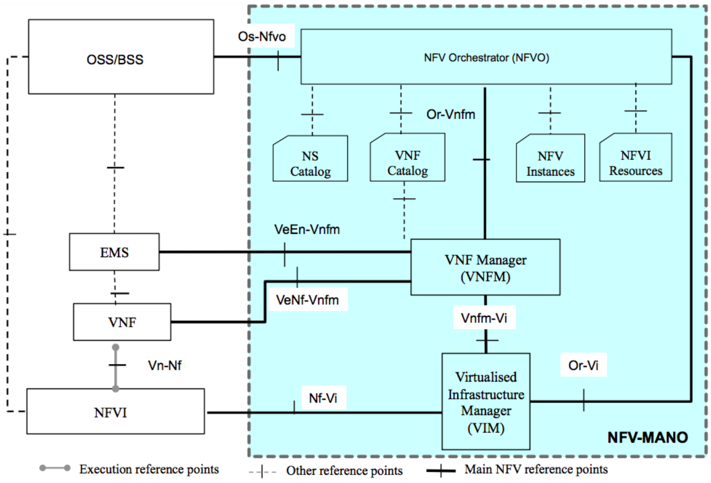


Fig. 1. NFV MANO architecture [6].

Further to these two, there are also others such as ExperiaSphere, Zoom, OPNFV [5], Open Orchestrator Project (OPEN-O), Tacker, Cloudify and Open Baton [7].

This concludes the section on the state of the art.

## III. Open Source MANO (OSM) and Open Networking Automation Platform (ONAP)

After the presentation of the adjacent concept, the main theme of the paper is presented. In this section, a presentation of the OSM and then of ONAP will be made.

### A. *Open Source MANO (OSM)*

Open Source MANO, or OSM, is mostly developed in Python and runs on Linux operating systems [8]. The OSM was released in 2016 [9]. The most recent version, that has been released to date, is 5 [10]. It is important to note that OSM 5 already includes references to network slice and orchestration, two very popular themes nowadays regarding to SDN/NFV [11].

The OSM uses three functional blocks (NFVO, VNFM and VIM) of the NFV MANO to perform the configuration and abstraction of VNFs, and orchestration and management of the infrastructure [12].

Some of the operations supported by the OSM are [12]:

- An orchestration service for VNF;
- The possibility of performing complex services;
- Support for different VIMs;
- Support for SDN controllers;
- Support for monitoring tools.

The OSM proposal is to map the ETSI ISG NFV architecture into an Open Source implementation available on the OSM website itself [10]. In the Figure 2, the mentioned OSM mapping can be consulted.

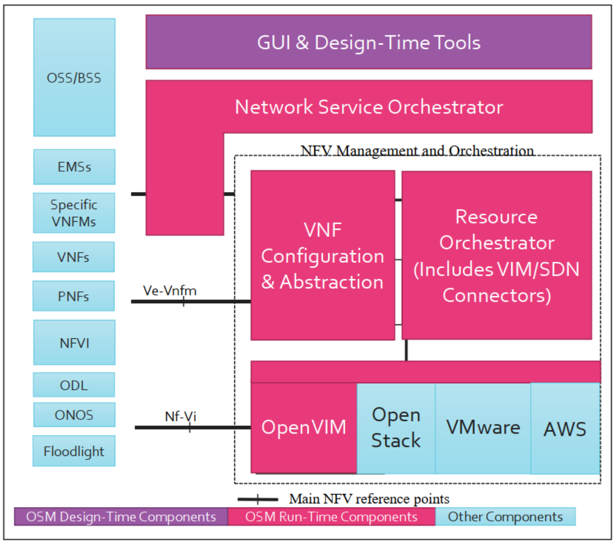

GUI & Design-Time Tools
OSS/BSS
Network Service Orchestrator
NFV Management and Orchestration
EMSs
Specific VNFMs
VNF Configuration & Abstraction
Resource Orchestrator (Includes VIM/SDN Connectors)
VNFs
PNFs
Ve-Vnfm
NFVI
ODL
ONOS
Floodlight
Nf-Vi
OpenVIM
Open Stack
VMware
AWS
Main NFV reference points
OSM Design-Time Components
OSM Run-Time Components
Other Components

Fig. 2. Mapping from OSM to ETSI NFV MANO [13].

The use of the OSM can have several objectives. In addition to providing management and orchestration components to the NFV, the use of the OSM can also be made to facilitate the management, development and design of VNFs and network services. Another goal may be the evolution of the reference architecture proposed by ETSI ISG NFV. Thus, the use of OSM can become massive [8]. Besides this, it can also grow with the implementation of 5G. The OSM is one of the MANO quite associated with 5G [9].

Regarding the OSM requirements and installation, this solution has some features. According to some references in the documentation, OSM already uses the container concept via docker. This is because the website[2] references a *dockerized* installation of OSM. In addition, since version 4, OSM has been targeting the cloud-native strand, like so many other technologies nowadays. Still since version 4 they have a lighter orchestrator and a new GUI.

On the installation requirements, according to the documentation[2], only one machine, or VM, with the characteristics shown in table I are required.:

TABLE I
REQUIREMENTS FOR OSM INSTALLATION

| | **Minimum** | **Recommended** |
|---|---|---|
| **CPU** | 2 | 2 |
| **RAM (GB)** | 4 | 8 |
| **Storage (GB)** | 20 | 40 |

In terms of operating system, Ubuntu 16.04 is referred to. Now about the installation, it appears to be quite simple, with all the steps being properly explained on the website[2]. There is also a reference to the components that we want to include in the installation.

After the installation is complete, the web interface login page can be verified. Once the installation is complete, 10 docker containers are installed.

In addition to the installation on a machine or VM, it is also described how to configure OpenStack, VMware vCLoud or Amazon Web Services to use OSM. In the end, a simple VNF can be started according to the steps described on the website[2].

Giving the presentation of the OSM is finished, ONAP will be presented in the next subsection.

### B. *Open Networking Automation Platform (ONAP)*

According to ONAP's own documentation, its solution comes to close a failure that existed at the level of telecom operators, cable and cloud. According to ONAP, there was a lack of a common platform, which offered different types of services and could also be competitive and compensatory [14].

ONAP is a solution with orchestration capabilities, which can be applied to both real and virtual elements. The ONAP solution is modular and supports YANG and TOSCA[3] data models. It is important to notice that YANG and TOSCA are data modeling language. It is stated in the document [14] that ONAP can be integrated with several VIMs, VNFMs, SDN Controllers and even devices that may already be somewhat obsolete. This type of integration can lead to financial savings and reuse of existing material.

Design-Time consists of 2 subsystems[4]:

- Service Design and Creation (SDC);
- Policy.

The SDC has 4 main components[5]:

- Catalog: is where data arrives using Design Studio;
- Design Studio: is used to create, modify, and add Resource and Service definitions to Catalog;
- Certification Studio: is the test and experiment point of SDC for future releases;
- Distribution Studio: is the intermediate point between the Certification Studio and the placement of assets into production. This component is not explicit, but it is represented in ”Recipe/Eng. Rules Policy Distribution”.

Regarding the architecture of ONAP, it is represented in a high level perspective in Figure 3. Note that in the Figure 3 the catalog is what makes the connection between Design-Time, which allows the reuse of service models already developed for example, and Run-Time, where the rules and policies coming from Design-Time are executed.

The ONAP runtime is where the ONAP Policy, Command, or Request, and Inventory components are located [14].

[2] *OSM Release FOUR*: https://osm.etsi.org/wikipub/index.php/OSM_Release_FOUR

[3] *TOSCA vs. Netconf a Comparison*: https://www.sdxcentral.com/nfv/definitions/tosca-vs-netconf-comparison/

[4] *ONAP Architecture*: https://wiki.onap.org/display/DW/Architecture

[5] *Service Design and Creation (SDC)*: https://wiki.onap.org/pages/viewpage.action?pageId=1015837

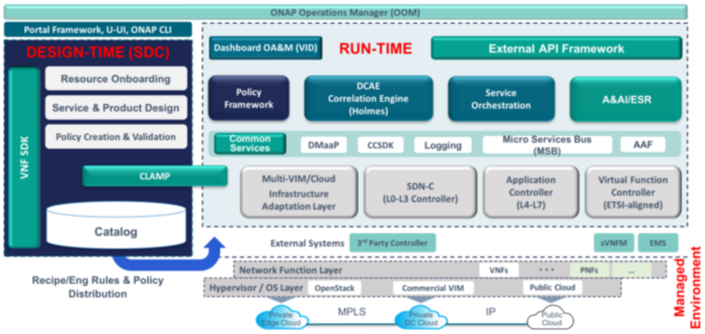


Fig. 3. ONAP high-level architecture [14].

On ONAP deployment, there are three possibilities for doing this[6] component by component, via Heat (OpenStack) or via Kubernetes/ONAP Operations Manager (OOM)[7]. Depending on the chosen option, other platforms may be used. If we choose Kubernetes we can use[8]: Amazon EC2, Cloudify, Google Cloud Engine, among others. In the following table are presented the requirements of each of the options[6]. It is important to note that ONAP allows each component to be installed independently. For example, the Service Orchestrator or Policy Framework can be installed as an isolated component. So there is no need to install all the components to run/test just one.

TABLE II
RESOURCES NEEDED TO INSTALL ONAP

| | Kubernetes | Heat | Single component |
|---|---|---|---|
| **VM** | 4 | 20 | 1 |
| **vCPU** | 32 | 88 | 1 |
| **RAM (GB)** | 128 | 176 | 2 |
| **Storage (GB)** | 160 | 1760 | 20 |

Note that the requirements presented in the Table II are for the Beijing version of ONAP. For the Casablanca version of ONAP the requirements in terms of infrastructure are less onerous[9]. About the versions of ONAP, the developers expect to release of two versions per year[10]. For the moment, the Casablanca version is already in use, but still at a very early stage[11].

The ONAP automation cycle can be seen in Figure. 4.

ONAP is based on a unified architecture [15]. Basically, this cycle can have the following path: Design > Create > Collect > Analyze > Detect > Publish > Respond [14].

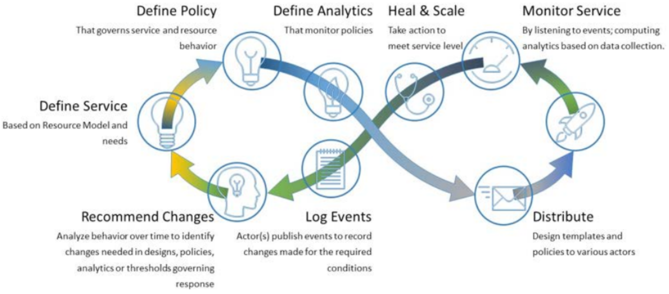
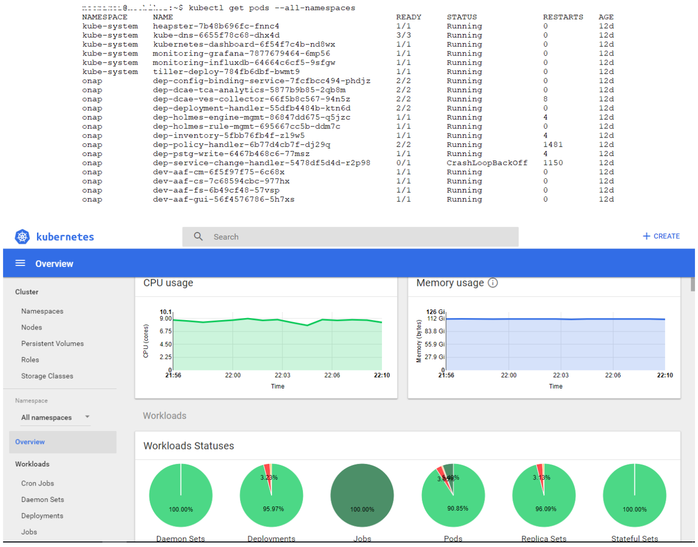

Define Policy
That governs service and resource behavior
Define Analytics
That monitor policies
Heal & Scale
Take action to meet service level
Monitor Service
By listening to events; computing analytics based on data collection.
Define Service
Based on Resource Model and needs
Recommend Changes
Analyze behavior over time to identify changes needed in designs, policies, analytics or thresholds governing response
Log Events
Actor(s) publish events to record changes made for the required conditions
Distribute
Design templates and policies to various actors

Fig. 4. ONAP Automation Cycle [14].

To conclude this section, in Figure 5 is presented an illustrative image of the execution of ONAP in a physical infrastructure composed by 2 servers. Initially, only one server was envisaged but, due to the limited number of pods (components) of ONAP[12], another machine was added. Both machines are running Ubuntu 16.04. One has 126GB of RAM and 366GB of storage and the other 39GB of RAM and 133GB of storage. At the top of Figure 5 is a list of some running pods. At the bottom of the same Figure is the portal of Kubernetes at Rancher. Rancher, that is a cluster Kubernetes management, is used since ONAP is instantiated on Kubernetes.



```
[redacted]:~$ kubectl get pods --all-namespaces
NAMESPACE     NAME                                         READY   STATUS             RESTARTS   AGE
kube-system   heapster-7b48b696fc-fnnc4                    1/1     Running            0          12d
kube-system   kube-dns-6655f78c68-dhx4d                    3/3     Running            0          12d
kube-system   kubernetes-dashboard-6f54f7c4b-nd8wx         1/1     Running            0          12d
kube-system   monitoring-grafana-7877679464-6mp56          1/1     Running            0          12d
kube-system   monitoring-influxdb-64664c6cf5-9sfgw         1/1     Running            0          12d
kube-system   tiller-deploy-784fb6dbf-bwmt9                1/1     Running            0          12d
onap          dep-config-binding-service-7fcfbcc494-phdjz  2/2     Running            0          12d
onap          dep-dcae-tca-analytics-5877b9b85-2qb8m       2/2     Running            0          12d
onap          dep-dcae-ves-collector-66f5b8c567-94n5z      2/2     Running            8          12d
onap          dep-deployment-handler-55dfb4484b-ktn6d      2/2     Running            0          12d
onap          dep-holmes-engine-mgmt-86847dd675-q5jzc      1/1     Running            4          12d
onap          dep-holmes-rule-mgmt-695667cc5b-ddm7c        1/1     Running            0          12d
onap          dep-inventory-5fbb76fb4f-zl9w5               1/1     Running            4          12d
onap          dep-policy-handler-6b77d4cb7f-dj29q          2/2     Running            1481       12d
onap          dep-pstg-write-6467b468c6-77msz              1/1     Running            4          12d
onap          dep-service-change-handler-5478df5d4d-r2p98  0/1     CrashLoopBackOff   1150       12d
onap          dev-aaf-cm-6f5f97f75-6c68x                   1/1     Running            0          12d
onap          dev-aaf-cs-7c68594cbc-977hx                  1/1     Running            0          12d
onap          dev-aaf-fs-6b49cf48-57vsp                    1/1     Running            0          12d
onap          dev-aaf-gui-56f4576786-5h7xs                 1/1     Running            0          12d
```

Fig. 5. Above, querying the list of installed pods and, in the bottom, the Kubernetes GUI.

The ONAP instantiation was not very simple. To deploy ONAP, OOM was used. The script used for the software requirements is what is presented on the website[13]. A change was made to the Rancher installation, i.e., the commands

[6] *Setting Up ONAP*: https://onap.readthedocs.io/en/beijing/guides/onap-developer/settingup/index.html

[7] *ONAP Operations Manager Project*: https://wiki.onap.org/display/DW/ONAP+Operations+Manager+Project

[8] *ONAP on Kubernetes*: https://wiki.onap.org/display/DW/ONAP+on+Kubernetes

[9] *Setting Up ONAP*: https://onap.readthedocs.io/en/casablanca/guides/onap-developer/settingup/index.html

[10] *Release Calendar*: https://wiki.onap.org/display/DW/Release+Calendar

[11] *Casablanca Maintenance Release*: https://wiki.onap.org/display/DW/Release+Planning#ReleasePlanning-CasablancaMaintenanceRelease

[12] *ONAP on Kubernetes on Google Compute Engine*: https://wiki.onap.org/display/DW/ONAP+on+Kubernetes+on+Google+Compute+Engine

[13] *oom_rancher_setup.sh*: https://github.com/onap/logging-analytics/blob/master/deploy/rancher/oom_rancher_setup.sh

followed were similar to those presented on the website itself [16].

Unfortunately, I did not have the opportunity to test the OSM.

This concludes the section on MANO solutions.

## IV. Conclusion and Future Work

Very briefly, it is concluded that ONAP touches all points of Management & Control, while OSM only refers to the Orchestration, Management & Policy topic of Management & Control according to the Linux Foundation. This can be seen in Figure 6.

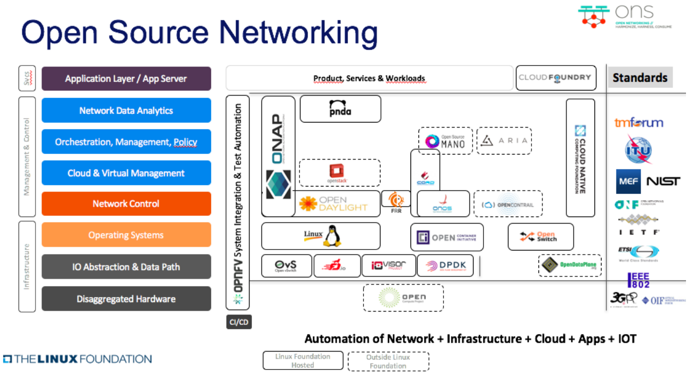


Fig. 6. Open Source Networking[14].

From our experience, the installation of ONAP is not complicated. The only experimental factor made for this article was the instantiation of ONAP in a physical infrastructure. Thus, the ONAP conclusion is based only on this fact. The hard part may be finding the way, or the right directions, to do so. In our view, one of the main problems of ONAP, compared to OSM, are the requirements. As can be seen in the previous section, the requirements of ONAP are much higher than the characteristics of the infrastructure where this solution can be based. From this point of view, the OSM is more advantageous because it does not need so many resources, as indicated in the website[2].

Another important issue is documentation. In terms of documentation, both have plenty. The point is that although we have been more concerned about ONAP, it seemed to us that the OSM documentation was better structured. ONAP has plenty documentation, but it seems to be spread across several websites. It did not seem to us that it was present in a place and that it was well structured. This factor is also important during the development and use of any technological solution. But one of the reasons may be that ONAP is a much more complex tool than OSM.

It is important to note that there are many other Open Source solutions competing with OSM and ONAP. Nevertheless, these others have received less attention.

In terms of future work, and extending the subject of MANO to solutions other than OSM and ONAP, other possibilities could also be analyzed. It would also be interesting to present performance results between the different solutions. Thus, it could be verified that there would be a more efficient solution compared to others. An analysis to management orchestration taking in attention the 5G technology will be an interesting thematic to analyze.

## References

[1] S. Van Rossem, W. Tavernier, B. Sonkoly, D. Colle, J. Czentye, M. Pickavet, and P. Demeester, "Deploying elastic routing capability in an SDN/NFV-enabled environment," in *Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on*. IEEE, 2015, pp. 22–24.

[2] V. Wijekoon, T. Dananjaya, P. Kariyawasam, S. Iddamalgoda, and A. Pasqual, "High performance flow matching architecture for openflow data plane," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on*. IEEE, 2016, pp. 186–191.

[3] B. B. Westcon, "What is the difference between SDN and NFV?" available at https://blogbrasil.westcon.com/qual-a-diferenca-entre-sdn-e-nfv.

[4] S. Central, "What is nfv mano?" available at https://www.sdxcentral.com/nfv/definitions/nfv-mano/.

[5] R. Mijumbi, J. Serrat, J.-L. Gorricho, S. Latré, M. Charalambides, and D. Lopez, "Management and Orchestration Challenges in Network Functions Virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, 2016.

[6] M. Ersue, "ETSI NFV Management and Orchestration - An Overview," in *Proc. of 88th IETF meeting*, 2013.

[7] C. Parada, J. Bonnet, E. Fotopoulou, A. Zafeiropoulos, E. Kapassa, M. Touloupou, D. Kyriazis, R. Vilalta, R. Muñoz, R. Casellas *et al.*, "5GTango: A Beyond-Mano Service Platform," in *2018 European Conference on Networks and Communications (EuCNC)*. IEEE, 2018, pp. 26–30.

[8] A. S. Saito and F. L. Verdi, "High availability support management in nfv environments using osm."

[9] M. Bhagwat, D. Clarke, P. Eardley, A. E. Armengol, G. de Blas, P. Gronsund, A. Hoban, S. Manning, T. Nakamura, F. J. Ramn Salguero, and A. Reid, "OSM: Experience with NFV Architecture, Interfaces and Information Models," *Open Source MANO, Technical Overview*, 2018.

[10] OSM, *Open Source MANO*, OSM, available at https://osm.etsi.org/.

[11] OSM, *OSM PoC 1: DevOps in Service Chains and 5G Network Slices*, OSM, available at https://osm.etsi.org/wikipub/index.php/OSM_PoC_1_-_DevOps_in_Service_Chains_and_5G_Network_Slices.

[12] G. Venâncio, V. F. Garcia, L. da Cruz Marcuzzo, T. N. Tavares, M. F. Franco, L. Bondan, A. E. Schaeffer-Filho, C. R. P. dos Santos, L. Z. Granville, and E. P. Duarte Jr, "Simplifying Lifecycle Management of Network Virtualized Functions."

[13] A. Israel, A. Hoban, A. Sepúlveda, F. Salguero, G. de Blas, K. Kashalkar, M. Ceppi, M. Shuttleworth, M. Harper, M. Marchetti *et al.*, "OSM release three," *Open Source MANO, Technical Overview*, 2017.

[14] ONAP, *ONAP Architecture Overview*, ONAP, available at https://www.onap.org/wp-content/uploads/sites/20/2018/06/ONAP_CaseSolution_Architecture_0618FNL.pdf.

[15] F. Slim, F. Guillemin, A. Gravey, and Y. Hadjadj-Aoul, "Towards a dynamic adaptive placement of virtual network functions under onap," in *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on*. IEEE, 2017, pp. 210–215.

[16] Rancher, *Rancher: Single Node Install*, Rancher, available at https://rancher.com/docs/rancher/v2.x/en/installation/single-node/.

[14] *How ONAP Will Merge Millions of Lines of Code from ECOMP and Open-O*: https://www.sdxcentral.com/articles/news/onap-will-merge-millions-lines-code-ecomp-open-o/2017/04/

# Review of Recent Work on Computational Intelligence in Games

Fernando Fradique Duarte
University of Aveiro
Aveiro, Portugal
fjosefradique@ua.pt

***Abstract*—Games provide an ideal environment for the development and testing of new techniques and technologies particularly in the field of Computational Intelligence. Techniques developed for game-playing are often transferred to other domains such as Psychology and Education further enhancing the scope of their use. Furthermore there is a significant commercial interest in the development of human-like game AI agents. This paper presents a review of the recent work on Computational Intelligence in Games focused primarily on the competitions hosted at the conference on Computational Intelligence and Games. The review provides background on each of these competitions and presents the most recent related work. Four of these competitions, namely the Fighting Game AI competition, the Ms. Pac-Man Vs. Ghost Team competition, the Hearthstone AI competition and the StarCraft AI competition are further reviewed in this regard and a brief summary of their evolution over time is also presented.**

***Keywords—Computational Intelligence, Games, Artificial Intelligence, Machine Learning***

## I. Introduction

Games have been used in scientific research for quite some time particularly in the field of Computational Intelligence (CI) but also in other fields such as Psychology, Sociology and Education [1]. Early research in Artificial Intelligence (AI) used mostly classical two-player board games such as chess [2]. Recently video games have begun to attract such equal attention. One of the reasons video games are interesting in terms of research and particularly CI an AI stems from the wide range of challenges they pose with each genre of game posing its own set of challenges. As an example in Hearthstone a Collectible card game (CCG) the AI agent must deal with hidden information and uncertainty. As another example in StarCraft a Real Time Strategy (RTS) game the AI agent must be able to perform both micro-management tasks (e.g. control units in real-time) as well as macro-management tasks (e.g. plan a higher-level strategy).

Another reason of interest in terms of research as to do with the fact that the techniques developed for game playing can be used in other fields of research such as Education, Robotics and Brain-Computer Interfaces (BCIs). Also, there is significant commercial interest in the development of more sophisticated game AIs that act in a more natural way, adapting themselves to the changes occurring in the environment similarly to the way human players do [1].

Proof of this growing interest are the various conferences that have emerged over the years hosting several competitions related to research on CI and AI in video games such as the conference on Computational Intelligence on Games (CIG) and  the conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE). This paper presents a review of the recent work on Computational Intelligence in Games focused primarily on the competitions hosted at the CIG conference. The review provides background on each of these competitions and presents the most recent related work. Four of these, namely the Fighting Game AI competition, the Ms. Pac-Man Vs. Ghost Team competition, the Hearthstone AI competition and the StarCraft AI competition are further reviewed in this regard and a brief summary of their evolution over time is also presented. These 4 competitions were chosen because of one or more of the following reasons: the fact that they are well established, relevant information (e.g. detailed competition results) was readily available, the base games are popular or have commercial versions and ultimately because they span several different game genres. The remainder of this document presents this review.

## II. Fighting Game AI Competition

The objective of the Fighting Game AI [3] competition is to promote the research and development of general fighting game AIs that are able to play against various different opponents (other AIs or human players) in any play mode using any character data. In the remainder of this section further background information about the competition is provided as well as a brief overview of the most recent related work. A summary regarding the evolution of the competition over time is presented at the end of the section.

### A. Background

Fighting games are a challenging game genre as they require the player to decide on which actions to perform from a set of possible actions in a very short interval of time. In FightingICE, the fighting game platform used for this competition there are 56 possible actions that the player can choose from within a required response time of 16.67 ms.

The FightingICE platform offers an environment for research that allows the design of flexible AIs. The setting of FightingICE takes place on a spatially limited two-dimensional stage. The 2 players or fighters (either human players or programmed AI agents) can move and perform attack and defense actions. Three different game characters can be used, each with its own set of unique skills (different effects achieved) and different requirements that must be met in order to perform these skills (for one of these characters, namely LUD, this data is unknown in advance).

Besides the character data just mentioned the AI agents also have access to the so called frame data containing information about the characters positions and their health points for example. This frame data however is given to the AI agents with a 15 frame delay instead of the current data in order to simulate the reaction delay of human players (this delay constraint can be circumvented for Visual-Based AIs). Agents must perform their actions through simulated key-inputs (imitating the input of a human player). By performing certain sequences of these actions within a given period of time, agents can perform combos to deliver additional damage to the opponent as well as combo-breakers (abort the opponent's combo).

There are 2 leagues in this competition, namely the Standard League and the Speedrunning League. In the Standard League the winner of a round between 2 AIs is the one with Health Points (HP) above zero at the time its opponent's HP has reached zero. In the Speedrunning League the competing AIs fight against a provided sample AI. The winner is the AI that can beat the sample AI in the shortest average time computed over 10 matches.

### B. *Related Work*

Various different approaches have been proposed in order to deal with the challenges inherit to this competition. This section discusses some of this most recent work.

In [4] the authors propose a Hierarchical Task Network (HTN) as a planner in order to create sequences of actions or plans. This approach allows the AI agent to make decisions that can take into account long-term goals and high-level strategies allowing the creation of longer plans (i.e. plan sequences of actions further in advance) as opposed to just single actions (i.e. choose the optimal action for the current game state). By generating several alternative such plans the agent can react more suitably to the constant changes occurring in the environment.

The work presented in [5] proposes 4 Dynamic Difficulty Adjustment (DDA) AI agents implementing Monte Carlo Tree Search (MCTS) in order to tailor the difficulty of the game according to the player's skill in real-time and throughout game play as the skill level of the player progresses. In contrast to the more traditional approach were players are required to choose a difficulty level at the beginning of the game which is then kept unchanged until the end, these agents can dynamically change the strategies and behaviors of the opponent AIs or even the environment to better suit the current skill level of the player. This in turn allows the game to be more enjoyable and long-lived as it promotes a better immersion of the player.

Acknowledging the popularity of the MCTS approach in this competition and its limitations, mostly due to response time constraints (i.e. the opponent's behavior is predicted by taking into account only a few randomly selected actions (5) from the set of all possible actions on the opponent's side), the authors in [6] propose the use of an Action Table (AT) in order to encode the opponent's playing patterns and use this knowledge to predict the opponent's playing actions. This means that the AI agent can incorporate the observed playing patterns of its opponent into its decision process in order to further improve its performance against its opponent.

The authors of [7] use a Deep Convolutional Neural Network (CNN) in order to predict the actions of the opponent. This CNN however is trained on non-visual information (i.e. features) such as energy points and position and size of the characters as visual information (i.e. images of the game) is not provided to the AI agents in this competition. Several experiments were performed in order to find the best way to arrange and group these features as well as to compare the results obtained by this CNN with those obtained by a simple Neural Network (NN).

Genetic Programming (GP) is used in [8] in order to help automate the creation of the character fighting strategies. This allows the creation of a wider diversity of AI agents with different strategic skills. These in turn can be leveraged to incorporate more adaptive behavior into the game making it more engaging to the human player. As an example of this, this process can enable the creation of more interesting and realistic AIs (i.e. non-determinist) then those obtained using more established industry techniques such as manually coded Finite State Machines (FSMs) (i.e. deterministic).

In [9] the authors propose to model the opponent using the Neuro-evolution of Augmenting Topologies (NEAT) algorithm. The basic architecture of this algorithm is an Artificial Neural Network (ANN). This ANN is trained using the data collected from the opponent's actions and later used in order to predict the probability associated to each of the movements that the opponent can make so that the AI agent can determine the best countermeasure according to each situation. A Genetic Algorithm (GA) is used to optimize (i.e. change) the architecture of the ANN to better model the opponent over time.

In [10] the authors propose a GA in order to discover combos (sequences of attacks allowing a player to damage the opponent while preventing the opponent from performing any action). While combos are an important feature in many fighting games acting as a rewarding system to the precise execution of commands and thus encouraging the player to continue playing the game and improving his skills, they can also result in unexpected and undesirable behaviors such as the occurrence of long or infinite combos for example which can ruin the gaming experience. Results demonstrated that this approach was able to find combos under various different configurations, some of which (combos) were not known to be possible within the fighting game platform used to perform the tests.

### C. *Evolution*

The Fighting Game AI competition started in 2013. In this first edition almost all of the participants (9) used rule-based AIs as their main approach. This approach (rule-based) continued to be the most popular amongst the participants on the next 2 editions in 2014 (6 entries for the 1 character competition) and 2015 (along with FSMs) although various other approaches began to emerge such as Opponent Modeling, GA and Fuzzy Logic. This scenario changed a bit in 2016 with the emergence of a new approach based on the combination of Rule-based algorithms with MCTS (6 entries against 5 for pure Rule-based). This approach dominated the top 3 ranking. Ever since then (also in 2017 and 2018) MCTS combined with several other techniques such as GA and Q-Learning has been the dominant approach used and as dominated the top ranking in terms of results. A brief summary of this evolution is depicted in Table I, showing the most used approach (MUA) as well as the winning approach (WA) and the number of participants (#E) for each year.

TABLE I. SUMMARY OF THE EVOLUTION OF THE COMPETITION IN TERMS OF THE APPROACHES USED OVER TIME

| Year | MUA | WA | #E |
|---|---|---|---|
| 2013 | Rule-based | FSM | 10 |
| 2014 | Rule-based | Dynamic Scripting | 10 |
| 2015 | Rule-based | Rule-based | 17 |
| 2016 | Rule-based + MCTS | Rule-based + MCTS | 13 |
| 2017 | Rule-based + MCTS | MCTS | 9 |
| 2018 | MCTS | MCTS | 7 |

## III. Ms. Pac-Man Vs Ghost Team Competition

The goal of the Ms. Pac-Man Vs. Ghost Team [11] competition is twofold: on one hand the competition aims to promote research on cooperation between agents acting in a fairly complex environment in order to achieve a mutual goal (capture Ms. Pac-Man). On the other hand due to the adversarial nature of the Pac-Man game the competition also promotes research on strong AI agents that can get good results in the game (stay alive and score as much points as possible). In the remainder of this section further background information about the competition is provided as well as a brief overview of the most recent related work. A summary regarding the evolution of the competition over time is presented at the end of the section.

### *A. Background*

The Ms. Pac-Man Vs Ghost Team competition started in 2016 and is a revival of the 2 previous competitions also based on the Ms. Pac-Man arcade game, namely the Ms. Pac-Man Screen Capture competition and the Ms. Pac-Man Vs Ghosts competition. An updated game engine, the introduction of Partial Observability (PO) constraints in the game and a new multi-agent approach to develop the ghost agents are some of the improvements introduced [12]. PO is a technique used to impair the ability of the player or agent to completely observe its environment.

Ms. Pac-Man, the arcade game on which the competition is based consists of 5 agents, Ms. Pac-Man and 4 ghosts interacting in a 2D maze environment. Explained in a very simplistic way the ghosts must try to capture Ms. Pac-Man while Ms. Pac-Man collects as much of the pills scattered on the corridors as possible in order to obtain a higher score. From time to time a reversal event occurs (e.g. Ms. Pac-Man eats a power pill) during which the ghosts change into the frightened mode and are instead chased and can be eaten by Ms. Pac-Man.

The competition is composed by 2 tracks. The first track concerns the development of a strong AI agent for Ms. Pac-Man that can operate under a PO constraint and score as many points as possible (stay alive, collect pills and eat as many ghosts as possible). The second track concerns the development of the AI agents for the 4 ghosts. These agents also operate under a PO constraint and must cooperate and try to coordinate their actions in order to prevent Ms. Pac-Man from consuming too much pills and ultimately trap and capture it.

### *B. Related Work*

Various different approaches have been proposed in order to deal with the challenges inherit to this competition. This section discusses some of this most recent work.

In [13] the authors propose an approach to create varying versions of an agent with different playing styles and skills that behave in a designer-specified fashion. In order to achieve this, the authors propose the use of a Neural Network to model the agent. A variant of a generational GA with two-point crossover is then used to evolve these NNs in order to find the most suitable ones. These agents can be used for a variety of purposes such as to automatically collect game metrics, as opponents in multi-player games or to help the developers balancing the games' mechanics by acting as proxy human players with arbitrary skill levels.

The work presented in [14] proposes an approach to evolve a diverse and versatile set of cooperating agents that are able to adapt to the players' skill level using GP. The learning process is based on a combination of cooperative and adversarial coevolution, whereby the Pac-Man agent (1 population) competes against the Ghost Team, composed of 4 cooperatively evolving populations. This adversarial learning scheme enables both types of agents to be evolved simultaneously which in turn somewhat simulates the learning process of an actual human player (playing with the Pac-Man agent).

The authors in [15] propose Case-based Reasoning (CBR) and Reinforcement Learning (RL) techniques to train the agents. This training process is achieved via the use of the Q-learning algorithm. In this case however the Q-table (table used to store the maximum expected future reward for each action at each state) is replaced by a case base (collection of past experiences). The use of cases allows the injection of domain knowledge into the learning process (retrieve similar problems in the case base and adapt their solution to the current context) while also enabling a richer representation of the game state.

Recognizing the importance of human-like agents in order to improve the gaming experience (e.g. challenge or collaborate with the human player to help him achieve a goal or get started with the game) the authors in [16] conduct a study in order to try to understand if it is possible to distinguish between a human player and an AI agent and if so what are the features that best characterize how a players' behavior may be perceived as human-like or not. This study was conducted using 3 AI agents (a strong AI with good performance in the game, a simplified version of the strong AI and an AI presenting a totally randomized behavior) and 5 human players with different experience and skills over the course of 17 recorded games that were later presented to human judges so that these could deliberate whether the specific player was a human or not.

PO is used in various game genres such as in horror games to build suspense for example (sudden appearances of other agents). Used wrongly however, PO can induce negative emotional responses from the player such as anxiety, fear and frustration while playing the game. Given this in [17] the authors conduct some experiments in order to investigate the effect of varying levels of PO on difficulty and enjoyment in a Pac-Man game. AI agents are used in order to assess the effects on difficulty while human players are used to investigate the effects on enjoyment.

Finally in [1] the authors present a summary of the research conducted over the years using the Pac-Man game and variants thereof. The study focuses mainly on research conducted on the field of CI and presents the various approaches used by the participants in this competition and its predecessors over the years. This overview also highlights other fields of research where Pac-Man was used such as in Biology, Psychology, Robotics and Education.

### C. Evolution

Competitions based on the Ms. Pac-Man arcade game started on 2007 with the Ms. Pac-Man Screen Capture competition (which ran until 2011). Later and building on the success of the Ms. Pac-Man Screen Capture competition the Ms. Pac-Man Vs Ghosts competition was launched and ran for four iterations between 2011 and 2012. In 2016 the Ms. Pac-Man Vs Ghost Team competition was launched featuring some changes relatively to the previous competitions such as an updated game engine and the introduction of PO. The introduction of these changes enriched the competition with a new set of challenges [1].

Over the course of these 3 competitions several different approaches were proposed such as [1] Rule-based, FSM, Tree Search, Monte Carlo (MC), Evolutionary Algorithms (EA), Neural Networks, Neuro-evolution and Reinforcement Learning. In [1] the authors review the literature on these approaches: Rule-based and FSMs can be found as early as 2003 mostly between 2008 and 2012 and still to a lesser degree in 2016 and 2017, Tree Search and MC can be found as early as 2009 mostly between 2010 and 2013 and to a lesser extent in 2014, EAs can be found as early as 1992 and mostly between 2010 and 2013 and also in 2014 and 2016, NNs can be found as early as 1999 and until 2010, Neuro-evolution can be found as early as 2005 and mostly in 2011, 2014 and 2016, lastly RL can be found as early as 2009 and mostly in 2015 and 2016 to a lesser degree. A brief summary of the results obtained in 2018 is depicted in Table II. This summary includes an overview of the winning approaches (WA), considering the top performing agents for both the Ms. Pac-Man controller (team P) and the Ghosts controllers (team G) as well as the number of participants (#E) for each track. It should be noted that very few entrants (only 4) participated on the second track of the competition, 2 of which were the default controllers provided, namely StarterGhostComm and StarterGhost.

TABLE II. SUMMARY OF THE COMPETITION RESULTS FOR 2018

| Year | Team | WA | #E |
|---|---|---|---|
| 2018 | P | Modular Multi-objective (Hyper) NEAT | 8 |
| | G | Rule-based | 4 |

## IV. HEARTHSTONE AI COMPETITION

The objective of the Hearthstone AI [18] competition is to promote the development of fully autonomous AI agents that are able to play in gaming environments featuring uncertainty and hidden information such as in the context of the Hearthstone game. In the remainder of this section further background information about the competition is provided as well as a brief overview of the most recent related work. A summary regarding the evolution of the competition over time is presented at the end of the section.

### A. Background

CCGs are a popular game genre. This game genre is also interesting for AI research due to the fact that players must deal with hidden information (the cards of the opponent are unknown) and uncertainty stemming from the vast number of possible combinations of states, rules and cards that may result in playing scenarios not even anticipated by the creators of the game.

In Hearthstone, the turn-based card video game used as the base platform for this competition, 2 players play against each other using pre-constructed decks of 30 cards and a selected hero with a unique power (draw a card, summon a minion, heal or deal damage). These cards differ for each hero (e.g. the Mage class offers more spells). Players use their limited mana crystals to draw cards in order to attack the opponent (e.g. cast spells or summon minions). The goal of the game is to reduce the opponents HP to zero.

The competition is composed by 2 tracks, namely the Premade Deck Playing track (PMD) and the User Created Deck Playing track (UCD). In the PMD track all participants receive a list of decks and playout all possible combinations against each other. The winner is determined by the average win rate. This track encourages research on agents that can use their own characteristics and the opponent’s deck to win the game. The UCD track allows agents to define their own deck. This track encourages research on finding a deck that can consistently beat a vast amount of other decks and also on optimizing the agent’s strategy according to the characteristics of their deck. Again the average win rate dictates the winner.

### B. Related Work

Various different approaches have been proposed in order to deal with the challenges inherit to this competition. This section discusses some of this most recent work.

In [19] the authors propose a modification to the MCTS algorithm in order to handle randomness and tackle imperfect information. The authors also propose a heuristic (board solver) to tackle the combinatorial complexity of the game (due to the large number of possible attacks and the varying order in which the attacks may be performed). This heuristic generates a sequence of attacks given a game state. Finally and due to the weaknesses of MCTS when dealing with huge branching factors and delayed rewards resulting from the actions taken, the MCTS algorithm is combined with a value network heuristic, specifically a Deep Neural Network that given a state of the game computes the predictions of the game outcome. MCTS uses these predictions to foresee the outcome of a playout without having to simulate it until the end.

An important element of player engagement in card games is the periodical addition of new cards as they potentially provide new gaming strategies. Playtesting is the process used to check new card sets for design flaws The authors of [20] propose an Evolutionary Algorithm (EA) to automate this playtesting process (deckbuilding). The EA creates new card decks which are then played by an AI agent against human-designed decks in order to evaluate their effectiveness. The authors also propose a new heuristic mutation operator to the EA based on the way human players modify their decks in order to limit the space of possible decks.

In the work presented in [21] the authors propose an AI agent based on an Expert System (ES). A symbolic approach with a semantic structure acts as an ontology to represent the static descriptions of the game mechanics and the dynamic game state memories (representation of the current state and the actions that led to it). The amount of expert knowledge represented in the ontology such as popular moves and strategies is reduced as these should be derived by the agent using its knowledge base (static knowledge representing generic information about the game and dynamic knowledge describing the entities currently active on a game session). At runtime the agent uses rules and performs queries on the semantic structure to do reasoning and strategic planning.

In [22] the authors propose an ensemble of various NN models (including CNN) to predict the likelihood of the first player (assuming it is his turn to play) in winning the game given the representation of an arbitrary intra-game state. The datasets used were extracted from a collection of playouts between weak AI agents. The proposed method requires minimal domain knowledge and uses basic feature preprocessing to extract the minion, hero and aggregated features. For the NN models this set of features was further extended to include as additional features the square and logarithm of all features (except minion features and hero class).

The work in [23] proposes a Stacking Generalization (SG) model (various machine learning algorithms stacked) with 2 layers to predict which of 2 AI agents playing against each other will win the game based on the information known at the given time. A Bayesian approach was used in order to optimize the hyper-parameters of the several algorithms used. The final model proposed consists of a conditional model composed of 2 SG models. The first SG model was well optimized and fine-tuned and is used when the test data is Independent and Identically Distributed (IID) (no new cards present). When this is not the case (non-IID scenario) a second SG model, more conservative (not as fine-tuned) is used instead.

Finally in [24] the authors propose an AI agent based on a modified version of the MCTS algorithm that integrates expert knowledge of 2 types in its search process (choose adequate moves). The first type of domain specific knowledge consists of a database of decks that is used to handle imperfect information (the cards that the opponent holds are not known). The second type consists of a heuristic function used to guide the MCTS rollout phase (simulation of the game until a terminal state is reached) in order to reduce the search space of the game (possible moves). Two heuristics, constructed as linear combinations of the features extracted from the given state of the game were tested. The first heuristic included a small number of hand-picked features while the second included additional features.

### C. Evolution

The Hearthstone AI competition started quite recently in 2018. Regarding the PMD track the top performing agents used simulation-search based algorithms such as MCTS or trained an evaluation function using EA. Concerning the UCD track the top performing agents used several different approaches including MCTS and Greedy EA.

TABLE III. SUMMARY OF THE RESULTS FOR 2018

| Year | Track | WA | #E |
|---|---|---|---|
| 2018 | PMD | MCTS, EA | 33 |
| | UCD | MCTS, Greedy EA | 17 |

A brief summary of the results obtained during the 2018 competition is depicted in Table III. This summary includes an overview of the winning approaches (WA), considering the top performing agents as well as the number of participants (#E) for each track.

## V. StarCraft AI Competition

The objective of the StarCraft AI [25] competition is to promote research on RTS game AI agents that are able to perform under uncertainty, manage resources and plan high-level strategies. In the remainder of this section further background information about the competition is provided as well as a brief overview of the most recent related work. A summary regarding the evolution of the competition over time is presented at the end of the section.

### A. Background

RTS games are a challenging game genre as they require the player to handle resource collection (to produce units and buildings), manage the construction of units and buildings (i.e. choose build order) and battle against enemies (control a large number of units in real time). These tasks are often referred to as micro-management tasks (unit control) and macro-management tasks (higher-level game strategy of the player). Such scenarios pose challenges to AI due to their dynamic nature (uncertainty), their huge state-action spaces and the need for both short and long decision making [26].

StarCraft, the base platform used in the competition is a RTS game featuring a strategic military combat simulation. Each player controls 1 of 3 races and must gather resources to expand their base and produce more units (an army). The winner of the game is the player that manages to destroy his opponent’s base.

The competition is organized into a single track, where the participant agents play against each other (1 vs 1 game) in a round-robin tournament. Games are limited to simulate 1 hour of gameplay. The agent with the greatest win percentage over all the rounds played is the overall winner of the competition.

### B. Related Work

Various different approaches have been proposed in order to deal with the challenges inherit to this competition. This section discusses some of this most recent work.

In [27] the authors propose Continual Online Evolutionary Planning (COEP), an evolutionary-based method capable of performing in-game (during the game) adaptive build order planning and optimization. COEP controls the macro-management tasks of the game (i.e. what builds to produce and in which order) allowing the agent to change its build order dynamically to quickly adapt to the opponent’s strategy. Four mutation operators, namely: clone (build at position *a* becomes the same as the build at position *b*), swap (2 builds swap positions in the build order), add (a build is inserted in the build order along with its requirements-other builds) and remove (a build is moved to the end of the build order) are implemented and used to effectively reorganize pre-existing build orders.

The work in [28] proposes Deep Learning to learn the macro-management tasks directly from game replays performed by highly skilled human players. In order to achieve this, replay files were processed and all events related to macro-management tasks such as material changes were extracted and used to simulate abstract StarCraft games via a build order forward model. The resulting action-state pairs obtained from the actions performed during these abstract games were then used in order to build the training dataset. A fully connected Neural Network was then trained on this dataset and used as the macro-management module of the AI agent.

In [29] the authors conduct a study over 5 algorithms used by AI agents playing StarCraft for resource gathering (choose resource locations in order to maximize the total amount of resources gathered). The algorithms tested are: Built-in (assign unit to go to the previous resource location), Mineral-lock (evenly distribute worker units over all resource locations), Queue-based Scheduling (attach queues to resource locations and designate free worker units to queues), Co-operative pathfinding (optimize the paths of the worker units to their designated resource locations), and Co-operative pathfinding + Queue (a combination of the previous 2 algorithms). The authors conclude that a trade-off between CPU time and resource gathering rate must be made (in general the algorithms that are more CPU intensive can gather more resources).

In [30] the author presents a review of CUNYbot, one of the entrants of the StarCraft: Brood War AI tournament. CUNYbot makes strategic decisions using a low-dimensional economic model (traditionally used to describe the behavior of countries). The parameters of this economic model are optimized between games via a GA algorithm in order to learn the capital/labor ratio for each of the built-in AI races. The bot also implements a reactive strategy based on the Cobb-Douglas model in order to model its opponents during the game and adapt its own strategy accordingly.

The work presented in [26] implements an agent able to pursue long-term plans that may require long sequences of actions to be achieved mimicking the usual behavior of human players in RTS games. In order to achieve this, the game state space is partitioned into a set of clusters (states with similar features) or abstract states. An option or a temporally-extended action in a Markov Decision Process (MDP) is then created for every abstract state and algorithm in the portfolio of game-playing algorithms (such algorithms receive the current state and output an action). The learning agent observes the abstract state and selects an option which acts according to its associated algorithm. The new state is observed as well as the reward received and the agent selects a new option repeating the learning process.

Finally in [31] the authors introduce the Deep RTS environment, an high-performance RTS game (and simulator) created specifically for AI research. Deep RTS supports accelerated learning (50,000 times faster compared to existing RTS games) and features a flexible configuration that enables research in several different RTS scenarios including partially observable state-spaces and map complexity. Deep RTS targets Deep Reinforcement Learning (DRL) research and aims to ease its use in more advanced games (e.g. ease the design of reward models).

### C. Evolution

The StarCraft AI competition started in 2010. In terms of the most popular techniques used it is difficult to choose 1 given the plethora of diverse approaches that have been used by the participants such as HTN, Breath First Search (BFS) pathfinding, FSM, Greedy search, GA, MCTS, A*, NN (e.g. Long-short Term Memory (LSTM)) to name just a few. An interesting fact is the large increase on the number of bots using file I/O in order to adapt their strategies (file I/O is allowed and agents can save experience from the rounds they play and use that information to change their strategies for the next rounds) from 4 (2017) to 19 (2018). The number of bots using Machine Learning (ML) techniques for this same purpose (i.e. adapt their strategy) also increased a bit from 2 (2017) to 7 (2018). This may be an indication that participants are trying to devise agents that are more adaptive to the changes occurring in the environment.

A brief summary of the evolution of the competition in terms of the winning approaches (WA), considering the top 3 ranked entrants (and the information available) and the number of participants (#E) for each year is depicted in Table IV (previous years to 2013 are not considered).

TABLE IV. SUMMARY OF THE EVOLUTION OF THE COMPETITION IN TERMS OF THE WINNING APPROACHES OVER TIME SINCE 2013

| Year | WA | #E |
|---|---|---|
| 2013 | Greedy Search | 8 |
| 2014 | FSM, Potential Flows | 13 |
| 2015 | FSM, Script-based | 16 |
| 2016 | LSTM, A* with Depth-first Search (DFS) | 16 |
| 2017 | Multi-agent, HTN | 20 |
| 2018 | HTN, BFS | 27 |

## VI. OTHER COMPETITIONS

This section presents an overview of the remaining competitions hosted at CIG. Due to space constraints this discussion is more high level. Nevertheless some background information about each of these competitions as well as their research challenges and most recent related work are provided.

### A. Short Video Competition

The goal of this competition is to act as a source of interesting videos showcasing CI. The videos are presented in a plenary session and the winners are decided by the vote of the audience.

### B. MicroRTS Competition

The goal of the microRTS [32] competition is to motivate research underlying the development of AI agents for RTS games while minimizing the amount of engineering required to participate so that participants can focus their efforts on the research aspects of the competition. Contrary to the StarCraft AI competition, in microRTS agents have access to a simulator (forward model) which they can use to simulate the effect of actions and plans, allowing planning techniques to be developed more easily. The competition is organized into 3 tracks: large state spaces and branching factors, partial observability and non-determinism. Examples of recent related work include several variants of MCTS [33]–[36], RL [37] and Evolutionary Multi-Objective optimization [38].

### C. Hanabi Competition

The goal of the Hanabi [39] competition is to motivate research on AI agents capable of playing the cooperative partially observable card game Hanabi. The Hanabi card game is played by 2 to 5 players using a deck of 5 suits (colors) of cards. Players cannot see their own cards but they can see the other player's cards. The goal of the game is to play each suit in rank order (1 to 5). The competition is organized into the Mixed track (agents are paired with a group of unknown agents) and the Mirror track (agents are paired with copies of themselves). The agent that achieves the highest score over a set of unknown deck orderings wins the competition. Examples of recent related work include the use of GA [40], Rule-based [41] and a mixture of Rule-based with MCTS [42], [43].

### D. General Video Game AI Competition

The goal of the General Video Game AI [44] competition is twofold: on one-hand promote research on the development of general video game playing controllers, that is a single AI agent capable of playing any game it is given without knowing beforehand which games are to be played and without using a simulator (forward model) for training. On the other hand also promote research on general video game content generation algorithms such as to generate levels for any game or playing rules for any level [45], [46]. The competition is organized in 4 tracks: Single Player Planning, 2-Player Planning, Level Generation and Rule Generation. Examples of recent related work include the use of Deep Reinforcement Learning [47], [48], MCTS [49], a mixture of Rule-based with MCTS [50] and Rolling Horizon Evolutionary methods [51].

### E. Angry Birds Level Generation Competition

The goal of the Angry Birds Level Generation [52] competition is to promote research on building computer programs that are able to automatically generate fun and challenging levels for the Angry Birds physics-based puzzle game. The objective of the Angry Birds game is to kill all the pigs using the birds provided. In order to achieve this, the player uses a slingshot to shoot birds at block structures (and destroy them) with pigs placed within and around these structures. The levels generated should also be stable concerning gravity, robust in terms of the objectives of the game (a single action should not destroy large parts of the generated structure) and challenging enough while still being solvable. The game level generators are evaluated on the overall enjoyment of the levels they create. Examples of recent related work include the use of Procedural Level Generation algorithms [53], [54] and Pattern-Struct with Preset-Model [55].

### F. Text-Based Adventure AI Competition

The goal of the Text-Based Adventure AI [56] competition is to promote research on AI agents that can play games with text-only interfaces. This competition can also potentially foster new developments in several research fields such as Natural Language Processing (NLP) and Automatic Model Acquisition. The Z-Machine, a text-based game engine is used in order to evaluate the developed AI agents. Agents are scored according to their score while playing an unseen game instance (the dominant criterion) and freedom from a priori bias. Examples of recent related work include the use of Language Models [57].

### G. Visual Doom AI Competition

The goal of the Visual Doom AI [58] competition is to promote research on AI agents that can play Doom, a First Person Shooter (FPS) game, using solely the screen buffer (pixels) information to base their decisions upon. Although agents can be developed by using any technique, Machine Learning methods such as DRL are encouraged (also well supported by ViZDoom the Doom-based AI research platform used in the competition). The competition is organized into 2 tracks: single player (finish the Doom game) and multiplayer (compete with other agents in Doom deathmatches). Examples of recent related work include the use of DRL [59], AutoEncoders [60] and Reinforcement Learning with Curriculum learning [61].

## Conclusion

This paper presented a review on the recent advances on Computational Intelligence in Games. Special focus was given to the competitions hosted at the CIG conference. These competitions were discussed in terms of their goals and the challenges they pose to researchers. The most recent related work was also presented. Four of these competitions were further reviewed in this regard. A brief overview of the evolution of each of these 4 competitions was also presented in terms of the approaches proposed by their participants over time, when such information was found relevant and available.

## References

[1] P. Rohlfshagen, J. Liu, D. Perez-Liebana, and S. M. Lucas, "Pac-Man Conquers Academia: Two Decades of Research Using a Classic Arcade Game," *IEEE Trans. Games*, vol. 10, no. 3, pp. 233–256, 2017.

[2] R. Miikkulainen, B. D. Bryant, R. Cornelius, I. V. Karpov, K. O. Stanley, and C. H. Yong, "Computational intelligence in games," in *Computational Intelligence: Principles and Practice*, G. Y. Y. and D. B. Fogel, Ed. IEEE Computational Intelligence Society, 2006, pp. 155–191.

[3] "--," 2018. [Online]. Available: http://www.ice.ci.ritsumei.ac.jp/~ftgaic/.

[4] X. Neufeld, S. Mostaghim, and D. Perez-Liebana, "HTN fighter: Planning in a highly-dynamic game," in *Proceedings of the 9th Computer Science and Electronic Engineering Conference*, 2017, pp. 189–194.

[5] S. Demediuk, M. Tamassia, W. L. Raffe, F. Zambetta, X. Li, and F. Mueller, "Monte Carlo Tree Search Based Algorithms for Dynamic Difficulty Adjustment," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 53–59.

[6] M. J. Kim and K. J. Kim, "Opponent Modeling based on Action Table for MCTS-based Fighting Game AI," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 178–180.

[7] N. Duc Tang Tri, V. Quang, and K. Ikeda, "Optimized Non-visual Information for Deep Neural Network in Fighting Game," in *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*, 2017, pp. 676–680.

[8] G. Martínez-Arellano, R. Cant, and D. Woods, "Creating AI Characters for Fighting Games using Genetic Programming," *IEEE Trans. Comput. Intell. AI Games*, vol. 9, no. 4, pp. 423–434, 2017.

[9] T. Kristo and N. U. Maulidevi, "Deduction of Fighting Game Countermeasures using Neuroevolution of Augmenting Topologies," in *Proceedings of 2016 International Conference on Data and*

*Software Engineering*, 2016.

[10] G. L. Zuin, Y. P. A. Macedo, L. Chaimowicz, and G. L. Pappa, "Discovering Combos in Fighting Games with Evolutionary Algorithms," in *Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, 2016, pp. 277–284.

[11] "--," 2018. [Online]. Available: http://www.pacmanvghosts.co.uk.

[12] P. R. Williams, D. Perez-Liebana, and S. M. Lucas, "Ms. Pac-Man Versus Ghost Team CIG 2016 competition," in *Proceedings of the 2016 IEEE Conference on Computatonal Intelligence and Games*, 2016.

[13] M. Morosan and R. Poli, "Evolving a Designer-Balanced Neural Network for Ms PacMan," in *Proceedings of the 2017 9th Computer Science and Electronic Engineering*, 2017, pp. 100–105.

[14] A. Dockhorn and R. Kruse, "Combining Cooperative and Adversarial Coevolution in the Context of Pac-Man," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 60–67.

[15] F. Domínguez-Estévez, A. A. Sánchez-Ruiz, and P. P. Gómez-Martín, *Training Pac-Man bots using Reinforcement Learning and Case-based Reasoning*. CoSECivi, 2017.

[16] M. Miranda, A. A. Sánchez-Ruiz, and F. Peinado, *Pac-Man or Pac-Bot? Exploring Subjective Perception of Players' Humanity in Ms. Pac-Man*. CoSECivi, 2017.

[17] P. R. Williams, S. M. Lucas, and M. Fairbank, "The Effect of Varying Partial Observability in Ms. Pac-Man," 2018.

[18] "--," 2018. [Online]. Available: http://www.is.ovgu.de/Research/HearthstoneAI.html.

[19] M. Swiechowski, T. Tajmajer, and A. Janusz, "Improving Hearthstone AI by Combining MCTS and Supervised Learning Algorithms," in *Proceedings of the 2018 IEEE Conference on Computatonal Intelligence and Games*, 2018, pp. 445–452.

[20] P. García-Sánchez, A. Tonda, A. M. Mora, G. Squillero, and J. J. Merelo, "Automated Playtesting in Collectible Card Games using Evolutionary Algorithms: A Case Study in Hearthstone," *Knowledge-Based Syst.*, vol. 153, pp. 133–146, 2018.

[21] A. Stiegler, K. Dahal, J. Maucher, and D. Livingstone, "Symbolic Reasoning for Hearthstone," *IEEE Trans. Games*, vol. 10, no. 2, pp. 113–127, 2018.

[22] Ł. Grad, "Helping AI to Play Hearthstone using Neural Networks," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2017, vol. 11, pp. 131–134.

[23] D. Deja, "Predicting Unpredictable Building Models Handling Non-IID Data, A Hearthstone Case Study," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2017, vol. 11, pp. 127–130.

[24] A. Santos, P. A. Santos, and F. S. Melo, "Monte Carlo Tree Search Experiments in Hearthstone," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 272–279.

[25] "--," 2018. [Online]. Available: http://cilab.sejong.ac.kr/sc_competition2018.

[26] A. R. Tavares and L. Chaimowicz, "Tabular Reinforcement Learning in Real-Time Strategy Games via Options," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 229–236.

[27] N. Justesen and S. Risi, "Continual Online Evolutionary Planning for In-Game Build Order Adaptation in StarCraft," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 187–194.

[28] N. Justesen and S. Risi, "Learning Macromanagement in StarCraft from Replays using Deep Learning," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 162–169.

[29] M. L. M. Rooijackers and M. H. M. Winands, "Resource-Gathering Algorithms in the Game of StarCraft," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 264–271.

[30] B. S. Weber, "Standard Economic Models in Nonstandard Settings - StarCraft: Brood War," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 417–424.

[31] P. Andersen, M. Goodwin, and O.-C. Granmo, "Deep RTS: A Game Environment for Deep Reinforcement Learning in Real-Time Strategy Games," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 149–156.

[32] "---," 2018. [Online]. Available: https://sites.google.com/site/micrortsaicompetition/home.

[33] A. Uriarte and S. Ontanon, "Single Believe State Generation for Partially Observable Real-Time Strategy Games," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 296–303.

[34] S. Ontañón, "Combinatorial Multi-armed Bandits for Real-Time Strategy Games," *J. Artif. Intell. Res.*, vol. 58, no. 1, pp. 665–702, 2017.

[35] N. A. Barriga, M. Stanescu, and M. Buro, "Game Tree Search Based on Non-Deterministic Action Scripts in Real-Time Strategy Games," *IEEE Trans. Games*, vol. 10, no. 1, pp. 69–77, 2018.

[36] Z. Yang and S. Ontanon, "Learning Map-Independent Evaluation Functions for Real-Time Strategy Games," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 301–307.

[37] P. A. Andersen, M. Goodwin, and O. C. Granmo, "Deep RTS: A Game Environment for Deep Reinforcement Learning in Real-Time Strategy Games," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 149–156.

[38] R. Dubey, J. Ghantous, S. Louis, and S. Liu, "Evolutionary Multi-objective Optimization of Real-Time Strategy Micro," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 133–140.

[39] "--," 2018. [Online]. Available: https://comp.fossgalaxy.com/competitions/t/11.

[40] R. Canaan, H. Shen, R. Torrado, J. Togelius, A. Nealen, and S. Menzel, "Evolving Agents for the Hanabi 2018 CIG Competition," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 409–416.

[41] M. Eger, C. Martens, and M. A. Cordoba, "An Intentional AI for Hanabi," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 68–75.

[42] M. J. H. van den Bergh, A. Hommelberg, W. A. Kosters, and F. M. Spieksma, *Aspects of the cooperative card game Hanabi*. Amsterdam, The Netherlands: Springer, 2016.

[43] J. Walton-Rivers, P. R. Williams, R. Bartle, D. Perez-Liebana, and S. M. Lucas, "Evaluating and modelling Hanabi-playing agents," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation*, 2017, pp. 1382–1389.

[44] "--," 2018. [Online]. Available: www.gvgai.net.

[45] K. Ahmed, M. C. Green, P.-L. Diego, and J. Togelius, "General Video Game Rule Generation," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 170–177.

[46] D. Perez-Liebana, J. Liu, A. Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, "General Video Game AI: a Multi-Track Framework for Evaluating Agents, Games and Content Generation Algorithms," *CoRR*, vol. abs/1802.1, 2018.

[47] W. Woof and K. Chen, "Learning to Play General Video-Games via an Object Embedding Network," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 285–292.

[48] R. R. Torrado, P. Bontrager, J. Togelius, and J. Liu, "Deep Reinforcement Learning for General Video Game AI," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 316–323.

[49] C. F. Sironi and M. H. M. Winands, "Analysis of Self-Adaptive Monte Carlo Tree Search in General Video Game Playing," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 396–400.

[50] A. Dockhorn and D. Apeldoorn, "Forward Model Approximation for General Video Game Learning," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018.

[51] R. D. Diego Perez-LiebanaGaina and S. M. Lucas, "Rolling Horizon Evolution Enhancements in General Video Game Playing," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 88–95.

[52] "--," 2018. [Online]. Available: https://aibirds.org/other-events/level-generation-competition.html.

[53] M. Stephenson and J. Renz, "Procedural Generation of Levels for Angry Birds Style Physics Games," in *Proceedings of the 2016 IEEE*

*Conference on Computational Intelligence and Games*, 2016, pp. 1–8.

[54] M. Stephenson and J. Renz, "Generating Varied, Stable and Solvable Levels for Angry Birds Style Physics Games," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 288–295.

[55] Y. Jiang, T. Harada, and R. Thawonmas, "Procedural Generation of Angry Birds Fun Levels Using Pattern-Struct and Preset-Model," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 154–161.

[56] "--," 2018. [Online]. Available: http://atkrye.github.io/IEEE-CIG-Text-Adventurer-Competition/.

[57] B. Kostka, J. Kwiecieli, J. Kowalski, and P. Rychlikowski, "Text-based Adventures of the Golovin AI Agent," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 181–188.

[58] "--," 2018. [Online]. Available: http://vizdoom.cs.put.edu.pl/.

[59] K. Shao, D. Zhao, N. Li, and Y. Zhu, "Learning Battles in ViZDoom via Deep Reinforcement Learning," in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*, 2018, pp. 389–392.

[60] S. Alvernaz and J. Togelius, "Autoencoder-augmented Neuroevolution for Visual Doom Playing," in *Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games*, 2017, pp. 1–8.

[61] Y. Wu and Y. Tian, "Training Agent for First-person Shooter Game with Actor-critic Curriculum Learning," in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

# On the use of the Finite Element Method to Represent Real-World Phenomena in Spatiotemporal Databases

José Duarte
DETI/IEETA, University of Aveiro
Aveiro, Portugal
hfduarte@ua.pt

*Abstract*—**The methods proposed in the spatiotemporal databases community to represent the continuous evolution of real-world phenomena from observations do not consider the physical characteristics of the phenomena and the external conditions with which they interact. As a result, the representation has no real physical meaning, and it is hard to establish error estimates and bounds. The finite element method approximates the behavior of a phenomenon using equations based on laws and principles of physics. It considers material properties and external conditions, can handle complex geometries, and provides error estimates and bounds. It requires some expertise to be used correctly, can be expensive, does not seem to be suitable to process large datasets of data on the evolution of real-world phenomena, and a structural model has to be defined for every problem. It can be used to predict unknown states, but its use is somewhat limited in the context being proposed.**

***Keywords—finite element method, morphing, region interpolation problem, spatiotemporal databases***

## I. Backgroung

Several technologies exist that can be used to collect data on the evolution of real-world phenomena (e.g., sequences of satellite images tracking the evolution of icebergs in the Antarctic, and video recording the evolution of biological tissue). Our goal is to represent the evolution of real-world phenomena in-between know observations using moving regions [1] (i.e., objects whose position, shape and extent change continuously over time) in spatiotemporal database management systems (STDBMSs).

In this context, creating moving regions from snapshots (observations) is called the region interpolation problem [2], and moving regions are represented using the *sliced representation* [3]. In the *sliced representation*, a moving region is an ordered collection of units. A unit represents the evolution of a geometry between a source and a target known geometries, during an interval of time. The evolution of a geometry within a unit is given by an interpolation function, $F^+$, that should have some properties of interest. In particular: it should have low complexity and allow the processing of large datasets, handle geometries with an arbitrary shape and complexity, generate only valid intermediate geometries, and provide a good approximation of the evolution of the phenomena, ideally with a known error (providing error estimates and bounds), that can be used in applied scientific work (e.g., to perform numerical analysis on the evolution of real-world phenomena).

In [4]–[6] the authors discuss the use of relational database management systems (RDBMSs) as a technology to support scientific computing and computer-based engineering, in particular, to simplify large scale finite element analysis (FEA). This approach differs from the objective (context) presented in this paper (e.g., in this paper, FEA is considered as a method that can potentially be used to create moving regions that will be used to implement operations to study the evolution of phenomena, and the relationships that they establish with other objects and phenomena, over time). The objective is not to use spatiotemporal databases to support FEA.

Morphing techniques are used successfully, for example, in animation packages and computer graphics. Their main goal is to obtain a natural continuous transformation of a geometry between two consecutive known geometries and can potentially be used to implement $F^+$. Several methods have been proposed in the literature (e.g., using: some type of decomposition [7], deformation transfer [8], and physical principles [9]). However, in general, the physical properties of the phenomena being represented and the external conditions with which they interact, that can have an impact on their evolution, are not considered, and no guarantees on the global and local distortions introduced by the methods are given. As a consequence, it is hard to evaluate the quality of the interpolation objectively and establish an approximation error w.r.t. the actual evolution of the phenomena.

Engineering analysis and computational science simulation are used successfully in many fields (e.g., in structural analysis and fluid flow prediction). They can simulate the physical properties of materials and their interaction with external conditions, predict how a phenomenon will evolve in the future, and provide error estimates and bounds. Therefore, this paper presents a critical overview of the use of the finite element method (FEM) in the context of the region interpolation problem. This should also serve as a reference when considering the use of other numerical methods in the same context.

This paper is organized as follows. Section II presents and characterizes the finite element method. Section III presents an overview on the use of the finite element method in the context of the region interpolation problem. Section IV presents a discussion on the main advantages, disadvantages, and challenges of using the finite element method in the new context, taking as a reference the properties defined in Section I for an ideal interpolation function $F^+$. Section V presents the conclusions and future work.

## II. The Finite Element Method

A number of important problems found in nature can be described using partial differential equations (PDEs). Some have no known analytical solution, or if an analytical solution is known, it is not practical to use it. In these situations, numerical methods (e.g., the finite element method (FEM), the boundary element method (BEM), the finite difference method (FDM), the finite volume method

(FVM), and the meshless method) can be used to solve these problems. Because each method has different variations, choosing the best method to be used is application dependent. For example, FEM is a very general method, with a solid mathematical foundation, widely used in continuum mechanics and structural analysis. FVM is preferred in computational fluid dynamics (CFD). It is a conservative method (e.g., it ensures the conservation of mass, momentum and energy at each element of the discretization) widely used to solve models based on conservation laws. It is also becoming common to encounter situations in which different methods are combined to solve a particular problem.

Several types of analysis can be performed (e.g., static, dynamic, linear, and nonlinear). Most real-world phenomena are dynamic and nonlinear in nature. For example, time-dependent (transient) analysis can be used to determine the dynamic response of a structure at different time steps. Nonlinearities include geometric, material, and contact nonlinearities [10] (e.g., large deformations, plasticity, material damage or fracture, and hyper-elasticity). Analysis (e.g., FEA) can be used to understand and predict the behavior, and optimize and control the design and operation, of structures subjected to static or dynamic loads.

Numerical methods have potential applications in several areas [10], and various packages are available to perform numerical analysis on physical phenomena.

FEM [10], [11] is used in engineering as well as in pure and applied sciences (e.g., in continuum mechanics and structural engineering) and it has potential applications in several areas (e.g., geomechanics, biomechanics, and environmental engineering). It is a powerful procedure for the analysis of structures with arbitrary geometry and general material properties, subjected to different types of loads. It can approximate the behavior of a physical (real) system in space and time (e.g., compute the displacements, stresses and strains in a structure under a load). Its main goal is to simulate (predict) with a high degree of accuracy the evolution (behavior) of a phenomenon or structure under certain conditions, using equations that follow established laws and principles of physics, giving error estimates and bounds on the quality of the solution [10]. Various FEM methods and variations have been proposed in the literature (e.g., the extended (XFEM) and the smoothed (S-FEM) finite element methods).

In FEM, a system is divided into a finite number of individual well-defined elements or components whose behavior, specified by a finite number of parameters, can be understood. These simple elements may have physical properties. Then, the solution of the system is given by the local solutions, computed for each element. The quality, validity, and accuracy of the solution depends on the quality of the discretization. In general, geometries with finer elements improve the quality of the simulation (e.g., local displacements and stresses can be captured in greater detail). The precision of the solution, and the efficiency of the method can be improved by choosing an appropriate element type for the discretization. It can be advantageous to use more than one element type to discretize a problem.

In general, FEM solves an equation of the form $\boldsymbol{r} = \boldsymbol{Ku}$, where $\boldsymbol{r}$ is a vector of known values (e.g., loads) $\boldsymbol{K}$ is a matrix of known values representing, for example, stiffness, and $\boldsymbol{u}$ are the unknowns at the nodes of the discretization (e.g., the nodal displacements). Boundary conditions and constraints can also be specified (e.g., known displacements).

### A. *Classification of a Problem*

In order to select an appropriate structural model and computational method for solving a specific problem [10] the following should be considered:

- Identify the relevant physical phenomena influencing the structure being studied, the nature of the problem, the material properties and the differential equations governing the phenomenon.
- Define the level of accuracy desired, and the variables being studied.

The choices made during this phase are extremely important and can have an impact on the accuracy and validity of the results obtained by the simulation [10]. In general, the following steps need to be defined when using FEM:

- Step 1. Select a structural model. This includes choosing an appropriate mathematical model representing the physical problem being studied (e.g., specifying material properties and constraints). Two important properties of an appropriate mathematical problem are effectiveness and reliability [10].
- Step 2. Select a discretization. Create nodes and elements and define boundary conditions and loads. As discussed previously, the accuracy of the analysis depends on the discretization.
- Step 3. Compute the stiffness matrix ($\boldsymbol{K_i}$) and the load vector for each element. The stiffness matrix represents the relationship between the loads and the displacements at each node in an element.
- Step 4. Assemble the global stiffness matrix ($\boldsymbol{K}$) and load vector, compute the unknown displacements, the reactions, and the strains and the stresses for each element. Direct and iterative solvers are available, and the choice on which one to use depends on the problem.
- Step 5. Analyze the results, also known as the postprocessing step (e.g., analyze the displacements, the stresses and the strains). This step is crucial. The results of a simulation should always be checked.

### B. *Error Recovery and Estimates*

Computational methods are applied to conceptual models of reality, and therefore can only compute approximate solutions. The main sources of error [10] are the model and the discretization. Strategies to minimize the error include improving the conceptual and the structural models and using a finer discretization. Because the conceptual and the structural models are in general not perfect, the simulation cannot reproduce exactly the real phenomenon even in a situation where the error is zero. In some problems, round-off errors introduced by finite precision arithmetic in computers can be significant. Several methods to estimate and reduce the error of a solution have been proposed (e.g., a posteriori error estimators and adaptive analysis procedures [12]). FEM provides error estimates and bounds that allow the use of adaptative self-correcting procedures.

### C. *Solving Simultaneous Algebraic Equations*

A system of simultaneous linear algebraic equations can be solved using direct (elimination methods) and iterative or approximate methods. Iterative methods (e.g., the Gauss-Seidel method) are best suited to solve very large systems of equations, in general, avoid round-off errors, and can have convergence problems. Elimination techniques (e.g., Gauss elimination and the Cholesky factorization) can have round-off errors, and difficulties handling ill-conditioned systems that can lead to bad solutions or singularity [13].

A comparison between the Gauss-Seidel and the Gauss elimination methods, commonly used in practice, can be made to have an idea about the algorithmic complexity of these methods. For solving a system of $n$ linear equations, the Gauss-Seidel method performs $n$ divisions, $n^2$ multiplications, and $n^2 - n$ additions in each iteration, the Gauss elimination method uses $n$ divisions, $(1/3)n^3 + n^2$ multiplications, and $(1/3)n^3 + n$ additions [14]. Other elimination and iterative methods are available (e.g., see [13], [14]). The choice of the method to be used depends on the problem being solved.

### D. *Time-Dependent Analysis*

When working with time-dependent problems in dynamic analysis, procedures are required to perform numerical integration in time. In the case of nonlinear dynamic analysis, time integration algorithms can be implicit or explicit [14], [10]. Implicit algorithms satisfy equilibrium conditions at each increment (time step) and are said to be unconditionally stable. Explicit algorithms do not satisfy equilibrium conditions at each time step and are said to be conditionally stable. As a consequence, errors may be amplified during analysis. In order to satisfy equilibrium conditions, implicit algorithms use iterative methods. This makes them more expensive and can cause convergence problems but can act as an error correction mechanism. Because explicit algorithms are conditionally stable, time steps must be small enough to guarantee the accuracy and validity of the solution and avoid numerical instability. Implicit algorithms impose no limit on the size of the time step used but it still has an impact on the accuracy of the solution.

The choice on the approach to be used depends on the problem being solved (e.g., explicit algorithms are generally used to solve highly nonlinear problems with many degrees of freedom [15]). If a suitable time step is chosen both techniques converge to an accurate solution. There are also situations in which it is advantageous to use both techniques for different time steps [15].

### E. *Nonlinear Analysis*

Linear analysis assumes that the shape and the material properties of the structure being simulated do not change significantly during deformation, displacements are infinitesimally small, no gaps or overlaps occur, the nature of the boundary conditions remains unchanged, and there is no time-dependence (In accordance with the steady state assumption [10].). The structure maintains its initial stiffness independently of the amount of deformation, stress developed in response to the load, and on how the load is applied. This assumption simplifies the problem formulation and its solution.

In nonlinear analysis a time-dependent non-steady state is assumed, and equilibrium must be achieved at all time steps. For example, assuming large displacements, rotations, and strains, for a body in motion, its volume, surface area, mass density, stresses, and strains can change continuously over time. Nonlinear problems are solved using iterative methods. This type of analysis does not always converge, and it is sensitive to small variations (perturbations) in the data. This makes it more complex and expensive. Some phenomena can only be simulated using nonlinear analysis, and some expertise is required to ensure the accuracy and the validity of the results. Nonlinear analysis allows the study of, for example, structural response to extreme events, performance under limit conditions and failure, impacts and large deformations, and phenomena that evolve dynamically. Sources of nonlinearities include [10]:

- Nonlinear geometry. Stiffness changes only due to changes in the shape of the geometry.
- Nonlinear material. Stiffness changes due to changes in the material properties during the analysis. Linear material models assume that stress is proportional to strain and that the model will return to its original shape once the load has been removed (i.e., no permanent deformations occur).
- Loss of elastic stability (buckling). Stiffness changes due to the applied loads. Nonlinear analysis can explain the post-buckling behavior of the structure (e.g., if it collapses or is still able to support the load after buckling).
- Contact stresses and nonlinear supports. Support conditions and contact stresses change during the application of the loads.

If large displacements, rotations, and strains occur, nonlinear analysis should be used. A problem can exhibit more than one type of nonlinear behavior [11]. Nonlinear analysis can be used if the nonlinear material properties of the structure being studied are known.

## III. Using the Finite Element Method in the Context of Spatiotemporal Databases

Several types of analysis can be performed, each with its own advantages, disadvantages, and limitations, and there are situations in which it is advantageous to use more than one type of analysis to solve a problem. For example, when analyzing the evolution of icebergs, situations with: a) large displacements, rotations, and strains, b) small displacements and strains, and large rotations, and c) small displacements, rotations, and strains may be encountered. That is, we can potentially use different types of analysis and methods to study the evolution of a phenomenon. It is impractical to analyze the use of all possible types of analysis and FEM methods proposed in the literature. Therefore, the use of a general formulation, called the displacement-based finite element method [10] (based on the principle of virtual work), for the analysis of solids and structures is considered in the remainder of this section. The following is also considered:

- We are not interested in a full analysis (We are interested, in particular, in the displacements: translation and rotation, at the nodes.). This can simplify the analysis.

- FEM cannot be directly applied to the region interpolation problem because it cannot interpolate a geometry between two known geometries. It can however predict unknown states.
- Problems solved using FEM can have millions of degrees of freedom. This is not expected to occur in the context of spatiotemporal databases, for most problems.
- Meshes, matrices, vectors, and functions can be stored in a spatiotemporal database extension (e.g., for PostgreSQL) using abstract data types (ADTs). This includes, for example, the discretization, the stiffness matrix, the loads and the boundary conditions.
- In FEM, some boundary conditions must be set so that the system of equations to be solved has a unique solution (e.g., some displacements must be known).
- Each node in the discretization has at most three degrees of freedom: rotation, and translation in x and y.
- The use of optimized procedures is not considered, and in FEM terms, a few seconds can be considered a considerable amount of time.

In the simplest case, the governing equilibrium equations (corresponding to the nodal point displacements) for the static analysis of structures and solids, assuming linearity and $n$ degrees of freedom, are given by [10]:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{r}, \tag{1}$$

$$\boldsymbol{K}\boldsymbol{u}(t) = \boldsymbol{r}(t), \tag{2}$$

where $\boldsymbol{r}$ is a vector of known loads or forces, $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{u}$ are the unknown nodal point displacements, and $t$ represents time. $\boldsymbol{r}$, $\boldsymbol{u}$, and $\boldsymbol{K}$ are assembled from individual $\boldsymbol{r}^{e}{}_{i}$, $\boldsymbol{u}^{e}{}_{i}$, and $\boldsymbol{K}^{e}{}_{i}$ for each element $i$ of the discretization. In (2) the displacements can be evaluated at any time $t$ independently of the displacement and loading history. This is not the case in dynamic analysis [10]. Equations (1) and (2) can be solved using direct and iterative methods [10]. Iterative methods are usually used to solve very large systems of equations. In our context we assume $n$ is much smaller than 1 million, therefore, we can use direct methods in most cases. Under the linear analysis assumption, $\boldsymbol{K}$ is constant. Therefore, $\boldsymbol{r}$ or $\boldsymbol{r}(t)$, and $\boldsymbol{K}$ or a factorization of $\boldsymbol{K}$ can be stored in the database and retrieved when necessary to compute $\boldsymbol{u}$ or $\boldsymbol{u}(t)$ at time $t$.

In the case of dynamic analysis, assuming linearity, the dynamic equilibrium equation has a characteristic form [10], [12] as in:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{r}, \tag{3}$$

where $\boldsymbol{u} = \boldsymbol{u}(t)$ are the unknown nodal displacements, $t$ represents time, $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}$ is the damping matrix, $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{r}$ is a vector of known loads or forces, and $\ddot{\boldsymbol{u}}$ and $\dot{\boldsymbol{u}}$ are the nodal acceleration and velocity vectors, respectively. $\boldsymbol{C}$ is neglected in some types of dynamic analysis. Equation (3) can be solved using direct integration and mode superposition methods [10].

For example, if using an implicit integration method to solve (3) (e.g., the Newmark integration method that is unconditionally stable), with constant mass, time step, and material properties, and no damping, the displacements in the next time step ($t + \Delta t$) are computed using information about previous time steps. $\boldsymbol{M}$, $\boldsymbol{K}$, $\boldsymbol{r}$, the initial conditions ${}^{0}\dot{\boldsymbol{u}}$, and ${}^{0}\ddot{\boldsymbol{u}}$, the integration constants, and a factorization $\check{\boldsymbol{K}} = \boldsymbol{K} + a_0\boldsymbol{M} + a_1\boldsymbol{C}$ (where $a_0$ and $a_1$ are integration constants) can be stored in the database. Then, at each time step 1) the effective loads (${}^{t+\Delta t}\boldsymbol{r}^{+}$) are computed, 2) the system of equations $\check{\boldsymbol{K}}\,{}^{t+\Delta t}\boldsymbol{u} = {}^{t+\Delta t}\boldsymbol{r}^{+}$ is solved for ${}^{t+\Delta t}\boldsymbol{u}$, and 3) ${}^{t+\Delta t}\ddot{\boldsymbol{u}}$ is computed. Whether or not some components can be computed once and stored in the database depends on the method used to solve (3), and how the problem is defined.

In the case of nonlinear analysis, using an updated Lagrangian formulation based on the principle of virtual work for general nonlinear analysis, assuming large displacements, rotations and strains (the area and the volume of the geometry change continuously), no nonlinearities in the boundary conditions, a negligible damping effect, displacement degrees of freedom only, and deformation-independent loads, the governing equilibrium equations are given by [10]:

$$({}^{t}\boldsymbol{K}_{L} + {}^{t}\boldsymbol{K}_{LN})\boldsymbol{u} = {}^{t+\Delta t}\boldsymbol{r} - {}^{t}\boldsymbol{f}, \tag{4}$$

$$\boldsymbol{M}\,{}^{t+\Delta t}\ddot{\boldsymbol{u}} + ({}^{t}\boldsymbol{K}_{L} + {}^{t}\boldsymbol{K}_{LN})\boldsymbol{u} = {}^{t+\Delta t}\boldsymbol{r} - {}^{t}\boldsymbol{f}, \tag{5}$$

$$\boldsymbol{M}\,{}^{t}\ddot{\boldsymbol{u}} = {}^{t}\boldsymbol{r} - {}^{t}\boldsymbol{f}, \tag{6}$$

for a static analysis (4), a dynamic analysis using implicit time integration (5), and a dynamic analysis using explicit time integration (6). Where ${}^{t}\boldsymbol{K}_{L}$ and ${}^{t}\boldsymbol{K}_{LN}$ are the linear and nonlinear strain incremental stiffness matrices at time $t$, ${}^{t}\boldsymbol{r}$ and ${}^{t+\Delta t}\boldsymbol{r}$ are the vectors of the external applied point loads at times $t$ and $t + \Delta t$, ${}^{t}\boldsymbol{f}$ is a vector of nodal point forces equivalent to the element stresses at time $t$, $\boldsymbol{M}$ is a time-dependent mass matrix, $\boldsymbol{u}$ is a vector of increments in the nodal point displacements, and ${}^{t}\ddot{\boldsymbol{u}}$ and ${}^{t+\Delta t}\ddot{\boldsymbol{u}}$ are vectors of nodal point accelerations at times $t$ and $t + \Delta t$.

Nonlinear problems are solved iteratively for each time step. The iteration process starts with some initial known values from a previous time step. Which components can be stored in the database depends on the method used to solve the problem (e.g., implicit or explicit integration) and the characteristics of the problem (e.g., are the external loads deformation-independent?). Since the solution at a time step $t$ depends on the solution of previous time steps, some precomputed time steps can be stored in the database to accelerate computation.

## IV. Discussion

This section discusses the advantages and disadvantages of using FEM, and numerical methods in general, in the context of spatiotemporal databases, having as a reference the properties defined in Section I for an ideal interpolation function $F^{+}$.

The main advantages of using numerical methods include the following. Numerical methods:

- Can handle a variety of problems (e.g., fluids, and systems with complex geometries and interconnected components), provide useful error estimates and bounds, and error recovery strategies are known and can be used.
- Solve equations based on established laws and principles of physics (i.e., consider the physical properties of materials and the external conditions with which they interact, that can have an impact on their evolution).
- Can approximate the behavior of real-world phenomena with a high accuracy and predict unknown states.

The main disadvantages of using numerical methods include the following:

- Parameter values may have to be provided by the user (i.e., the process in general is not automatic), and the values chosen can have a significant impact in the accuracy and validity of the results.
- The most appropriate type of element to be used depends on the problem, and hybrid meshes can obtain better results in some situations. The ideal discretization depends on the problem and on what is being analyzed.
- Some problems can only be solved using iterative methods (e.g., nonlinear problems) which makes them more expensive. Nonlinearity is abundant in the physical world.
- In time-dependent problems the integration time step chosen can have a significant impact on the accuracy and validity of the analysis. Guidelines exist to find an optimum time step. In general, the shorter the time step the greater the accuracy. In some types of analysis, the solution for an arbitrary time step $t$ depends on solutions from previous steps. For example, given a phenomenon evolving for 20 seconds. If we want to know its state at time step $t = 15s$, assuming that the state at $t = 0s$ is known, we can compute for $t = 15s$ directly from $t = 0s$ with more or less impact on the accuracy and validity of the solution. If, for example, the optimum time step for the problem is $1s$, then we would have to compute for $t = 1s, 2s, 3s, \ldots, 15s$. It seems reasonable that the optimum time step should be used. A possible solution for this limitation is to precompute and store intermediate states in the database.
- An improper choice of a structural model, using an inappropriate numerical procedure, or type of analysis, for example, can lead to “improperly posed”, inaccurate or invalid solutions, that may be so subtle that cannot be perceived by a nonexpert. Therefore, some level of expertise is required. For exemple, nonlinear analysis requires a significant amount of expertise.
- The time spent in the pre-processing and post-processing steps of complex problems can largely exceed the time needed to compute a solution. These steps, in general, require user intervention. In the context of spatiotemporal databases, automatic processes are preferred.
- FEM cannot interpolate a geometry between two known states (i.e., it cannot be used directly in the context of the region interpolation problem).
- Unless a mathematical model is known for a problem and material being analyzed, one has to be constructed which is not a trivial task. This can limit the use of FEM to specific problems and types of materials. In general, the analysis is problem-dependent.
- Overall, FEM does not seem to be suitable to process large datasets of data on the evolution of real-world phenomena, possibly involving nonlinearities.

## V. Conclusion and Future Work

The finite element method is a powerful tool, and care and some level of expertise are needed to use it properly. For every problem a structural model must be defined, which is not a trivial task. It provides error estimates and bounds, and error recovery strategies can be used. It considers the material properties of the phenomena and the external conditions with which they interact and that can have an impact on their evolution. In some situations, the time step used for analysis can have an impact on the accuracy and validity of the solution, and arbitrary time step displacements are computed using information from previous known (or computed) time steps displacements.

It is important to note however that in the context of spatiotemporal databases, we are interested on the evolution (changes) of the nodal displacements (i.e., the translation and rotation of the nodes of the geometry representing the phenomenon) over time, not on a full analysis, and a relatively small number of degrees of freedom (much less than 1 million) are expected to be found in most of the problems being solved. This can simplify or make a finite element analysis less expensive. On the other hand, situations with large displacements, rotations, and strains, and nonlinearities are expected to be encountered, and most problems are time-dependent. The goal is not to use spatiotemporal databases as a data management technology to support the finite element method.

The finite element method can be used in the context of spatiotemporal databases to predict unknown states of real-world phenomena. However, it cannot be used directly to interpolate a geometry between a source and a target known geometries. It can handle complex geometries, but its use is limited to problems and materials for which a mathematical model is known. It provides error estimates and bounds, can simulate the behavior of a phenomenon with high precision, and the level of accuracy can be adapted to the needs of the user. Overall, it does not seem to be suitable to process large datasets of data on the evolution of real-world phenomena, possibly involving nonlinearities. It requires input from the user, some level of expertise, and the results should always be interpreted and analyzed with care.

As is, the finite element method can be used in specific situations, but it does not provide a solution for the problem that we want to solve in the context of spatiotemporal databases. A possible line for research is to study how it could be combined with morphing techniques to improve the interpolation quality of the latter and how it could be used to create a ground truth. Some interesting questions are raised for future work and investigation on this subject:

- Study how morphing and numerical methods can be used together (e.g., to improve the quality of morphing techniques).
- Study the use of meshless methods. These methods avoid some of the problems associated with the use of a discretization.
- Create moving regions for a use case using the finite element method and study its performance and the quality of the representation.

## Acknowledgments

I thank the reviewers for their comments and suggestions. José Duarte has a research grant awarded by FCT with reference PD/BD/142879/2018. This work was performed in the context of the project POCI-010145-FEDER-032636.

## References

[1] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. Lorentzos, M. Schneider and M. Vazirgiannis, “A Foundation for Representing and Querying Moving Objects,” *ACM Trans. Database Syst.*, 2000, vol. 25, no. 1, pp. 1–42.

[2] M. McKenney and J. Webb, “Extracting Moving Regions from Spatial Data,” in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 438–441.

[3] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider, “A Data Model and Data Structures for Moving Objects Databases,” in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 2000, pp. 319–330.

[4] G. Heber and J. Gray, “Supporting Finite Element Analysis with a Relational Database Backend Part I: There is Life beyond Files,”, Microsoft Research Technical Report, MSR-TR-2005-49, April, 2005.

[5] G. Heber and J. Gray, “Supporting Finite Element Analysis with a Relational Database Backend Part II: Database Design and Access,”, Microsoft Research Technical Report, MSR-TR-2006-21, March, 2006.

[6] G. Heber, C. Pelkie, A. Dolgert, J. Gray, and D. Thompson, “Supporting Finite Element Analysis with a Relational Database Backend Part III: OpenDX Where the Numbers Come Alive,”, Microsoft Research Technical Report, MSR-TR-2005-151, December, 2005.

[7] M. Alexa, D. Cohen-Or, and D. Levin, “As-rigid-as-possible shape interpolation,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’00*, 2000, pp. 157–164.

[8] R. W. Sumner and J. Popović, “Deformation transfer for triangle meshes,” *ACM SIGGRAPH*, 2004, p. 399.

[9] H. B. Yan, S. M. Hu, and R. Martin, “Morphing based on strain field interpolation,” *Comput. Animat. Virtual Worlds*, 2004, vol. 15, no. 3–4, pp. 443–452.

[10] K. J. Bathe, *Finite Element Procedures*, 2nd Ed. Klaus-Jürgen Bathe, 2016.

[11] S. S. Bhavikatti, *Finite Element Analysis*. New Age International, 2005.

[12] R. L. Taylor, O. C. Zienkiewicz, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*, 6th Ed. Elsevier, 2005.

[13] S. Chapra, C. and R. Canale, P., *Numerical Methods for Engineers*, 7th Ed. McGraw-Hill Education, 2014.

[14] D. Logan, *A First Course in the Finite Element Method*, 5th ed. Cengage Learning, Inc, 2011.

[15] L. Noels, L. Stainier, and J.-P. Ponthot, “Combined implicit/explicit time-integration algorithms for the numerical simulation of sheet metal forming,” *J. Comput. Appl. Math.*, 2004, vol. 168, no. 1, pp. 331–339.

[16] J. M. Gere, *Mechanics of Materials*, 6th Ed. Bill Stenquist, 2004.

# Clustering of Time Series using Visibility Graphs and Quantile Graphs

Vanessa Silva
*Faculty of Sciences, University of Porto*
Porto, Portugal
vanessa.silva@dcc.fc.up.pt

***Abstract*—Mining interesting features from time series is a crucial task for time series clustering. Typically, such features are obtained from time series specific characteristics such as trend, period, seasonality and other global measures. A recent approach consists on mapping the time series into a graph and then characterize the time series from a graph point of view, that is, using topological metrics. This approach depends on the mapping of the series onto the graph, which is not always an obvious task. In this work two concepts of existing mappings are explored and it is shown that the joint use of these mappings can be an advantage for the grouping of time series. In order to evaluate the proposed approach, the time series grouping based on the metrics extracted from the visibility networks and the quantile networks of time series is applied to a set of specific time series models. The results are promising and show the networks' potential for time series grouping.**

***Index Terms*—Clustering, Time Series, Complex Networks, Topological Features**

## I. Introduction

A time series is a collection of observations indexed in time. The main purpose of time series analysis is to develop mathematical models that provide plausible descriptions of the characteristics of the data with a view to forecasting, simulation and control [1].

The classification of time series is an intrinsic activity to facilitate the handling and the organization of the enormous amount of information that we can capture. However, this is still a very explored field given the great diversity of data and the difficulty in finding an ideal model for the accomplishment of such series classification [2].

The analysis of complex networks has been receiving increasing interest from the research community [3] and this led to the emergence of the new field of Network Science [4]. This field has shown to be very promising with respect to data clustering tasks [5], through the use of topological graph measurements that are currently available [6].

Several network-based time series analysis approaches have been recently proposed, based on mapping time series to the network domain. The mappings proposed in the literature are based on concepts such as correlation [7], phase space reconstruction [8], recurrence analysis [9], visibility [10] or transition probabilities [11]. Some mappings result in networks that have as many nodes as the number of observations in the time series, such as visibility mappings, but others, such as a quantile based mapping [11], allow to reduce the dimensionality of the series while preserving the characteristics of the time dynamics. Network-based time series analysis techniques have been showing promising results and have been successful in the description, classification and clustering of time series of real datasets. Examples of this include automatic classification of sleep stages [12], characterizing the dynamics of human heartbeat [13], distinguishing healthy from non-healthy electroencephalographic (EEG) series [14] and analyzing seismic signals [15].

The general problem of time series clustering concerns the separation of a set of time series into clusters, with the property that the series of the same group have a similar structure and characteristics, and different from the series of other groups. A fundamental problem in the clustering and classification analysis is the choice of a relevant metric. Existing time series clustering approaches are mainly based on methods of distance, such as the Euclidean distance in space points in order to separate the group of time series of clusters, and on approaches based on resources extracted in time domain, frequency domain and wavelet decomposition of the time series, which are later grouped using grouping methods [16]. These approaches have limitations, distance-based methods do not produce the best results and proving to be insufficient. And the methods based on the characteristics of the series depends on the calculation of these characteristics which is not a trivial task since there are several ways to reach your result (or its approximation), as well as the need to pay attention to some parameters, given the diversity of existing time series.

In this work we propose a new approach to group time series in different classes. This approach consists of constructing three network mapping (natural visibility graphs, horizontal visibility graphs and quantile graphs), for each of the time series and calculating the global topological metrics of these networks, that is, average grade, average path length, number of communities, clustering coefficient, and modularity. This set of metrics forms a vector of topological characteristics of the networks that served to feed a clustering algorithm, k-means.

We show the potential of this new approach in a large set of simulated time series models, linear and nonlinear models, and confirm that the use of the different types of mapping of these models in networks results in a set of features that can capture information encoded in each one of the models and thus distinguish them from unsupervised manner.

This paper is organized as follows. We start, in the section II, with a presentation of the background on time series and complex networks. In the section III, we presented a background of time series mappings to complex networks, in the section IV we describe of the approach proposed for this work and presented of the obtained results and their analysis. Finally, in the last section (V), we presented the main conclusions and we mention some of future work.

## II. Basic Concepts

### A. Time Series

A time series $Y_T = (y_1, \ldots, y_T)$ is a set of observations collected at different (usually equidistant) points in time. The main characteristic of a time series is the serial dependence between the observations which restricts the applicability of many conventional statistical models traditionally dependent of the assumption of independent and identically distributed (i.i.d) observations. The main purpose of the analysis of a time series is to develop mathematical models that provide plausible descriptions of the characteristics of the data with a view to forecasting, simulation and control [1].

In our work, we are essentially interested in extracting the most relevant characteristics of the time series through the science of networks so that we can distinguish different models of time series. The traditional approach to analyses the data under these circumstances is to decompose the time series in *trend*, *cycle*, *seasonality* and *random components*. Trend component indicates the long term behavior of the series. The cycle is characterized by smooth and repeated oscillations of rise and fall in the time series around the trend. Seasonal component corresponds to the oscillations of ascent and descent that occur with a fixed period, for example, within a year that they are usually related to the seasons of the year. And the random component represents all the other effects resulting from a multiplicity of factors and of unpredictable nature.

We say that a time series is stationary when its statistical characteristics (mean, variance) are constant over time, that is, its data oscillate around a constant mean with the variance of the fluctuations remaining essentially the same. The stationarity implies that correlation between observations depends only on the time lag between the observations. Most of the methods and models used in time series area imply that the series are stationary.

*1) Time Series Models:* There are many time series models available in the literature. The models are classified into linear and nonlinear. In this work we will simulate a large set of some time series models most used and useful in series analysis. Below we present with more detail each one of these models to use.

The simplest time series process is the purely random process or *white noise*, $(\epsilon_t)$. A white noise is a sequence of i.i.d. random variables with zero mean and constant variance, $\sigma_\epsilon^2$. A particular case is the Gaussian white noise, where $\epsilon_t$ are independent normal random variables [17].

*a) Linear models:* Linear time series models are essentially models for which the conditional mean is a linear function of past values of the time series [18].

**AR**$(p)$**:** $y_t$ is an autoregressive process of order $p$ if it satisfies the following equation [17]:

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t$$

where $p$ is the number of autoregressive terms and $\epsilon_t$ is a white noise process. This model explicitly specifies a linear relationship between the current and past values as suggested by its name. In this work we will study three particular AR models: two AR(1) models with $\phi_1 \in \{-0.5, 0.5\}$ and one AR(2) model with $\phi_1 = 1.5$ and $\phi_2 = -0.75$.

**ARIMA**$(p, d, q)$**:** an autoregressive moving average, ARMA, process combines AR processes and Moving Average, MA, processes which consist of a linear combination of i.i.d random variables (white noise) [17]. Thus, $y_t$ is an ARMA process of order $(p, q)$ if it satisfies the equation:

$$\begin{aligned} y_t &= \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \\ \left(1 - \sum_{i=1}^{p} \phi_i B^i\right) y_t &= (1 + \sum_{i=1}^{q} \theta_i B^i)\epsilon_t \\ \Phi(B) y_t &= \Theta(B)\epsilon_t \qquad (1) \end{aligned}$$

where the white noise $\epsilon_t$ is usually a Gaussian process, $\phi_i$, $i = 1, \ldots, p$ are constants such that $\Phi(z) = 1 - \sum_{i=1}^{p} \phi_i z^i \neq 0$ for $|z| \leq 1$ and $\theta_i$, $i = 1, \ldots, q$ are constants such that $\Theta(z) = 1 + \sum_{i=1}^{q} \theta_i z^i \neq 0$ for $|z| \leq 1$. $B$ represents the backshift operator, $By_t = y_{t-1}$.

Now, assume that you have a nonstationary time series, $x_t$ but whose $d$th-difference $y_t = \nabla^d x_t$ is a stationary ARMA$(p, q)$ process. This means that the time series $y$ whose value at time $t$ is the difference between $x_{t+d}$ and $x_t$. Then $x_t$ is said an AutoRegressive Integrated Moving Average, ARIMA$(p, d, q)$, process and satisfies the following equation:

$$\Phi(B) y_t (1 - B)^d x_t = \Theta(B)\epsilon_t \qquad (2)$$

In this work we will study an ARIMA(1,1,0) model.

**ARFIMA**$(p, d, q)$: a generalization of the ARIMA$(p, d, q)$ process by allowing the parameter $d$ to assume real values. An time series is said to be a autoregressive fractionally integrated moving average, ARFIMA, model if it satisfies the equation:

$$(1 - \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d x_t = (1 + \sum_{i=1}^{q} \theta_i B^i) + \epsilon_t \qquad (3)$$

where $(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k$, parameter $d$ is said the long memory parameter since it controls the rate of decay of the autocorrelation function. When $d \neq 0$ the rate of decay is hyperbolic meaning that persistence in the autocorrelations: there is significant dependence between observations separated by long time intervals [17]. In this work we will study two ARFIMA$(1, 0.4, 0)$ models with $\phi_1 \in \{-0.5, 0.5\}$.

*b) Nonlinear models:* The initial development of nonlinear time series analysis focused on several nonlinear parametric forms. We can distinguish specifications for the conditional mean and specifications for the conditional variance [19].

**SETAR**$(1)$**:** the self-exciting threshold autoregressive (SETAR) models of order 1 specify the nonlinearity in the conditional mean. These are very useful for processes in which regime changes occur, where the idea is to approximate a nonlinear function in a linear function dependent on the regime that changes according to the process values [20]. This model can be presented as follows:

$$y_t = \begin{cases} \alpha y_{t-1} + \epsilon_t, & if \quad y_{t-1} \leq r \\ \beta y_{t-1} + \gamma \epsilon_t, & if \quad y_{t-1} > r \end{cases} \tag{4}$$

where $r$ represents a real threshold and $d$ is the delay parameter.

**INAR**$(1)$**:** the INAR models have been proposed to model correlated integer-valued time series [21]. These models are based on thinning (random) operations defined on the integers. The most common such operation is the binomial thinning defined as follows. Let $X$ be an integer valued random variable and $0 < \alpha < 1$. Then $\alpha * X = \sum_{i=1}^{X} \chi_i$ where $\chi_i \sim Be(\alpha)$, meaning that $\alpha * X | X \sim Bi(X, \alpha)$. The time series $y_t$ is said an INAR$(1)$ if it satisfies:

$$y_t = \alpha * y_{t-1} + \epsilon_t \tag{5}$$

where $\alpha \in [0, 1]$, $\epsilon_t$ are integer valued time series and $*$ defines the binomial thinning operation.

**GARCH**$(p, q)$**:** an autoregressive conditional heteroscedastic model was created to model the volatility (or conditional variance) that is not constant in time, in a homogeneous time model. The basic idea of ARCH models is that the asset return is serially uncorrelated, but dependent, and that the dependence can be described by a simple quadratic function of its lagged values [22]. GARCH is a generalization of the ARCH$(q)$ process that propose that conditional volatility be a function not only of the squares of past errors, which is the case of ARCH models, but also of their own past values [18]. To model a time series $\sigma_t^2$ using a generalized ARCH process, we define:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 \tag{6}$$

where $\omega > 0$ and $\alpha_i, \beta_i \geq 0$, $\epsilon_t$ is an uncorrelated random variable, $z_t$ a white noise with variance 1 and $\sigma_t$ the standard deviation ($\epsilon_t = \sigma_t z_t$). $p \geq 1$ represents the order of dependence of the conditional variance and $q \geq 0$ represents the order of dependence of past shocks. In this work we will study GARCH$(1, 1)$ simulated model.

### *B. Complex Networks*

*graphs* are a very appropriate modeling tool to represent a set of elements that interact and which exhibit emergent collective properties, in which the elements and their relations are represented by *nodes* (or *vertices*) and *links* (or *edges*), respectively. Such a graph has non-trivial topological properties, due to the specific characteristics of the system it represents [23]. We call this graphical representation of *complex network*. We will use these two terms, graph and network, interchangeably.

*1) Graph Terminology and Concepts:* A *graph* $(G)$ is then an ordered pair $(V(G), E(G))$, where $V(G)$ represents the set of *nodes* and $E(G)$ the set of *links* between pairs of elements of the set $V(G)$. The number of nodes, also known as the *size* of the graph, is written as $|V(G)|$ and the number of links as $|E(G)|$. A $k$-graph is a graph of size $k$.

Two nodes are *neighbors* or *adjacent* if they are connected by a link, that is, if $(v_i, v_j) \in E(G)$ then $v_i$ and $v_j$ are neighbors. We can distinguish between *directed* links, which connect a source node to a target node, and *undirected* links, when there is no such concept of orientation. In the first case the graph is called directed or *digraph*.

A graph can also be *weighted*, this means that at each link $(v_i, v_j)$ is associated with a weight (or cost) $w_{i,j}$, and this weight can be positive or negative.

A graph is classified as *simple* if it does not contain multiple links, two or more links connecting the same pair of nodes, and it does not contain self-loops (a link connecting a node to itself).

*a) Path:* A *path* is a sequence of nodes in which each consecutive pair of nodes in the sequence is connected by a link.

*b) Connectivity:* The concept of *connectivity* is extremely important in networks. We say that two nodes are then *connected* if there is a path between them and are *disconnected* if such a path does not exist. In undirected graphs, if nodes $v_i$ and $v_j$ are connected and nodes $v_j$ and $v_k$ are connected, then $v_i$ and $v_k$ are also connected. This property can be used to partition the nodes of a graph in non-overlapping subsets of connected nodes known as *connected components* [6]. Although there is at least one path between any two nodes in the same component, there is no path between nodes belonging to different components [4].

*2) Topological Metrics:* There is a vast set of topological metrics of available graphs [6], each reflecting some particular features of the system under analysis. In our work we are essentially concerned with studying a specific set of simple metrics. We present below a brief description of them.

*a) Average degree:* The degree of a graph is a fairly important local property of each node, this represents the number of links that the node has for the other nodes, in undirected graphs [4]. We denote by $k_i$ the degree of the $i$-th node.

In digraphs, we distinguish between the *in-degree*, $k_i^{in}$, and the *out-degree*, $k_i^{out}$. The first represents the number of links that point to node $v_i$, and the second the number of links that point from node $v_i$ to other nodes. The total degree, $k_i$, in a digraph is given by the sum of the two.

In weighted graphs we may want to obtain the weighted degree that is similar to the previous measure, the difference is that instead of adding the quantity of connections, we sum the weights of each of the links [6].

From this measure we can obtain the average degree ($\bar{k}$) that can be easily obtained by calculating the arithmetic mean of the degrees of all nodes in the graph.

*b) Average path length:* A path is a sequence of nodes in which each consecutive pair of nodes in the sequence is connected by a link. It may also be useful to think of the path as the sequence of links that connect those nodes. In digraphs the path follows the direction of the source node for the target node.

We denote by $\bar{d}$, the arithmetic mean of the shortest paths ($d$) among all pairs of nodes (both ways for directed graphs), the path length being the number of links, or the sum of the links weights if the graph is weighted, in the path [4]. It should be noted that $\bar{d}$ is measured only for the node pairs that are in the same component.

For a directed graph the average path length is given by:

$$\bar{d} = \frac{1}{|V(G)|(|V(G)|-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{|V(G)|} d_{i,j} \tag{7}$$

*c) Global clustering coefficient:* This measure ($C$), also called global transitivity, measures the total number of closed triangles in the graph, that is, it measures the degree to which the nodes in a graph tend to cluster. It is calculated by the ratio of the number of closed triangles ($N_{\triangle}$) to the number of possible triangles, that is, the amount of connected triplets of nodes ($N_3$) [6]. In this work we will call it only as a *clustering coefficient*.

For undirected and unweighted graphs, we mathematically have to:

$$C = \frac{3N_{\triangle}}{N_3} \tag{8}$$

Factor three explains the fact that each triangle can be seen to consist of three different triangles, one with each of the nodes as the central node, and ensures that $0 \leq C \leq 1$. For directed graph the direction of the edges is ignored.

For weighted graphs there are several generalizations of clustering coefficient, here we use the definition by A. Barrat [24], this is a local vertex-level quantity, its formula is:

$$C_i = \frac{1}{k_i^w (k_i - 1)} \sum_{j,h} \frac{(w_{i,j} + w_{i,h})}{2} a_{i,j} a_{i,h} a_{j,h} \tag{9}$$

where $k_i^w$ is the weighted degree, $a_{i,j}$ are elements of the adjacency matrix, $k_i$ is the degree and $w_{i,j}$ are the weights.

*d) Number of communities:* Number of communities($S$) measures the number of denser subgraphs in a network, that is, subsets of nodes within the graph such that connections between the nodes are denser than connections to the rest of the network.

The function we used to help us calculate this metric tries to find densely connected subgraphs, also called communities here, via random walks. The idea is that short random walks tend to stay in the same community. This function is the implementation of the Walktrap community finding algorithm [25].

*e) Modularity:* Measures how good the division of the graph is in specific communities, that is, how different are the different nodes, belonging to different communities, from each other. A high modularity value, ($Q$), indicates a graph with a dense internal community structure, that is, with many edges between nodes within communities and sparse connections between nodes of different communities [26].

If a particular network is split into $c$ communities, $Q$ can be calculated from the symmetric $c \times c$ mixing matrix $E(G)$ whose elements along the main diagonal, $e_{ii}$, give the fraction of connections between nodes in the same community $i$ while the other elements, $e_{ij}(i \neq j)$ identify the fraction of connections between nodes in the different communities $i$ and $j$. The calculation of $Q$ can then be performed as follows [6], [26]:

$$Q = \sum_i [e_{ii} - (\sum_j e_{ij})^2] \tag{10}$$

The situation $Q = 1$ identifies networks formed by disconnected modules.

## III. Mappings from Time Series to Complex Networks

Several network-based time series analysis approaches have been recently proposed, based on mapping time series to the network domain. The objective is to map a time series for a complex network using a particular concept, in our particular work, based on the concepts of visibility and probability of transition existing in the literature.

### A. Natural Visibility Graph

This method was proposed for the first time by Lacasa et al. [10]. Named natural visibility graph (NVG), each node in the graph corresponds, in the same order, to the time series data and two nodes are connected if there is a line of visibility between the corresponding data points, that is, if it is possible to draw a straight line in the time series that joins the two corresponding data points that intercepts no data "height" between them. If we consider each time instant as a node of a graph, then two nodes are connected if the tops of the corresponding vertical bars are visible to each other, that is, if there is a straight line from the top of the two bars that does not intersect other bars. This mapping is illustrated in figure 1 with a toy time series and the resulting network.

The resulting graph has as many nodes as the number of observations in the time series. The nodes are numbered sequentially in time and each node corresponds to an observation $(t_a, y_a)$. Two nodes $(t_a, y_a)$ and $(t_b, y_b)$ are connected (have visibility) if any other observation $(t_c, y_c)$ with $t_a < t_c < t_b$ satisfies:

$$y_c < y_b + (y_a - y_b)\frac{(t_b - t_c)}{(t_b - t_a)} \tag{11}$$

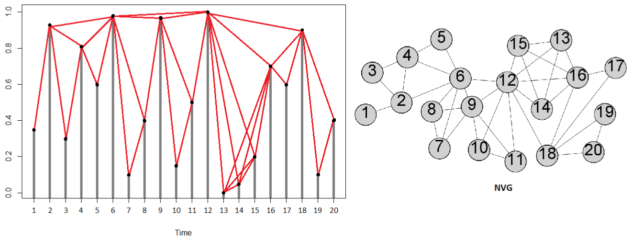


Fig. 1: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the natural visibility algorithm. The red lines in the time series plot represent the lines of visibility (and hence the links of the graph) between all data points.

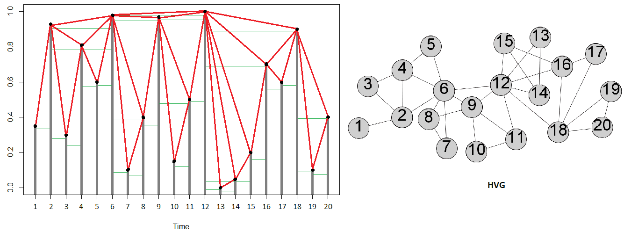


Fig. 2: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the horizontal visibility algorithm. The green lines represent the horizontal lines of visibility between all the data points and the red lines the respective connections between the points.

The graphs obtained always have the following characteristics [10]:

- **Connected:** each node sees at least its nearest neighbors (left-hand side and right-hand side).
- **Undirected:** the way the algorithm is built up, there is no direction defined in the links. However, this direction could be defined considering the direction of the time axis.
- **Invariant under affine transformations of the series data:** the visibility criterion is invariant under rescheduling of both the horizontal and vertical axis, as well as in horizontal and vertical translations.
- **"Lossy":** some information regarding the time series is inevitably lost in the mapping from the fact that the network structure is completely determined in the (binary) adjacency matrix. For instance, two periodic series with the same period as $Y_a = ..., 3, 1, 3, 1, ...$ and $Y_b = ..., 3, 2, 3, 2, ...$ would have the same visibility graph, albeit being quantitatively different. One possible solution would be the use of weighted networks, where weights determine the height difference of the associated data, for example.

### *B. Horizontal Visibility Graph*

In order to reduce the computational complexity associated to NVG, Luque et al. [27] proposed in 2009 a simplified NVG method called the horizontal visibility graph (HVG), which inherits all NVG features mentioned above.

In this alternative, two nodes in the graph are connected if it is possible to draw a horizontal line in the time series joining the two vertical bars, corresponding to the two data, which does not intercept any height of the intermediate data. In the figure 2 we give a simple illustration of this method, with a toy time series and the resulting network.

Formally, two nodes $(t_a, y_a)$ and $(t_b, y_b)$ are connected, have visibility, if the following condition is fulfilled:

$$y_a, y_b > y_c \tag{12}$$

for all $t_c$ such that $t_a < t_c < t_b$.

The HVG is always a subgraph of the NVG associated with the same time series, for example, if we analyze both graphs in the figures 1 and 2 we can easily verify that all the links present in the HVG are present in NVG, but there are links in NVG that are not in HVG, two examples are links $(7, 9)$ and $(13, 15)$. Therefore, the HVG nodes will always have a degree less than or equal to the nodes of the corresponding NVG, since they will have "less visibility" and consequently will have less quantitative information.

### *C. Quantile Graph*

Quantil graphs (QG) were introduced by Campanharo et al. [11]. It is a different approach from previous methods of visibility, but captures oscillations over time. This method divides the time series into $Q$ quantiles, $q_1, q_2, ..., q_Q$, and each quantile, $q_i$, is associated to a node $v_i$ of the graph. So the graph has as many nodes as the number of quantiles. Two nodes $v_a$ and $v_b$ are connected by a weighted directed link $(v_a, v_b, w_{a,b})$, where the weight $w_{a,b}$ represents the number of times an observation $(t_n, y_n)$, belonging to the quantile $q_a$, is followed by an observation $(t_{n+1}, y_{n+1}$, belonging to the quantile $q_b$. The weights are normalized such that the adjacency matrix becomes a Markov transition matrix, where $\sum w_{a,b} = 1$. The resulting networks are weighted and directed. This mapping is illustrated in figure 3 with a toy time series and the resulting network.

These networks have a significant loss of information on small amplitude variations, especially if the value of $Q$ is very small. Its connectivity represents the causal relationships contained in the dynamics of the process it represents.

For this work we chose to use $50$ quantiles and let us refer to the generated QGs as 50-QG or simply Q50.

## IV. Clustering of Time Series Models

The purpose of cluster analysis is to discover the natural groupings of a set of patterns, points, or objects. It consists of the empirical formation of groups of objects, called clusters, with high intra-cluster similarity and low inter-cluster similarity. That is, given a representation of $n$ objects, the goal is to find $k$ clusters based on a measure of similarity, so that the similarities between objects in the same cluster are high,

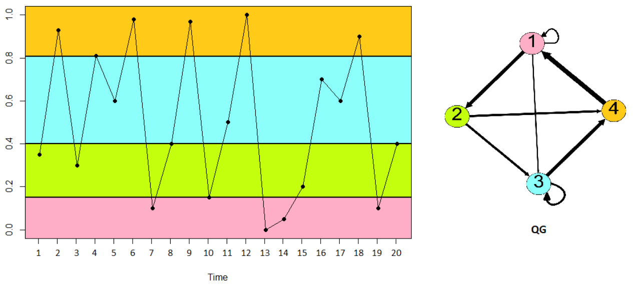


Fig. 3: On the left side, we present the plot of a toy time series and, on the right side, the network generated by the quantile algorithm. The different colors represent the region (in the time series plot) corresponding to the different quantiles (in this case $Q = 4$). Repeated transitions between quantiles result in edges in the network with larger weights represented by thicker lines.

whereas the similarities between objects in different clusters are low [28].

In the clustering of time series, we can distinguish between two main categories, the one that performs clustering on a set of time series with the purpose of grouping them in different clusters, and the one that performs clustering on ”windows” of a single time series whose objective is to find similarities and differences between different windows of time. Here we focus on the first category.

One of the major problems in time series clustering analysis is the choice of a relevant metric to perform grouping. Approaches involving the measure of similarity between global characteristics of the time series were proposed to improve old approaches based on similarity measures (eg, Euclidean distance, Dynamic Time Warping, autocorrelation, spectrum, ...), between the actual observations of time series [16]. In this paper we present a new approach in this direction that consists of resorting to the science of complex networks.

More precisely, we propose an approach that involves the measurement of similarity between characteristics, where the set of resources (topological metrics) used for clustering analysis are extracted from complex networks that are created for each of the time series that we want to group. Therefore, the proposed approach involves the following tasks:

1) For each of the time series under analysis, we generated the corresponding natural visibility graph (NVG), the horizontal visibility graph (HVG) and the quantile graph (QG). For the QG we use a total of 50 quantiles.
2) For each of the networks (or graphs) we have computed the five topological metrics mentioned in subsection II-B: $\bar{k}$, $\bar{d}$, $S$, $C$, and $Q$.
3) Since the interval of each of the calculated measures can vary significantly, we apply the Min-Max normalization so that each measure is in the range $[0, 1]$, preventing some measures from dominating the others in the grouping process.
4) After calculating all the metrics, we obtain a vector with fifteen topological features, which will be scaled through the principal component analysis (PCA) [29] and then by the t-distributed stochastic neighbor embedding (t-SNE) technique [30], that will feed the $k$-means.

All the computations are performed in `R` [31] (version 3.4.4), using specific packages, such as `igraph` [32] for graph generation and calculation of metrics, and `timeSeries` [33], `fracdiff` [34], `fGarch` [35], and `rugarch` [36], for the simulation of some time series models mentioned below. The simulation of some time series models required the implementation of the appropriate procedure.

The main idea of this new approach is to show that the use of complex networks can easily distinguish time series models from a wide set of different models, showing that the topological characteristics of the networks corresponding to the series can capture the global nature of the same. We also want to show that the joining of the visibility methods and the quantile methods improves the clustering of the series in contrast to the use of only one mapping method. This is our main contribution, since until now no other paper joins different concepts of network mapping.

### A. Dataset

In order to apply the proposed approach we decided to simulate a large set of time series models that are most common and widely used in the statistical theory and practice of time series analysis. We generate, using `R` software and appropriate packages, 100 sample of size $T = 10000$ of each of the 10 models previously presented (subsection II-A), in a total of 1000 time series.

We refer to these models as follows:

- **White noise:** only `White Noise`;
- **AR models:** `AR(1)-0.5`, `AR(1)0.5` for AR(1) processes with parameters $\phi_1 \in \{-0.5, 0.5\}$; and `AR(2)` for AR(2) process;
- **ARIMA models:** only `ARIMA(1,1,0)`;
- **ARFIMA models:** `ARFIMA(1,0.4,0)-0.5`, `ARFIMA(1,0.4,0)0.5` for ARFIMA$(1, 0.4, 0)$ processes with parameters $\phi_1 \in \{-0.5, 0.5\}$;
- **SETAR models:** only `SETAR(1)`;
- **INAR models:** only `INAR(1)`;
- **GARCH models:** only `GARCH(1,1)`.

The time series are then mapped into networks using the NVG, HVG and 50-QG methods. The resulting 3000 $(1000{*}3)$ networks are characterized by the topological metrics. We obtain a data frame of 15 variables (features) and 1000 instances.

### B. Results

We performed 7 types of clustering analysis, using 7 different feature vectors, namely, the metrics of only one of the mapping methods (3 different vectors), two to two metrics of the mapping methods (3 different vectors), and finally, a vector containing all the metrics obtained from the three different mapping methods, and we compared the results with the true classification, (the original time series models), using

| **Mappings** | **Adjusted Rand Index** | **Average Silhouette** |
|---|---|---|
| NVG | 0.36 | 0.51 |
| HVG | 0.63 | 0.66 |
| Q50 | **0.64** | **0.73** |
| NVG-HVG | 0.68 | 0.63 |
| NVG-Q50 | **0.78** | **0.75** |
| HVG-Q50 | **0.79** | **0.73** |
| NVG-HVG-Q50 | **0.80** | **0.73** |

TABLE I: Clustering evaluation metrics for the different clustering analysis. The results refer to the evaluation metrics for the dataset.

the clustering evaluation metrics mentioned earlier, in order to support our assertion that using two different concepts of mapping methods is an advantage for better recognition of network characteristics (and time series inevitably).

We divide this results into two parts: the results obtained from the principal components analysis (as a dimensionality reduction technique) and then the results obtained from the clustering analysis (using $k$-means algorithm and knowing *a priori* the correct number of clusters in the dataset, that is, $k = 10$).

To evaluate the results of clustering, we chose two evaluation metrics, namely, *adjusted Rand index* [37] and *average silhouette*. The two measures have different functions, the first is a measurement of the accuracy of the results: compares the clusters obtained with the true clusters. The second measures the quality of clusters obtained without knowledge of the true clusters. Adjusted Rand index takes values between $-1$ and 1. It is negative if the index is less than the expected index. Its expected value is 0 in the case of random clusters. A larger Adjusted Rand Index means a higher agreement between two partitions. And the average silhouette takes values between $-1$ and 1, where a high value indicates that the object is well compatible with its own cluster but not with neighbor clusters. If most objects have a high value, the cluster configuration is appropriate.

The results of the clustering evaluation metrics we choose (adjusted Rand index and average silhouette) obtained for the different combinations of feature vectors are presented in the table I, columns 2 and 3, respectively.

The colors represent the two maximum values of the corresponding column, with the darker color highlighting the maximum value and the lighter color the second maximum value.

We can observe that the feature vector that is closest to the real clusters is the one corresponding to the junction of the three proposed mapping methods, (NVG, HVG and 100-QG), with a value of 0.80 in a range of $[-1, 1]$. The vectors that obtained the best values (0.75 and 0.73, in a range of $[-1, 1]$) of the average silhouette are the vectors corresponding to the metrics of the graphs obtained by the two concepts of mapping together. We thus show that the best results are those that use in the dataset the two types of time series mapping concepts in networks (concept of visibility and concept of probabilities of transition). Thus, we prove that the addition of more information about the data do translate into a better result, as we expected. This evidence is further reinforced by the fact that the combination of the two visibility methods yields (NVG and HVG) 0.68 for adjusted Rand index, which is much lower than the best results obtained (0.80, 0.79 and 0.78) that correspond to the vectors obtained from the visibility and quantile methods.

If we focus on feature vector corresponding to just one type of mapping, we note that the that best capture the characteristics of the time series are the 50-QGs with an adjusted Rand index of 0.64 compared to the NVG and HVG that obtained 0.36 and 0.63, respectively, and the average silhouette value is the highest (0.64). This is in agreement with the expected one, since this method better captures the variability of the observations of the time series. And the visibility of methods capture more global structural properties of time series.

Let us now analyze in more detail some results obtained. First we will analyze the results obtained by the PCA corresponding to the characteristic vector using the three mappings (the best value of adjusted Rand index). And then we will analyze the results of clusters obtained for this vector.

*1) PCA Results:* Figure 4 represent the biplot obtained by the PCA for the feature vector that obtained the best adjusted Rand index.

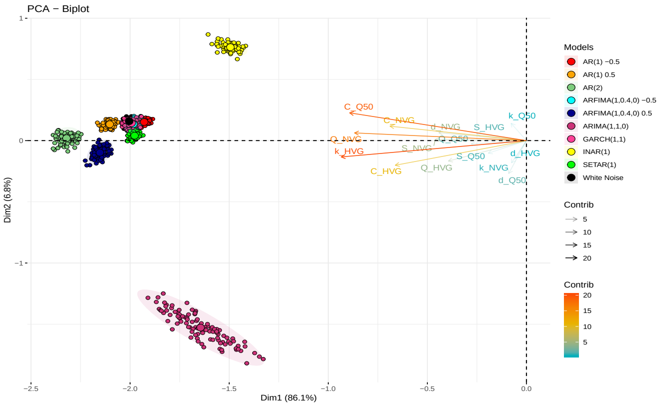


Fig. 4: Results of the PCA analysis. Objects belonging to different groups have different colors, and the arrows represent the contributions of the features to the PCs (the larger the size, sharpness, and closer to orange the greater the contribution of the feature).

We can start by noting that several of the objects belonging to different classes are actually separated into clusters in this bidimensional space, showing that most of the networks are first grouped correctly using the two principal components.

We can also verify that the apparently more dissimilar objects are those corresponding to models with specific characteristics such as trend (`ARIMA(1,1,0)`), periodicity (`AR(2)` and `ARFIMA(1,0.4,0)0.5`), counting (`INAR(1)`), and regime changes (`SETAR(1)`). On the other hand there is a greater difficulty in distinguishing the networks corresponding to the `ARFIMA(1,0.4,0)-0.5`, `White Noise` and

GARCH(1,1) models, which are processes that are somewhat similar, mainly the last two, and therefore expected.

We can still verify that, just as the arrows in the plot themselves suggest, different topological metrics of the three mapping methods contribute to distinguish different time series models. The $\bar{d}$ and $C$ of Q50, $\bar{k}$ of NVG, and $C$ of NVG and HVG, contribute to distinguish second PC, that is, INAR(1), ARIMA(1,1,0), AR(2), and ARFIMA(1,0.4,0)0.5 models. The $C$ and $Q$ of NVG, $\bar{k}$ and $C$ of HVG, and $C$ of 50-QG, contribute to distinguish first PC, that is, AR(2), ARFIMA(1,0.4,0)0.5, AR(1)0.5, ARIMA(1,1) and INAR(1).

*2) Custer Results:* In figure 5 we present an enhanced jitter strip chart, where the width of the jitter is controlled by the density distribution of the data within each class. We can see an almost perfect attribution of the objects by the different clusters, with the exception are ARFIMA(1,0.4,0)-0.5, GARCH(1,1) and White Noise networks which are not distinguishable and are assigned to the same clusters. We conclude that these are the most similar models from the point of view of the complex networks, and consequently more difficult to distinguish, that in the perspective of the analysis of time series have very similar characteristics.

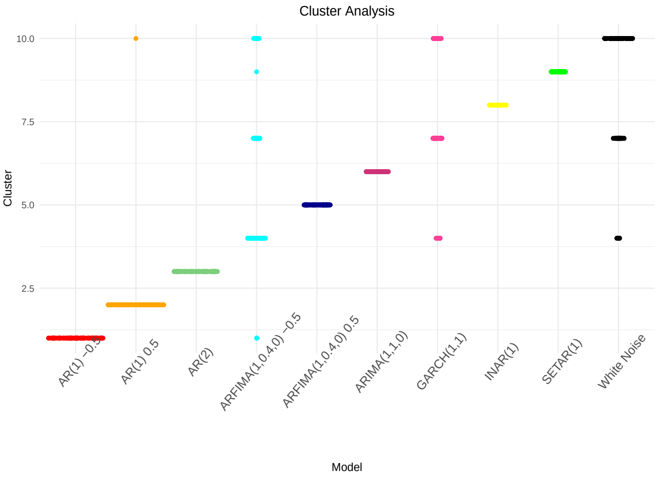


Fig. 5: Plot the distribution of the objects corresponding to the time series models by the different clusters.

As we specify *a priori* in the algorithm $k$-means that the number of clusters is 11, the algorithm ”divides” these three models into three different clusters. But we can see from the breadth of the traces that this distribution is not uniform, this emphasizes a possible ”capacity” to distinguish, mainly, the ARFIMA(1,0.4,0)-0.5 and White Noise models, which are mostly distributed by cluster 4 and 10, respectively.

Although there is not a completely perfect assignment by clusters, we can conclude that it is a good and relevant result.

## V. Conclusion

Classical approaches to time series analysis present severe limitations when analyzing sets of time series. A recent and very promising conceptual approach relies on mapping the time series to complex networks, where the large set of network science methodologies can help in grouping time series.

Our objective with this work is to contribute to the improvement of the methods of clustering of time series, using a complementary area. For this we construct a dataset of 3000 synthetic complex networks, distributed by 10 types of time series models, and we analyse using data mining tools.

The results show that our approach is able to group almost all different time series models using a set of basic topological metrics of complex networks based on different mapping methods.

The main advantage of the proposed approach is to be a completely nonparametric method that can serve as a solution to the parametric and statistical methods of time series analysis. We show that different mappings complement each other, identifying different characteristics of time series. Results show the validity and discrimination power, we were able to distinguish networks corresponding to non-stationary from stationary time series models, counting from non-counting time series models, periodic from non-periodic time series models, and state models from state models time series. More specifically, out of 10 different network types we can distinguish perfectly 7 from them. However, we could not group the networks corresponding to the ARFIMA(1,0.4,0)-0.5, GARCH(1,1) and White Noise models into different clusters, given their very similar characteristics.

In future work, we want to explore new sets of topological metrics as well as new types of mapping of series for network, in order to improve our results, more specifically, to be able to separate perfectly the models that we can not achieve in this work.

## References

[1] Box GE, Jenkins GM, Reinsel GC, Ljung GM. Time series analysis: forecasting and control. John Wiley & Sons; 2015.
[2] Paparrizos J, Gravano L. Fast and accurate time-series clustering. ACM Transactions on Database Systems (TODS). 2017;42(2):8.
[3] Newman ME. The structure and function of complex networks. SIAM review. 2003;45(2):167–256.
[4] Barabási AL. Network Science. Cambridge University Press; 2016.
[5] Kantarci B, Labatut V. Classification of complex networks based on topological properties. In: Cloud and Green Computing (CGC), 2013 Third International Conference on. IEEE; 2013. p. 297–304.
[6] Costa LdF, Rodrigues FA, Travieso G, Villas Boas PR. Characterization of complex networks: A survey of measurements. Advances in physics. 2007;56(1):167–242.
[7] Zhang J, Small M. Complex network from pseudoperiodic time series: Topology versus dynamics. Physical review letters. 2006;96(23):238701.
[8] Gao Z, Jin N. Complex network from time series based on phase space reconstruction. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2009;19(3):033137.
[9] Marwan N, Donges JF, Zou Y, Donner RV, Kurths J. Complex network approach for recurrence analysis of time series. Physics Letters A. 2009;373(46):4246–4254.
[10] Lacasa L, Luque B, Ballesteros F, Luque J, Nuno JC. From time series to complex networks: The visibility graph. Proceedings of the National Academy of Sciences. 2008;105(13):4972–4975.
[11] Campanharo AS, Sirer MI, Malmgren RD, Ramos FM, Amaral LAN. Duality between time series and networks. PloS one. 2011;6(8):e23378.

[12] Zhu G, Li Y, Wen PP. Analysis and classification of sleep stages based on difference visibility graphs from a single-channel EEG signal. IEEE journal of biomedical and health informatics. 2014;18(6):1813–1821.

[13] Shao ZG. Network analysis of human heartbeat dynamics. Applied Physics Letters. 2010;96(7):073703.

[14] Campanharo A, Ramos F. Distinguishing different dynamics in electroencephalographic time series through a complex network approach. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics. 2017;5(1).

[15] Telesca L, Lovallo M. Analysis of seismic sequences by using the method of visibility graph. EPL (Europhysics Letters). 2012;97(5):50002.

[16] Hennig C, Meila M, Murtagh F, Rocci R. Handbook of cluster analysis. CRC Press; 2015.

[17] Shumway RH, Stoffer DS. Time series analysis and its applications. Springer; 2017.

[18] Cryer JD, Chan KS. Time Series Analysis With Applications in R. New York: Springer; 2008.

[19] Franses PH, Van Dijk D. Non-linear time series models in empirical finance. Cambridge University Press; 2000.

[20] Tong H. Threshold models in time series analysis—30 years on. Statistics and its Interface. 2011;4(2):107–118.

[21] Silva I, Silva ME, Pereira I, Silva N. Replicated INAR(1) processes. Methodology and Computing in applied Probability. 2005;7(4):517–542.

[22] Tsay RS. Analysis of financial time series. vol. 543. John Wiley & Sons; 2005.

[23] Albert R, Barabási AL. Statistical mechanics of complex networks. Reviews of modern physics. 2002;74(1):47.

[24] Barrat A, Barthelemy M, Vespignani A. The Architecture of Complex Weighted Networks: Measurements and Models. In: Large Scale Structure And Dynamics Of Complex Networks: From Information Technology to Finance and Natural Science. World Scientific; 2007. p. 67–92.

[25] Pons P, Latapy M. Computing communities in large networks using random walks. In: International symposium on computer and information sciences. Springer; 2005. p. 284–293.

[26] Clauset A, Newman ME, Moore C. Finding community structure in very large networks. Physical review E. 2004;70(6):066111.

[27] Luque B, Lacasa L, Ballesteros F, Luque J. Horizontal visibility graphs: Exact results for random time series. Physical Review E. 2009;80(4):046103.

[28] Jain AK. Data clustering: 50 years beyond K-means. Pattern recognition letters. 2010;31(8):651–666.

[29] Abdi H, Williams LJ. Principal component analysis. Wiley interdisciplinary reviews: computational statistics. 2010;2(4):433–459.

[30] Van Der Maaten L. Accelerating t-SNE using tree-based algorithms. Journal of machine learning research. 2014;15(1):3221–3245.

[31] R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2018. Available from: https://www.R-project.org.

[32] Csardi G, Nepusz T. The igraph software package for complex network research. InterJournal. 2006;Complex Systems:1695. Available from: http://igraph.org.

[33] Wuertz D, Setz T, Chalabi Y. timeSeries: Rmetrics - Financial Time Series Objects; 2017. R package version 3042.102. Available from: https://CRAN.R-project.org/package=timeSeries.

[34] original by Chris Fraley S, U Washington, port by Fritz Leisch at TU Wien; since 2003-12: Martin Maechler; fdGPH SR, fdSperio, etc by Valderio Reisen, Lemonte A. fracdiff: Fractionally differenced ARIMA aka ARFIMA(p,d,q) models; 2012. R package version 1.4-2. Available from: https://CRAN.R-project.org/package=fracdiff.

[35] Wuertz D, Setz T, Chalabi Y, Boudt C, Chausse P, Miklovac M. fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling; 2017. R package version 3042.83. Available from: https://CRAN.R-project.org/package=fGarch.

[36] Ghalanos A. rugarch: Univariate GARCH models.; 2018. R package version 1.4-0.

[37] Campello RJ. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters. 2007;28(7):833–841.

# Comparison Study of Well-Known Inverted Pendulum Models for Balance Recovery in Humanoid Robot

Mohammadreza Kasaei, Nuno Lau and Artur Pereira
IEETA / DETI University of Aveiro 3810-193 Aveiro, Portugal
{mohammadreza, nunolau, artur}@ua.pt

*Abstract*— **Bipedal robots are essentially unstable because of their complex kinematics as well as high dimensional state space dynamics, hence control and generation of stable walking is a complex subject and still one of the active topics in the robotic community. Nowadays, there are more humanoids perform stable walking, but fewer show effective push recovery under pushes.**

**In this paper, we firstly review more common used abstract dynamics models for a humanoid robot which are based on the inverted pendulum and show how these models can be used to provide walking for a humanoid robot and also how a hierarchical control structure could fade the complexities of a humanoid walking. Secondly, the reviewed models compare together not only in an analytical manner but also by performing several numerical simulations in a push recovery scenario using MATLAB. These theoretical and simulation studies quantitatively compare these models regarding regaining balance. The results showed that the enhanced version of Inverted Pendulum Plus Flywheel is the ablest dynamics model to regain the stability of the robot even in very challenging states.**

**Keywords:** Humanoid robot, Inverted Pendulum, Stable walk engine, Push recovery.

## I. Introduction

Nowadays researches in the field of humanoid robots are increasing and one of the common targets of these researches is realizing a humanoid robot which is able to operate in our dynamic daily-life environments with the same skill as humans. The application of this type of robot is not just doing our daily life tasks but also they can be used in several different applications such as rescue missions, helping incapable peoples, etc. Unlike wheeled robots, humanoid robot can adapt to our environments without facing limitations like gaps, uneven terrain and so on. It's just because of their similarity in kinematics as well as dynamics with a human. One of the essential requirements for using humanoid robots in such environments is capability to perform tasks in a safe manner and the most important part of this requirement is stable locomotion. Generally, a humanoid robot have more than 20 degrees of freedom (DoF), therefore, they have complex dynamics as well as kinematic. In particular, they are unstable inherently, therefore they need robust dynamics controllers to have mobility and robustness similar to a human. During recent years, in order to develop a stable locomotion, several successful types of research have been introduced and can be generally divided into four categories: Central Pattern Generators (CPG), passive dynamics control, heuristic-based methods and model-based methods [1], [2]. CPG methods are known as biologically inspired methods which try to design locomotion using generating some rhythmic patterns for each limb. Indeed, they are generally composed of several oscillators which are connected together in a specific arrangement. Passive dynamics methods describe the behaviors of robots by their passive dynamics and without using any sensors or control. These methods describe walking by considering the center of mass in pendulum falling until ground reaction forces redirect this motion into the next step cycle. Heuristic approaches (e.g., genetic algorithms, reinforcement learning, etc.) are generally based on learning methods. To have acceptable performance, these approaches require a lot of training samples. Thus, the learning phase in these approaches takes a considerable amount of time. These approaches are not commonly suitable to apply on a real robot due to the high potential of damaging the hardware during the learning phase [2]. All the above approaches are fall beyond the scope of this paper. The main focus of this paper is on model-based methods.

In model-based approaches, a dynamics model of a robot is employed to generate reference trajectories of locomotion. In order to model the dynamics of robots, two different types of point of view are exist. In the first point of view, the whole body dynamics (true model) of a robot is considered and in the second point of view, the overall dynamics of a robot is approximated by a simplified model. Although several significant achievements have been achieved based on both perspectives but a trade-off should be considered to select perspective. For instance, a true dynamics model can (not always) provide more accurate results but these methods are not only computationally expensive but also their results are totally platform-dependent.

The rest of this paper is focused on the second point of view to show how a simplified model provides insight into the fundamental principles of humanoid locomotion. Moreover, some well-known simplified dynamics models of humanoid robots are reviewed and compared with each other. This paper is structured as follows: Section II gives an overview of related work. In the Section III, overall hierarchical architecture of a walking engine is presented, and each level of this structure are briefly explained. Moreover, formulation of the presented dynamics model in Section II are reviewed. Simulation results of the comparison are demonstrated in Section IV. Finally, conclusions and future research are presented in Section V.

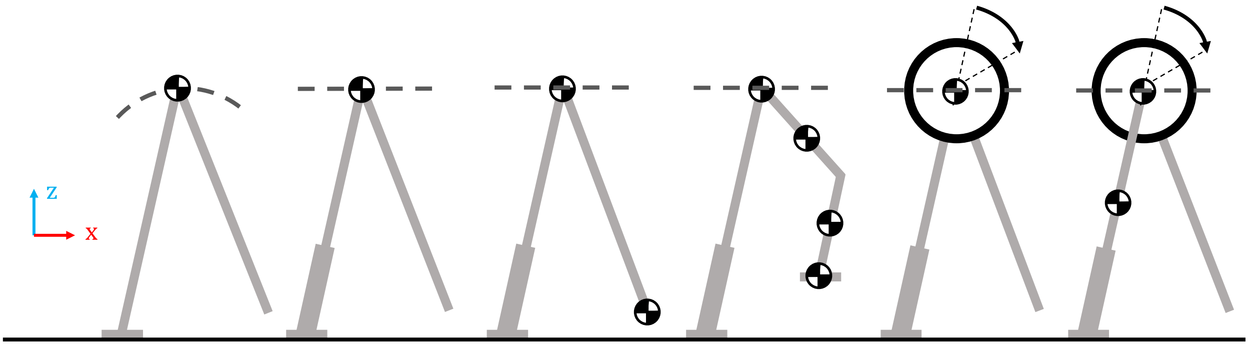


Fig. 1. Schematics of the presented simplified dynamics models in related work. In these schematics, gray links show the massless links which do not have any effects in the models. Black dashed lines indicate the trajectories of the COM. Name of the models from left to right are IP [3], LIPM [4], TMIPM [5], MMIPM [5], LIPPFM [6], Enhance LIPPFM [7].

## II. Related Work

The basic idea behind of using a simplified model instead of an exact model is organizing a complex system as a hierarchy. Generally, in hierarchical control approaches, a simplified model is used to determine the overall behaviors of the system in an abstract way and then by using a detailed full-body inverse dynamics controller, these behaviors can be converted to individual actuator inputs. [8]. It's obvious that the performance of the system depends on the ratio of matching between the template and the exact model.

Several simplified models have been proposed and Inverted Pendulum (IP) [3] is one of those which is computationally efficient and straightforward to understand. This model describes human dynamics in single support and provides a low-dimensional and physically-accurate model. In this model, the overall dynamics of a robot is approximated by a single mass which is connected to ground by a massless rod.

Kajita and Tani [4] extended this model by defining a height constraint to the horizontal plane of the system, this simplification not only causes to reduce the computational cost but also provides an appropriate framework to control. Indeed, this constraint causes the dynamics of the system becomes completely first-order linear dynamic system. Later, Kajita et. al. [9] introduced the Three Dimensional Linear Inverted Pendulum Model (3D-LIPM) and showed how this model can be used to generate walking in a 3D space. Afterward, In [10], a preview control method based on Zero-moment point (ZMP) was designed to control the system.

Albert, et. al. [5] proposed Two Masses Inverted Pendulum Model (TMIPM) which is an extended version of LIPM that considers the mass of swing leg in order to increase the gait stability. In their method, trajectories of COM have been generated using a linear differential equation according to a predefined ZMP and swing leg trajectories. They extended their model by considering the dynamic influence of the thigh, the shank and the foot of the swinging leg. Actually, the extended model composed of four masses and it has been named Multiple Masses Inverted Pendulum Model (MMIPM). Unlike LIPM, TMIPM and MMIPM do not have a direct solution because of dependency of motions of the masses to each other through the kinematic linkage, they proposed an iterative algorithm to define the trajectory of the torso. It should be noted that in their models, they considered the height of COM is a constant similar to LIPM.

Shimmyo, et. al. [11] proposed another dynamics model which was composed of three masses which were located on the base link, the right leg, and the left leg. In order to use preview controller for generating the walking trajectories, they assumed two assumptions which were Constant Mass Distribution and Constant Mass Height. They showed the effectiveness of their method by the experimental results.

In all of the above models, the upper body is considered as a single mass, since the body of a humanoid robot has several DoF (i.e. waist, arms, and neck) and their motions can generate a momentum around the COM. If this effect is considered, the ground reaction force will not pass through the COM. As a consequence, if a proper method to manage these momentums is not considered, the robot could not keep its stability and may fall down [7]. To cope with this issue, some extensions to the LIPM have been proposed that considered the angular momentum around COM [6], [12]. In [6], the legs of the robot are considered to be massless and extensible. Besides, to model centroidal angular momentum about COM, a flywheel (also called a reaction wheel) is used instead of a point mass (LIP Plus Flywheel Model or LIPPFM). According to this model, they proposed the capture point as well as capture region concepts which can be used to answer to this question: when and where to take a step while robot faces a massive magnitude push?. Later, Stephens [13] used this model to determine decision surfaces that could describe when a particular recovery strategy (e.g., ankle, hip or step) should be used to regain balance.

Kasaei, et. al. [7] proposed an enhanced version of LIPPFM and developed a reliable walking engine for biped robot based on this model. They released the height constraint of the COM and showed how this enhancement allows a more human-like motion and more stable walking. Latter, In [14], they extended their model by considering

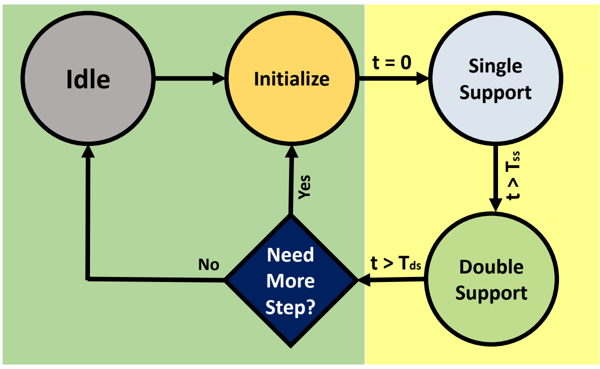


Fig. 2. Walking state machine with associated timer. Each state has a specific duration.

the mass of stance lag and showed the dynamics of the system could be represented using a first order differential equation by linearizing the model about the vertically upward equilibrium. Besides, they showed how this model could be used to plan and track the walking reference trajectories.

In the rest of this paper, a general hierarchical structure of biped walking will be presented and also we will explain how the presented models can be used to generate walking trajectories. Furthermore, the presented models will be compared together in a push recovery simulation scenario.

## III. Walking Engine

Walking is periodic locomotion which can be generated by repeating a series of steps and can be modeled using a state machine which is depicted in Fig. 2. As is shown in this figure, our walking engine composed of four distinct states which are Idle, Initialize, Single Support and Double Support. In the Idle state, the robot is standing in place, and no walking trajectories are commanded. During Initializing state, the robot is going to be ready to start walking by moving its COM from between its stance feet to the first support foot. During Single Support as well as Double Support states walking trajectories has been generated and commanded. Moreover, a timer has been associated with this walking state machine to trigger a state transition. The timer increases $t$, and it will be reset once it reaches the duration of double support state.

In addition to this state machine, a hierarchical architecture is used to fade the complexities of the controller. The overall architecture of this controller is depicted in Fig. 3. As is shown in this figure, it composed of four layers which will be described in the rest of this section.

### *A. Foot Step Planner*

This layer has three main tasks which are (i) generating a set of predefined foot positions (ii) generating the ZMP trajectories and (iii) generating the trajectories of the swing leg. All of these trajectories should be generated based on given step info and the predefined constraints (e.g., maximum step length, the minimum distance between feet, etc.). In our target, each step consists of two phases, single support, and double support and can be defined as follow:

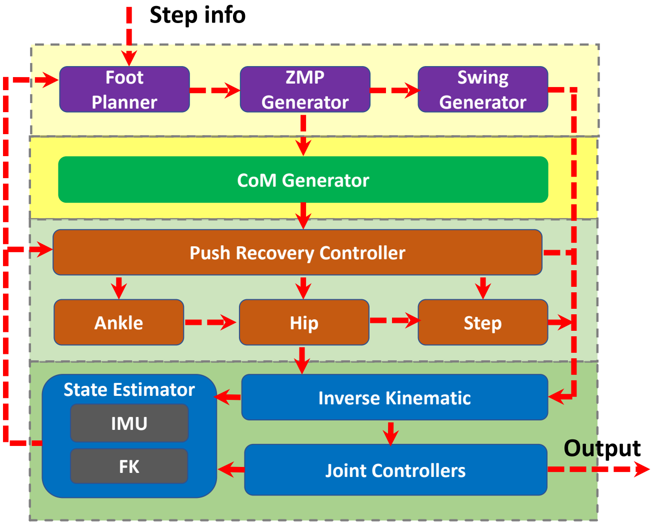


Fig. 3. Overall architecture of our hierarchical walking engine. It composed of four layers which is depicted by different colors.

$$\text{Step} \equiv \{L_{sx}, L_{sy}, T_{ss}, T_{ds}\} \tag{1}$$

where $L_{sx}$, $L_{sy}$, $T_{ss}$, $T_{ds}$, represent step length, step width, single support duration and double support duration respectively. These parameters should be selected based on the size of the robot, the capability of the robot and the tasks that the robot should perform. ZMP trajectories can be defined based on these parameters. The best intuitive choice for the ZMP trajectory during single support phase is the middle of the supporting foot, and it moves proportionally to the COM during double support phase. According to these assumptions, reference ZMP generator is formulated as follow:

$$r_{zmp} = \begin{cases} \begin{cases} f_{i,x} \\ f_{i,y} \end{cases} & 0 \le t < T_{ss} \\ \begin{cases} f_{i,x} + \frac{L_{sx} \times (t - T_{ss})}{T_{ds}} \\ f_{i,y} + \frac{L_{sy} \times (t - T_{ss})}{T_{ds}} \end{cases} & T_{ss} \le t < T_{ds} \end{cases} , \tag{2}$$

where $t$ represents the time which is reset at the end of each step ($t \geq T_{ss} + T_{ds}$), $T_{ss}$ , $T_{ds}$ are the duration of single and double support phases, respectively, $f_i = [f_{i,x} \quad f_{i,y}]$ is a set of predefined foot positions on a 2D surface ($i \in \mathbb{N}$). Thus, by determining these parameters and using Equation 2, reference ZMP trajectories can be generated.

After generating footsteps and ZMP trajectories, swing leg trajectories should be defined according to the dynamics model of the robot. In the case of considering mass less swing leg, these trajectories can be generated using arbitrary methods (e.g., polynomials, cubic spline, etc.). In other cases, these trajectories should be generated according to the dynamics model of the system.

### B. Gait Stability and COM Trajectories Generator

Several criteria for analyzing the balance of a humanoid have been proposed and Zero-moment point (ZMP) is one of the well-known approaches. Conceptually, ZMP is a point on the ground plane where the horizontal inertia and the gravity forces negate each other. Vukobratovic, et. al. [15] were the first ones that used ZMP as the main criterion to develop a stable walking for a humanoid robot. In case of no external forces or torques the ZMP can be defined using the following equation:

$$p_x = \frac{\sum_{i=1}^{k} m_i x_i(\ddot{z}_i + g) - \sum_{i=1}^{k} m_i z_i \ddot{x}_i}{\sum_{i=1}^{k} m_i(\ddot{z}_i + g)} \quad , \tag{3}$$

where $k$ represents the number of body parts which is considered in the dynamics model, $m_i$, $x_i$, $z_i$ represent the mass and positions of the $i_{th}$ body part.

Generally, trajectories of COM are generated based on the dynamics model of the system and some predefined trajectories such as ZMP and also the swing leg in case of considering the mass of the swing leg. In some of the presented dynamics models in Section II, an analytical solution exists to generate this trajectory (e.g., LIPM) and it can be counted as an important property of a dynamics model because it's not only straightforward but also computationally is cheap. In other cases which a direct solution is not feasible, these trajectories are generated based on either some assumptions like [11] or it can be formulated as an optimization problem which is generally expensive regarding computations cost. In the rest of this subsection, COM trajectories generators of the presented models in Section II are briefly summarized.

*a) LIPM:* According to the Equation 3 the dynamics model of LIPM is as follow:

$$\ddot{x} = \omega^2(x - p_x) \quad , \tag{4}$$

where $\omega = \sqrt{\frac{g}{z}}$ represents the natural frequency of the pendulum. This equation can be represented as a state space system:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega^2 \end{bmatrix} p_x \quad . \tag{5}$$

*b) TMIPM:* The dynamics of this model can be represented in state space form using the Equation 3:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\alpha & 1 \end{bmatrix} \begin{bmatrix} p_x \\ \beta \end{bmatrix}$$
$$\begin{cases} \alpha = \frac{g}{z_c} + \frac{m_c}{m_s \times z_c}(\ddot{z}_s + g) \\ \beta = \frac{m_c}{m_s \times z_c}(x_s \times (\ddot{z}_s + g) - \ddot{x}_s z_s) \end{cases} \quad , \tag{6}$$

where $x_s$, $z_s$ are the position of the swing leg in x and z-direction, $m_s$, $m_c$ represent the mass of swing leg and the remaining masses of the robot respectively.

*c) MMIPM:* The dynamics of the system is represented by the following differential equation:

$$\ddot{x} = \omega^2(x - p_x) + \underbrace{\sum_{i=1} \frac{m_i}{m_c \times z_c}\Big((x_i - p_x)(g + \ddot{z}_i) - \ddot{x}_i z_i\Big)}_{f(t)}, \tag{7}$$

where $z_c$ is the height of COM, $m_i$ represent the mass of $i_{th}$ part of the swing leg. As it explained before, there is not a direct solution for this model and in such situations, the trajectories of COM should be generated using an iterative algorithm. Thus for generating the COM trajectories, first, the system assumed as a TMIPM and generate the trajectories of the COM using the Equation 6 and predefined swing leg trajectories, then, based on a direct kinematic approach, the motions of $m_i$ are determined and then based on that motion $f(t)$ is calculated. This procedure executes while a $f(t)$ found that satisfies the condition.

*d) LIPPFM:* This model considers the momentum around the COM, and the equations of motion of this model can be represented using a first-order state space system as follow:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \omega^2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\omega^2 & -(mL)^{-1} \\ 0 & 0 \\ 0 & I_w^{-1} \end{bmatrix} \begin{bmatrix} p_x \\ \tau_w \end{bmatrix}, \tag{8}$$

where $m$ is the mass of flywheel, $L$, $I_w$ and $\tau_w$ represent the length of the pendulum, the rotational inertia of flywheel around flywheel center of mass and the flywheel torque respectively.

*e) Enhanced LIPPFM:* In this model, the accuracy of the model has been improved by releasing the constraint on COM's height as well as considering the mass of pendulum in the model.

$$\begin{bmatrix} \dot{\theta}_a \\ \ddot{\theta}_a \\ \ddot{\theta}_w \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\mu \times (g + \ddot{Z}_c)}{\gamma} & 0 & 0 \\ \frac{-\mu \times (g + \ddot{Z}_c)}{\gamma} & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_a \\ \dot{\theta}_a \\ \dot{\theta}_w \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{\gamma} & \frac{-1}{\gamma} \\ \frac{-1}{\gamma} & \frac{\gamma + I_w}{\gamma \times I_w} \end{bmatrix} \begin{bmatrix} \tau_a \\ \tau_w \end{bmatrix}$$
$$\begin{cases} \gamma = M \times L^2 + I_p \\ \mu = m \times l + M \times L \end{cases} \quad , \tag{9}$$

where $\theta = [\theta_a \quad \theta_w]^\top$ is a vector of pendulum and flywheel angles respecting to the vertical axis. $M$ and $m$ are the masses of flywheel and the pendulum, $L$ and $l$ are the lengths from the base of the pendulum to flywheel center of mass and to pendulum center of mass respectively. $g$ describes the gravity acceleration, $\ddot{Z}_c$ represents the acceleration of COM in Z-direction, $I_p$ is rotational inertia of pendulum about the base of pendulum and $I_w$ represents rotational inertia of flywheel around flywheel center of mass.

### C. Push Recovery Strategies

A Feed-forward walking can be developed based on the trajectories of the ZMP and COM but this type of walking is not robust enough in facing with unexpected errors which can be raised from several sources, such as external disturbances, inaccurate dynamic model and etc. For instance, during walking on rough terrain environments, several forces will be applied to the robots. Hence, to keep the stability of the robot during walking, several criteria have been defined and the most important one is keeping the ZMP inside a polygon that is defined by the foot or feet touching the ground (support

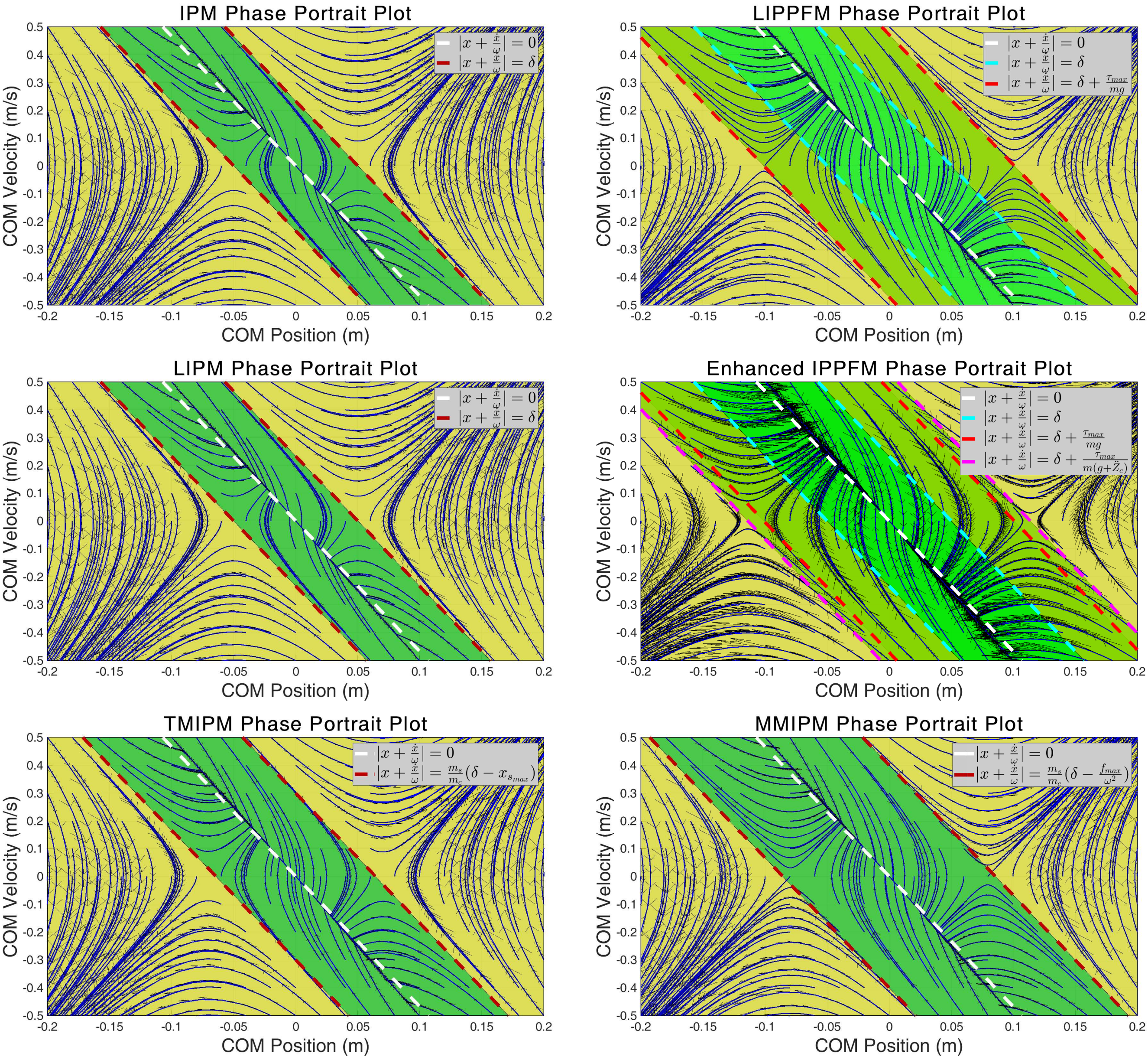


Fig. 4. Push recovery evaluation results. Yellow regions represent unstable regions where robot should take a step to regain its stability. Green (including light and dark) regions represent the stable regions which mean robot is able to regain its balance. Dash lines show the border of each recovery strategies.

polygon). Human uses three distinctive actions including ankle, hip, and step recovery strategies to provide chances to regain balance. Ankle strategy tries to keep the stability of the robot by applying compensating torques at the ankle. Although this strategy improves the stability of the robot in some situations robot should use joints of the waist and the hips to prevent falling (hip strategy). In the case of significant disturbances, the stability of the robot can not regain using these strategies and robot should take a step.

### D. Low Level Controller

This level consists of three main modules including state estimator, inverse kinematics solvers, and joints position controller. The main task of this level is estimating position, velocity, and acceleration of the COM using IMU data that is mounted on torso or hip of the robot, and also a forward kinematic model of the robot which uses the values of the joint encoders.

## IV. Simulation Results and Comparison

All of the presented models are able to generate walking for a humanoid robot in a general walking scenario but some of them are able to provide in a more stable manner. In order to compare the performance of these models, a push recovery simulation scenario has been defined. The goal of this scenario is examining the ability of models concerning regaining balance in different situations. In these simulations, the robot is considered to be in single support phase and start from a specified initial condition $(x_0, \dot{x}_0)$

TABLE I
PARAMETERS USED IN THE SIMULATIONS.

| name | description | value | min | max |
|---|---|---|---|---|
| $m_c$ | mas $(kg)$ | 7.00 | | |
| $m_1$ | mas of thigh$(kg)$ | 1.50 | | |
| $m_2$ | mas of shin $(kg)$ | 1.50 | | |
| $m_3$ | mas of foot $(kg)$ | 0.50 | | |
| $Z_c$ | height of COM $(m)$ | 0.45 | 0.40 | 0.50 |
| $L_0$ | length of pendulum $(m)$ | 0.50 | | |
| $L_1$ | length of thigh $(m)$ | 0.28 | | |
| $L_2$ | length of shin $(m)$ | 0.28 | | |
| $\delta$ | length of foot $(m)$ | 0.10 | | |
| $\tau_w$ | flywheel torque $(N/m)$ | 0.00 | -5.00 | 5.00 |
| $\ddot{Z}_c$ | Acceleration in Z-direction $(m/s^2)$ | 0.00 | -0.07 | 0.07 |

and robot should regain its stability without taking a step. According to the results of these simulations, we can find a specific answer for each model to this question: *"when and which strategy(s) should be used to avoid falling?"*. Moreover, these numerical simulations allow the validation of the proposed formulations for each dynamics model. These simulations have been performed using MATLAB, and the most important parameters of the simulated robot as well as their ranges are shown in Table I.

For each dynamics model, a set of simulations have been run according to the set of initial parameters assumed for the simulated robot. The simulation results are depicted in the plots of Fig. 4. In these plots, each curve shows the result of a single simulation run. For each simulation, the simulated robot is started from single support with a specified initial condition and simulation. The initial condition is selected over the range of [-0.2 0.2] at interval $0.02m$ for $x_0$ and [-0.5 0.5] at interval $0.1m/s$ for $\dot{x}$ ( for each model 231 simulations were conducted ). Also, as is shown in the plots, yellow regions show the unstable regions which mean robot could not keep its stability and green regions show stable regions which mean robot is able to regain its balance.

## V. CONCLUSION

This paper presented a comparative study of some well-known dynamics model for balance recovery in a humanoid robot. This study was started by introducing some dynamics models and brief explaining of how they generate the COM trajectories according to the input step parameters. Moreover, An overall architecture of a walking engine was presented to explain how the generated trajectories use to produce walking. To validate the formulation and also compare them together, a set of simulations have been carried out using MATLAB. The results of the simulations are depicted in Fig. 5. As this figure shows, Enhanced IPPFM is the ablest model to keep the stability of the robot even in very challenging conditions. In future work, we would like to involve more complex dynamics models and also consider the stepping strategy in our comparisons.

## ACKNOWLEDGMENT

This research is supported by Portuguese National Funds through Foundation for Science and Technology (FCT) through FCT scholarship SFRH/BD/118438/2016.

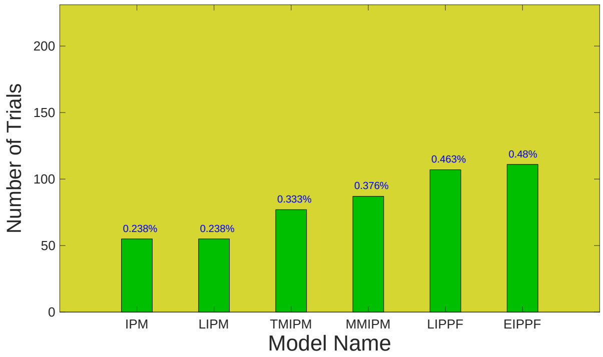


Fig. 5. Summary of the simulation results.

## REFERENCES

[1] H. Picado, M. Gestal, N. Lau, L. Reis, and A. Tomé, "Automatic generation of biped walk behavior using genetic algorithms," *Bio-inspired systems: Computational and ambient intelligence*, pp. 805–812, 2009.

[2] M. Kasaei, N. lau, A. Pereira, and E. Shahri, "A reliable model-based walking engine with push recovery capability," in *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, April 2017, pp. 122–127.

[3] H. Hemami and C. Golliday Jr, "The inverted pendulum and biped stability," *Mathematical Biosciences*, vol. 34, no. 1-2, pp. 95–110, 1977.

[4] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 1405–1411.

[5] A. Albert and W. Gerth, "Analytic path planning algorithms for bipedal robots without a trunk," *Journal of Intelligent and Robotic Systems*, vol. 36, no. 2, pp. 109–127, 2003.

[6] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 200–207.

[7] S. M. Kasaei, N. Lau, and A. Pereira, "A reliable hierarchical omnidirectional walking engine for a bipedal robot by using the enhanced lip plus flywheel," in *Human-centric Robotics-Proceedings Of The 20th International Conference Clawar 2017*. World Scientific, 2017, p. 399.

[8] S. Faraji and A. J. Ijspeert, "3lp: a linear 3d-walking model including torso and swing dynamics," *arXiv preprint arXiv:1605.03036*, 2016.

[9] S. Kajita, O. Matsumoto, and M. Saigo, "Real-time 3d walking pattern generation for a biped robot with telescopic legs," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 2299–2306.

[10] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1620–1626.

[11] S. Shimmyo, T. Sato, and K. Ohnishi, "Biped walking pattern generation by using preview control based on three-mass model," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 11, pp. 5137–5147, 2013.

[12] T. Komura, H. Leung, S. Kudoh, and J. Kuffner, "A feedback controller for biped humanoids that can counteract large perturbations during gait," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 1989–1995.

[13] B. Stephens, "Humanoid push recovery," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE, 2007, pp. 589–595.

[14] M. Kasaei, N. Lau, and A. Pereira, "An optimal closed-loop framework to develop stable walking for humanoid robot," in *Autonomous Robot Systems and Competitions (ICARSC), 2018 IEEE International Conference on*. IEEE, 2018, pp. 30–35.

[15] M. Vukobratovic, A. Frank, and D. Juricic, "On the stability of biped locomotion," *Biomedical Engineering, IEEE Transactions on*, no. 1, pp. 25–36, 1970.

# An Overview on Bidirectional Transformations

José Nuno Macedo
*Department of Informatics*
*University of Minho*
Braga, Portugal
ze_nuno_eu@hotmail.com

***Abstract*—Bidirectional transformations are a way to maintain consistency between various related sources of information. From Model-Driven software, to relational databases and Domain-Specific Languages, there are various applications of this technology that provide a stable and reliable methodology and set of tools to solve problems in these areas.**

**This report presents bidirectional transformations, providing an insight into this methodology. Some approaches to practical software development using this technique are also described.**

***Index Terms*—bidirectional transformation, synchronization, structured programming, model update**

## I. Introduction

Software development has evolved considerably over the last decades. The problems presented have become increasingly complex, with increasingly complex solutions arising to combat them. Open-source development, several frameworks, alternative programming languages and paradigms and software testing are some solutions for the clutter and confusion that complex projects can create, bringing some stability to the chaos of complex software. Some of these tools, such as frameworks, help by guiding the developer in the right direction, possibly being more restrictive while doing so. Some improve development experience by providing tools to detect errors or oversights that can and will happen in large projects, but can be attenuated through collaborative development and the use of software testing techniques.

While most of these tools help the project become more robust, the availability of formally proven tools is still lackluster - there is no mathematical approach that can, for most frameworks and software testing, guarantee correctness. Therefore, several times, the security provided by these tools is an illusion, allowing the developer to be more daring in their approach with no scientific backing.

Several problems in software engineering are based on maintaining several data structures that must be consistent in their behaviour and content. Consistency between the several data structures can be provided by a third-party program, thus increasing the burden on the developer due to the need to develop more software to complement the original intention. This can be entirely avoided by using an approach that deals with these consistency needs automatically. Bidirectional transformations are a mechanism to improve the development around such problems.

This reports aims to introduce bidirectional transformations as a concept, supported by some examples and the description of practical approaches to this concept. In section II, the concept of bidirectional transformations is introduced, along with some examples that illustrate possible applications of this technique. In section III, a practical approach, known as lenses, is approached. In section IV, *BiGUL*, a putback-based bidirectional programming language, is briefly introduced. Section V concludes this overview, pointing to more relevant existing work.

## II. Bidirectional Transformations

A bidirectional transformation model (from now on referred as BX) contains different representations of shared data. When any representation of the shared data is modified, all of the representations are modified as well, reflecting this change accordingly. This results in a permanently synchronized environment, where every representation of data is consistent with the others. In this report, the focus will be set on the binary case, that is, when two different representations of data need to be synchronized. This is a general case present in various software projects, and BX is applicable in most of them.

### *A. Converting Data*

A BX is an interesting approach for data conversion. Assuming there are two different representations of the same data that need to be consistent, a BX can be used to maintain the consistency. An example of this is a code editor, with syntax checking and automatic correction of common errors. In this example, there are two different representations of the code written by a user. The most obvious is the textual representation, that is, the code that the user edits. There is, however, a different representation, which is the internal representation of said code.

This internal representation is needed as there can be a lot of redundant and useless information in the code the user writes. For example, white spaces, indentation and comments are typically all useless in terms of the actual program being written. Of course, a code editor might want to not ignore comments and format them appropriately, but the fact that some redundant information remains is still valid. At the same time, the internal representation is usually in a different format than the external one. Since the code is being parsed, it is expected that an *Abstract Syntax Tree* is being generated. This is a tree where each node represents an instruction or segment of the program, being fundamentally different from the external representation as the data is not a simple string anymore, being instead an appropriate data type for the information stored. Figure 1 represents the behaviour of the code editor.

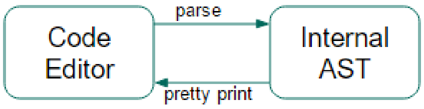


Fig. 1. How the code editor behaves.

When an user writes code in the code editor, it is expected that the code editor will, in turn, detect any mistakes and produce a warning message. When this happens, a change that was applied in the external representation must be reflected appropriately in the internal representation so as to maintain consistency. If consistency is not kept, it can be possible that an error in the code is not flagged by the code editor, or that a correct line of code is flagged as incorrect. Unless the code editor regularly checks for consistency faults or rebuilds the whole structure, the two data representations will stop being synchronized, resulting in an unpleasing experience for the user, where the software they use sabotages the workflow instead of providing valuable feedback.

Typically, this type of applications are developed using two separate tools, one for each direction, that is, in this example, a *parser* and a *pretty printer* are developed separately, and then integrated into the main software. A BX can simplify the development process of such tools by providing a consistent approach for this development. Some tools for BX development also provide security in terms of assuring the synchronization of the generated BX.

### B. *Database views and updates*

The view-update problem on databases [1] was one of the earliest problems to be studied as a BX, albeit the terminology used was different. For this problem, consider an extremely complex database representing the employees of a company, and the corresponding projects of each. There are various ways of consulting such database, some more complex than others. It is possible to look at one table individually, to look at several tables one at a time, to join various tables that share common values.

In this example, an *Employees* table contains the name and details of the employees of the company. Naturally, a big volume of information is contained in this table, such as address, phone number, e-mail, accounting details, among others. A very simplified version of this table is presented in Table I.

TABLE I
EMPLOYEES TABLE

| Id | Name | Age |
|---|---|---|
| 1 | John | 32 |
| 2 | Jack | 27 |

In the same database, another table, designated the *Projects* table, contains information relative to projects which are being developed at said company. Examples of data that can be found in this table include budget, expected finish date, starting date, infrastructure, employees involved, status of the project. A very simplified version of this table is presented in Table II.

TABLE II
PROJECTS TABLE

| Id | Employee | Status |
|---|---|---|
| 1 | 1 | Review |
| 2 | 1 | Evaluation |

By joining the two tables, a new, more complete table, is created. This table is created by uniting the two previous tables, matching the lines where the employee *ID* is equal. In it, it is easier to see which employee is working on which project. Thus it can be concluded, on Table III, that John is working in projects 1 and 2.

TABLE III
UNION OF THE EMPLOYEES AND PROJECTS TABLES

| Project_Id | Employee | Status | Age |
|---|---|---|---|
| 1 | John | Review | 32 |
| 2 | John | Evaluation | 32 |

Table III can be read as a *view* of the *source*, which is the entire database, containing tables I and II. For complex databases, a view is necessary to generate a readable table, as various tables have to be connected in order to form a readable result. However, this view poses a problem.

If the user performs any type of changes in the view, then those changes need to be immediately reflected on the database accordingly. This is because the consistency must be kept, but, more important than that, because otherwise the changes the user performs could be lost.

In this example, if the user deletes the name *John* from both entries, there is no well-defined behaviour for the BX. In the *Projects* table, projects could be kept with null entries for the employees or the projects could be kept and the status changed to *Cancelled*. In the *Employees* table, *John* could be removed, representing that he was fired, or he could be kept, and perhaps relocated to a different project or section of the company.

The correct behaviour must be specified beforehand, in a way that assures consistency between user actions. The propagation of an update from the view to the source must also be correct, such that it does not destroy the database in any way.

### C. *Development based on Software Models*

When developing software, a possible approach is to create a model representing the part of the software that is to be worked on. As such, a layer of abstraction is created, and the developer can focus on just the important aspects, present in the model, while abstracting the irrelevant aspects, which are hidden by the model. In this situation, BX are relevant as the model is a view of the original software, which is the source. As such, changes applied to the model should be reflected on the original software appropriately, and vice versa.

A software product can have several different models related to it. They can be different representations of the same component, or different components. However, the purpose is always to abstract. The updates to be performed on the original software when a model is changed can be extremely complex in some cases, as some models, through abstraction, create various layers of complexity in the process of actually applying the changes to the original software.

One of the most commonly known examples where this can be applied is the object-relational mapping (ORM) technique. According to it, in object-oriented programming languages, the software should not access the database directly, opting instead for having a layer for communication with the database. This layer contains various objects that represent parts of the database, and all of the communications with it are described inside, such that a neat and clean interface is provided for the communication. In a sense, these objects are models of parts of the database. The developer does not have to be concerned with how the database works or responds, he only has to interact with the simple-by-design objects.

## III. Lenses

### A. Definition

While there are several techniques for approaching BX in terms of development, this report will shift the focus into the lenses approach. This is an approach that has gained a lot of traction in recent years [2] [3] [4]. An asymmetrical lens consists of two functions, given a source $S$ and a target $V$:

$$get : S \to V \tag{1}$$

$$put : S \times V \to S \tag{2}$$

The *get* function, given a source, produces a corresponding view, such that *get s* would produce a view *v*. The *put* function, given the old source and an updated view, updated the source with the changes applied to the view, such that *put(v',s)* would produce a new, updated source. It is asymmetrical due to the fact that the view determined by the source.

For the database example, a view can be a certain join of two tables, obtainable by a *get* function. The *put* function then updates the database with the updated table.

Additionally, a lens is *well-behaved* when it satisfies:

$$\forall s \quad put(s, get\ s) \quad = s \tag{3}$$

$$\forall s, v \quad get(put(s, v)) \quad = v \tag{4}$$

The law 3 is known as the *GetPut* law. It states that putting a view as it was taken from the source does not change it. This is a desired behaviour in various applications of BX- it is expected that, if a joined table from a database is unchanged, then the database itself should remain unchanged. This property is sometimes referred as a hippocratic property, that is, a property that prevents unnecessary harm in the system.

The law 4 is known as the *PutGet* law. According to it, when putting a view into the source, and then taking a view from the source, the resulting view should be equal to the original view. In some situations, it is desired, in others, not so much. Putting an invalid view into the database, for example, can result in ignoring the put command, and therefore performing a get afterwards will yield different results. In the database example, inserting an employee with a negative age can be considered an invalid put, and therefore discarded. However, when assuming that the view to put into the source is valid, then this property is also generally desired. It is sometimes referred as a correctness property, as it guarantees that the put action is performing desired results.

A well-behaved lens is *very well-behaved* when it satisfies:

$$\forall s, v, v' \quad put(put(s, v), v') = put(s, v') \tag{5}$$

The property 5 is known as the *PutPut* property. According to it, putting a view into the source and then immediately putting another view in the resulting has the same effect as only putting the last view into the source. This happens when putting a new view overwrites the results of putting a previous view.

There are other types of lenses besides symmetrical lenses, such as symmetric lenses [5], edit lenses [6] and matching lenses [7].

### B. Practical Approach

There have been various practical approaches to BX systems in programming over the last few years. The focus, however, will be set in a lens library for the Haskell programming language [8]. This library provides a set of tools for building and manipulating lenses in Haskell. This set of tools allows the developer to very easily manipulate complex data structures, by simplifying the access of these data structures through lenses.

In practice, this library provides ways to build *put* and *get* functions adequately for each problem. In Listing 1, several examples of code are presented.

Listing 1. Examples of lenses using this library

```
source = (”hello”,(”world”,”!!!”))

get_view1 = ^._1
get_view2 = ^._2._1

view1 = get_view1 source
view2 = source^._2._1

put_view1 = set _1 42
put_view2 = set (_2._1) 42

update1 = put_view1 source
update2 = set (_2._1) 42
            (”hello”,(”world”,”!!!”))
```

In this block of code, a source is declared, according to the type $S\times(S\times S)$, where $S$ is a text string. In fact, this is a simple source, but it is used for demonstration purposes. Two *get* functions are defined, namely *get_view1* and *get_view2*. The ˆ operator indicates a specification of a *get*, and the following argument is the position of the record that is related to this operation. In *get_view1*, *_1* is used to specify the first record, and in *get_view2*, *_2._1* is used to specify the first record inside the second record.

The *view1* and *view2* constants represent views, calculated from applying the *get* functions to the source. Two alternative but equivalent definitions are shown in these definitions. The constant *view1* contains the first record, that is, *"hello"*, while *view2* contains the first record inside the second record, that is, *"world"*.

For the definition of *put*, two examples are also displayed. The function *put_view1* puts the value *42* in the first record, and the function *put_view2* puts the value *42* in the first record inside the second record. As such, applying *put_view1* onto the source produces a new source of value *(42,("world","!!!"))*, and applying *put_view2* onto the source produces a new source of value *("hello",(42,"!!!"))*.

It is important to note that Haskell is a strongly-typed programming language. As such, it is not always trivial to change the type of the data in a simple way. However, the lenses provided in this library are flexible in this sense, as they allow the developer to either create simple lenses that preserve the type of information, or slightly more complex lenses that can modify the type of information while still abiding to the same rules and operators that are used for the simple lenses. In fact, lenses can also be composed, and as such, a complex lens for a complex source can be defined as a composition of various simple lenses, which are easy to understand and debug individually, but act as building blocks of a powerful tool.

For more complex data types, this library allows for the use of a Template Haskell construct to automatically derive adequate lenses. In practice, this results in efficient development cycles for the developer, as manually creating lenses for complex data structures can be time-consuming and confusing. This library is open-source and well documented, with hundreds of examples fit to various different problems.

## IV. BiGUL

Putback-based bidirectional programming is an approach to bidirectional programming in which part of the development cycle is automatically generated by construction. In fact, it defines that, for a BX to be properly defined, only the putback, that is, the *put* function, must be defined, and the *get* function can be automatically derived from it. The opposite is not true - given a *get* function, there are several ways to define the *put* function.

Considering the previous example of the database, when a view, that is, a joined table, is changed, by deleting an employee, there are several ways to reflect that change on the database. However, given a definition on how to reflect the change on the database, the reverse process, that is, getting the information from the database, is unique.

*BiGUL* (Bidirectional Generic Update Language) [9] is a putback-based bidirectional programming language, complementar to the Haskell programming language. It is formally verified with the Agda [10] programming language, therefore guaranteeing that any putback transformation written in BiGUL is well-behaved, that is, that they obey the prepositions 3, 4 and 5.

While *BiGUL* is an elaborate tool that allows for the development of complex and deep systems, it is not as user-friendly as the lenses library presented in subsection III-B. As such, this overview shall only cover some constructs present in it briefly, while still encouraging readers to read upon and experiment with this tool [11].

The following constructs are building blocks available in the *BiGUL* language, for the definition of *put*:

- ***Replace*** - Replaces a value in the source with the provided value in the view.
- ***Skip*** - Ignores the value in the view, thus keeping the source intact.
- ***Fail*** - Fail, producing an error message. Useful for defining incorrect behaviour.
- ***CaseS*** - Produce a case statement, similar to the *switch* statement in several imperative languages, where the source is matched against conditions until one is matched, and then the associated code is executed.
- ***CaseV*** - Similar to *CaseS*, but matching against the view.
- ***RearrS*** - Rearrange the source into a more desirable intermediate data representation. Useful for facilitating some more complex matches between structures that differ between the source and view.
- ***RearrV*** - Similar to *RearrS*, but rearranging the view.
- ***Align*** - Match a list of information in the view with a more complex list of information in the source, therefore describing how to perform a *put* into a more complex structure.
- ***Update*** - Syntactic sugar, allows the use of pattern matching to simplify the usage of some of the previous constructs.

Having these constructs defined, complex *put* operations can be defined by composing them adequately, according to the syntax of *BiGUL*. By using them, the developer has a formal guarantee that the BX is well-behaved, as well as only having to specify part of the program and having the rest be generated “for free”.

The developer can also force the usage of a self-defined *put*, instead of using the available tools to build one. While this may allow for an easier definition of *put*, it also means that there is no formal backing or automatic generation of *get*, so the advantages of the tool are lost. However, if there is a guarantee that the supplied *put-get* pair is correct, then this approach can facilitate the development of some programs where some components can be difficult to express in *BiGUL*, and therefore be expressed directly instead.

## V. Conclusions

In this report, bidirectional transformations are presented as an underlying problem to several software projects, and at the same time, as the solution. In fact, thinking of software problems as bidirectional transformations can provide more insight on how to adequately approach them, using techniques that are safe and simple to use when correctly employed.

Some contextualization is provided through the data conversion, database updates and software model development examples. However, it is important to keep in mind that bidirectional transformations are not stuck in these areas - these are merely examples of a more generic approach to synchronization problems.

Contextualized into bidirectional transformations, the lenses approach is a recent and interesting approach to solving complex problems. In fact, it is an abstract concept, which has been implemented into, among others, the library described in subsection III-B.

*BiGUL*, on the other hand, implements the concepts of bidirectional transformations into a concrete programming language, supported by formal backing and automated program generation. As opposed to the lenses library, which is a complementary library for the Haskell programming language, *BiGUL* describes a standalone language.

There is plenty of work in the area of bidirectional transformations, ranging from work on XML Schemas [12] to applications of this technique to databases. The refinement of this approach represents an important step in the evolution of synchronization approaches for software problems, as it proposes interesting approaches to problems that are relatively common in software development but not always correctly handled.

## References

[1] F. Bancilhon and N. Spyratos, "Update semantics of relational views," *ACM Trans. Database Syst.*, vol. 6, no. 4, pp. 557–575, Dec. 1981. [Online]. Available: http://doi.acm.org/10.1145/319628.319634

[2] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt, "Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem," *ACM Trans. Program. Lang. Syst.*, vol. 29, no. 3, May 2007. [Online]. Available: http://doi.acm.org/10.1145/1232420.1232424

[3] A. Bohannon, B. C. Pierce, and J. A. Vaughan, "Relational lenses: A language for updatable views," in *Proceedings of the Twenty-fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '06. New York, NY, USA: ACM, 2006, pp. 338–347. [Online]. Available: http://doi.acm.org/10.1145/1142351.1142399

[4] A. Bohannon, J. N. Foster, B. C. Pierce, A. Pilkiewicz, and A. Schmitt, "Boomerang: Resourceful lenses for string data," in *Proceedings of the 35th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '08. New York, NY, USA: ACM, 2008, pp. 407–419. [Online]. Available: http://doi.acm.org/10.1145/1328438.1328487

[5] M. Hofmann, B. Pierce, and D. Wagner, "Symmetric lenses," *SIGPLAN Not.*, vol. 46, no. 1, pp. 371–384, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1925844.1926428

[6] ——, "Edit lenses," in *Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '12. New York, NY, USA: ACM, 2012, pp. 495–508. [Online]. Available: http://doi.acm.org/10.1145/2103656.2103715

[7] D. M. Barbosa, J. Cretin, N. Foster, M. Greenberg, and B. C. Pierce, "Matching lenses: Alignment and view update," in *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming*, ser. ICFP '10. New York, NY, USA: ACM, 2010, pp. 193–204. [Online]. Available: http://doi.acm.org/10.1145/1863543.1863572

[8] E. A. Kmett, "Lenses, folds and traversals," https://github.com/ekmett/lens, 2018.

[9] H.-S. Ko, T. Zan, and Z. Hu, "Bigul: A formally verified core language for putback-based bidirectional programming," in *Proceedings of the 2016 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*, ser. PEPM '16. New York, NY, USA: ACM, 2016, pp. 61–72. [Online]. Available: http://doi.acm.org/10.1145/2847538.2847544

[10] U. Norell, "Towards a practical programming language based on dependent type theory," 2007.

[11] Z. Hu and H.-S. Ko, *Principles and Practice of Bidirectional Programming in BiGUL*. Cham: Springer International Publishing, 2018, pp. 100–150. [Online]. Available: https://doi.org/10.1007/978-3-319-79108-1_4

[12] Z. Hu and J. de Lara, Eds., *Theory and Practice of Model Transformations - 5th International Conference, ICMT 2012, Prague, Czech Republic, May 28-29, 2012. Proceedings*, ser. Lecture Notes in Computer Science, vol. 7307. Springer, 2012. [Online]. Available: https://doi.org/10.1007/978-3-642-30476-7

# Heart Failure Prediction Using Real Data Processed by Machine Learning Techniques

1st Daniel Badran
*dept. of Mechanics and Industrial Management*
*Faculdade de Engenharia da Universidade do Porto*
Daniel_badran@hotmail.com

***Abstract*—In the past decade, the health industry has been producing huge amounts of data that could be used to aid doctors diagnose or even predict future possible illnesses. The purpose of this paper is to describe a project that creates a linear regression model using the historical relationship between a dependent variable and multiple explanatory independent variables, to predict the future of the dependent variable for a given duration of time, for different patients. The model and the prediction are established using Gretl; precisely using time series model. Later on, we construct a comparison table between different algorithms used in the study, to deduce which one is the most accurate after testing each one in MATLAB.**

***Index Terms*—Heart failure, machine learning, Coronary Heart Disease, Linear Regression, prediction**

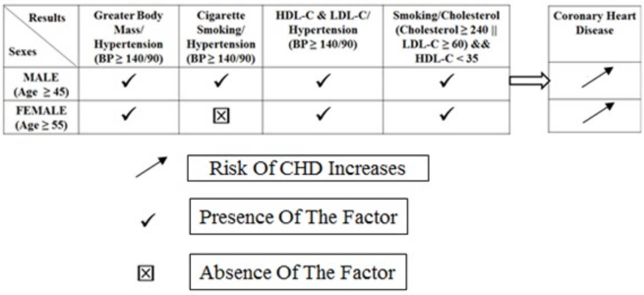
## I. Introduction

Coronary Heart Disease (CHD), is characterized by a wax-like substance called plaque that cumulates up inside the coronary arteries; this disease can be caused by age, a bad diet or genetic factors [1]. These arteries supply oxygen-rich blood to your heart muscle. When plaque builds up in the arteries, the condition is called atherosclerosis. The buildup of plaque occurs over many years.CHD, specifically cardiovascular diseases (CVDs) are the number one cause of death globally; more people die annually from CVDs than any other cause. Over 17.6 million people died from CVDs by 2012, where that number was maintained at 17.7 million in 2015, representing 31.43% of all global mortality. [2] Out of these deaths, an estimated 7.4 million were due to Coronary Heart Disease (CHD) and 6.7 million were due to strokes. Most cardiovascular diseases can be prevented by addressing behavioral risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population-wide strategies. Patient's cardiovascular disease or ones that are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidemia or already established disease) need early detection and management using counseling and medications. The paper is structured as follows, Section II will be dedicated to illustrating and explaining previous work to give a feel where this idea came from and to present the thesis. Section III will present our proposed method as well as our approach to resolving this problem. Section IV will feature all the experimentations and the obtained results.

## II. Related Work

### *A. Prediction of Coronary Heart Disease Using Risk Factor Categories*

The main idea behind the study was to use seven risk factors; Low-Density Lipoprotein cholesterol, High-Density Lipoprotein cholesterol, sex, age, diabetes, smoking, and blood pressure). Factors such as obesity left ventricular hypertrophy, family history of premature coronary heart disease and estrogen replacement therapy have been taken into consideration, as input in order to create prediction algorithms using regression models formed by linear and logistic regressions to forecast CHD risk, for a middle-aged population. The following table in figure 1 illustrates the obtained results.



| Results / Sexes | Greater Body Mass/ Hypertension (BP ≥ 140/90) | Cigarette Smoking/ Hypertension (BP ≥ 140/90) | HDL-C & LDL-C/ Hypertension (BP ≥ 140/90) | Smoking/Cholesterol (Cholesterol ≥ 240 \|\| LDL-C ≥ 60) && HDL-C < 35 | Coronary Heart Disease |
|---|---|---|---|---|---|
| MALE (Age ≥ 45) | ✓ | ✓ | ✓ | ✓ | ↗ |
| FEMALE (Age ≥ 55) | ✓ | ☒ | ✓ | ✓ | ↗ |

↗ Risk Of CHD Increases

✓ Presence Of The Factor

☒ Absence Of The Factor

Fig. 1. Study's Obtained Results.

### *B. Reduced Number of Circulating Endothelial Progenitor Cells Predicts Future Cardiovascular Events*

The objective of this project was to study over a period of 10 months using multiple cardiovascular events (Cardiovascular death, unstable angina, myocardial infarction, PTCA Percutaneous Transluminal Coronary Angioplasty: Procedure to open up blocked coronary arteries, CABG Coronary Artery Bypass Grafting: Surgery that improves blood flow to the heart, or ischemic stroke) to predict CHD risk. [3] To start the test EPCs were marked by CD34+KDR+; then the analysis was established using a normal distribution with the Kolmogorov-Smirnov fit test then compared by the Mann-Whitney U test using ANOVA. Comparison of categorical variables was generated by the Pearson $X^2$ test. The obtained results demonstrated the following:

- Documented Coronary Artery Disease (CAD) had higher number of risk factors.
- Reduced level of EPC is a surrogate marker of vascular function, and cumulative risk prediction and an identifier for future CHD risk.

### *C. Coronary Heart Disease Prediction From Lipoprotein Cholesterol Levels, Triglycerides, Lipoprotein (a), Apolipoproteins A-I and B, and HDL Density Subfractions*

The main idea behind this project was to predict coronary heart disease from cholesterol levels and HDL density subfractions using Cox Proportional Hazards Regression Analysis, precisely using RR model. Analyses were performed separately; divided by sex. Mean lipid values were calculated for participants with and without incident CHD after age and race adjustment. The lowest CHD risk was found in the lowest LDL-C quintile, in women and men, and CHD risk accelerated with increasing values of LDL-C for both sexes. [4]

TABLE I
OBTAINED RESULTS

| CHD Risk Level | LDL-C Level | Sex |
|---|---|---|
| High | High | M-F |
| High | High | M-F |
| Low | Low | M-F |

### *D. Combination of Data Mining Methods With New Medical Data To Predict the Outcome Of Coronary Heart Disease*

The goal of this study is to develop a data mining algorithm for predicting survival of the CHD patient. The analysis was performed during follow-ups using 3 data mining prediction models. The first being Support Vector Machine (SVM) with 10 folds cross-validation algorithm, which turned out to be the most accurate algorithm, the second model is, an artificial neural network using multi-layer perceptron with backpropagation, and the final model was decision trees. The results were divided into 3 major factors, accuracy, sensitivity and specificity. [5]

TABLE II
THIS STUDY'S RESULTS

| Model | Accuracy Sensitivity Specificity |
|---|---|
| SVM | 92.1% ; 92.87% ; 89.11% |
| ANN | 91.0% ; 91.73% ; 88.12% |
| Decision Tree | 89.6% ; 90.98% ; 84.16% |

## III. PROPOSED METHOD

### *A. Proposed Solution*

Since the prediction will be based on 21 factors (Weight, Systolic Pressure, Diastolic Pressure, Pulse, Oxygen, Daily weight difference, Daily systolic pressure difference, Daily diastolic pressure difference, Daily pulse difference, Daily oxygen difference, Average variation in weight for the past three days, Average variation in systolic pressure for the past three days, Average variation in diastolic pressure for the past three days, Average variation in pulse for the past three days, Average variation in oxygen for the past three days, Average variation in weight for the past seven days, Average variation in systolic pressure for the past seven days, Average variation in diastolic pressure for the past seven days, Average variation in oxygen for the past seven days, Time). The time variable is added as a periodic variable, and the dummy variables are added to calculate the response (class variable).

### *B. Carried approaches*

- Linear Regression:
  We propose to use the linear regression model to predict the heart failure risk level using the previously mentioned variables as input. Linear regression is the most basic type of regression and commonly used predictive analysis. The overall idea of linear regression is examining two things: does a set of predictor variables accurately predict an outcome variable? Which variables, in particular, are significant predictors of the dependent variable? [6] Linear regression uses the historical relationship (scatter plot) between an independent (x-axis) and a dependent variable (y-axis) to predict the future values of the dependent variables. The line with the smallest set of distances between the data points is the regression line. The trajectory of this line will best predict the future relationship between the two variables. In our case, the input is all the previously mentioned variables except for class variable, which is used as output. The following is the main equation of linear regression:

  $$h_\theta(x) = \theta_0 + \theta_1(x) + \theta_2 \qquad (1)$$

  $h_\theta$ (x) is the dependent value that the equation is trying to predict. $\theta_0$ and $\theta_1$ are selected so that the square of regression residuals is minimized. $\theta_2$ is the linear residual
- Time Series Linear Regression
  Time series is a sequence of numerical data points in successive order. Commonly, time series is a sequence taken at successive equally spaced points in time, also called discrete time data. Time series are often used to examine how the changes associated with the chosen data point are compared to the shifts in other variables over the same time period. In forecasting, time series uses information regarding historical values and associated patterns to predict future activity. [7]F The main characteristic of time series is seasonality, in which, data experiences regular and predictable changes that recur within every defined time interval, in other words, periodic fluctuations. Seasonality may be caused by different factors, one of these factors is repetitive and generally is considered as regular predictable patterns in the level of time series. [7] Sometimes cyclic patterns may occur, its when data exhibits rises and falls that are not among the fixed period. In general, the average length of cycles is longer than the

length of a seasonal pattern, and the magnitude of cycles tends to be more variable than the magnitude of seasonal patterns. [8]

- Ordinary Least Square (OLS)
  The OLS or linear least square is a method for estimating the unknown parameters in a linear regression model, with the goal of minimizing the sum of the squares of the differences between the observed responses in the given dataset and those predicted. To choose the most efficient OLS model, the following assumptions must be satisfied: [9]
  - All variables contained in a model are statistically significant. Meaning all their values must be greater than 0. Furthermore, no essential variable is omitted.
  - Residuals should not deviate significantly from 0 in any subset of the time series. Residuals represent the difference between the actual and the fitted value of each observation in the data series.
  - Residuals have a constant variance throughout the series. If the variance is not constant, then a number of remedies can be used, such as weighted regression power transformation, or generalized autoregressive conditional heteroscedasticity techniques may be applied to the data. Weighted regression is a technique that divides all series by the standard deviation of the errors term. A power transformation searches for the exponent of the series between -1 and +1 that results in a constant variance.
  - Residuals are free from autocorrelation for all lags.
  - Residuals are normally independently distributed. Failure of this assumption is linked to the failure of the previously mentioned condition.
  - Residuals are not a function of the lagged values of each of the independent variables.
  - X values in a series are not a function of the lagged residuals. If the combination of X values is a function of the lagged residuals, then a one-way causal model is the wrong functional term.
  - Residuals distribution is invariant over time, that is, one subset of the series data should have the same covariance structure [10] as another subset.

$$Y = \beta_0 + \sum_{p}^{j=1} \beta_j X_j + \tag{2}$$

  Y is the dependant variable. $\beta_0$ is the intercept of the model. $X_j$ is is the $j^th$ explanatory variable of the model (j=1p. $\epsilon$ is the random error with variance $\sigma^2$.

- Classification Learner
  The classification learner helps us train models to classify data using supervised machine learning. These machine learning tasks are comprised by interactively exploring data, selecting features, specifying validation schemes, training models and asserting results. Few examples of this classification:
  - Decision Trees: Used to visually and explicitly represent decisions and decision making. It uses a tree-like model of decisions. Its a commonly used tool in data mining and machine learning for deriving a strategy to reach a particular goal. [11]
  - Support Vector Machines (SVM): Are a set of supervised learning methods, used for both classification and regression challenges. SVMs are the coordinates of the individual observation. SVM is a frontier which best segregates the two classes hyper-plane/line. [12]

### C. *Used Softwares*

- Gretl
  Gretl (GNU Regression, Econometrics and Time-Series Library) is an open-source cross-platform software package. It has a graphical user interface. Gretl offers features necessary for performing econometrics and time series analyses. [13]
- MATLAB
  Developed by MathWorks, MATLAB is a tool used to design and analyse systems or data using matrix-based MATLAB language to express computational mathematics.

## IV. Experiments and Results

### A. *Experimental Protocol*

Our sample is a dataset of 1589 observations from all 3 patients; the dataset was split into 70% for training and 30% for testing. The models creation and training were conducted using the linear regression time series algorithm. Start date of our dataset was entered and the end date was calculated automatically by Gretl.

### B. *Obtained Results*

We plot the selected variable (in our case we will plot the Class variable) using time series plot. This will display the variables progress through time.

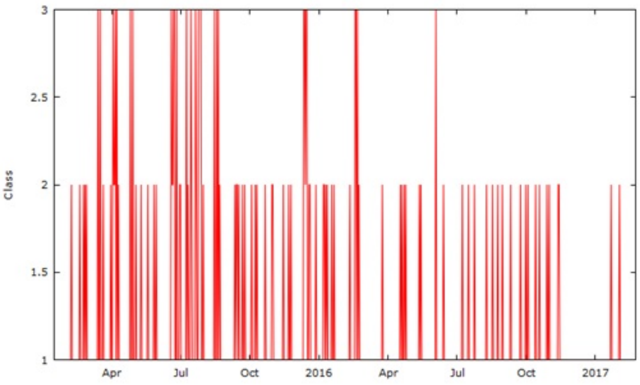


Fig. 2. Patient 1 Class Variable Graph.

The Class variable must be regressed against time by means of one-time variable and multiple dummy variables. Dummy

variable or indicator variable is an artificial variable created to represent an attribute with two or more distinct categories/levels. [14]
In order to compute the seasonality, one dummy variable is excluded. We obtained as results, R-squared = 0.708052 and Adjusted R-squared = 0.696440 = 69.6
Overall, we have an adjusted R-squared [15] of 69.6%. This is calculated based on independent variables.

$$R_Adjusted^2 = 1 - (1 - R^2)(N - 1)/(N - p - 1) \quad (3)$$

Where: $R^2$=Sample R Squared. p=Number Of Predictors. N=Total Sample Size.
An in-sample forecast is run by using a constrained sample from our dataset. The obtained results for an in-sample forecast are Mean Absolute Error = 0.15601, Mean Percentage Error = 2.8008 and Mean Absolute Percentage Error = 12.935.

- Mean Absolute Error, [16] measures the difference between values (sample and population values) predicted by a model or an estimator and the values actually observed.
- If we take the mean absolute percentage error of 12.935% we note that the actual value and predicted value are 12.935% apart. The mean percentage error is calculated using the following formula: [16]

$$(1/N)\sum_{k-1}^{N}|A_k - F_k|/A_k \quad (4)$$

$A_k$ is the actual value. $F_k$ is the forecasted value. N is the total sample size.
The error of 12.9% can be considered as a good result.

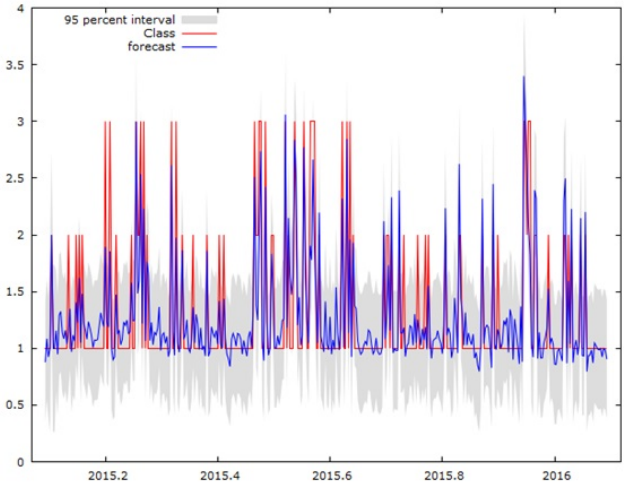


Fig. 3. Patient 1 Sample Prediction.

In this step, we will reset the sample to full range and rerun our linear regression model and add observations to the data, so we could predict the off-sample value of our targeted variable Class. After adding 10 observations to predict future unknown values, the above graph represents the results with 95% confidence interval. The blue line indicates the value of the classes each day; Mean Absolute Error = 0.24816, Mean Percentage Error = 5.4221 and Mean Absolute Percentage Error = 19.174. As the percentage error is 5.4221 this means, the accuracy is 94.5779%; with R-squared = 0.396388 and Adjusted R-squared = 0.365047 = 36.5%.

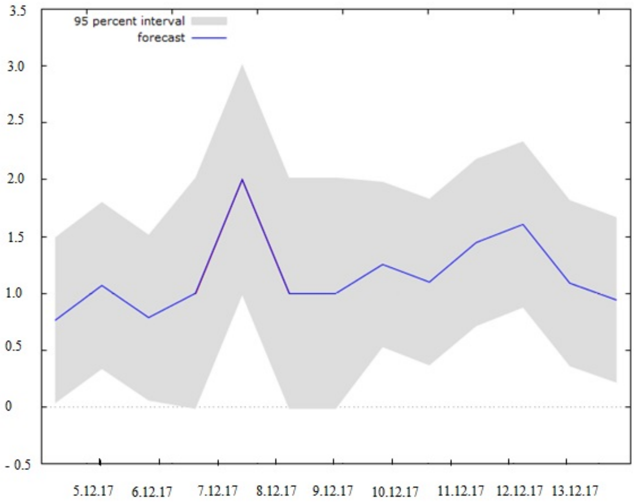


Fig. 4. 10 Day Off-Sample Prediction.

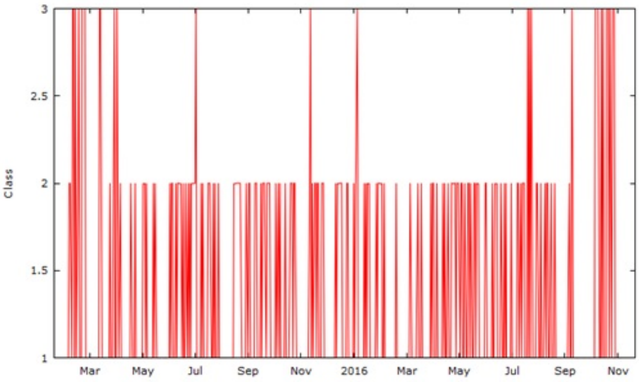


Fig. 5. Patient 2 Class Graph.

The in-sample forecast has the following results, Mean Absolute Error = 0.34503, Mean Percentage Error = -7.9653 and Mean Absolute Percentage Error = 25.823.

- If we take the Mean Percentage Error of 25.823% we note that the actual value and predicted value are 25.823% apart, which is considered unacceptable in clinical studies.

After adding 6 observations to predict future unknown values the above graph represents the results, with 95% confidence interval. The blue line indicates the value of the classes each day; showcasing a Mean Absolute Error = 0.2259, Mean

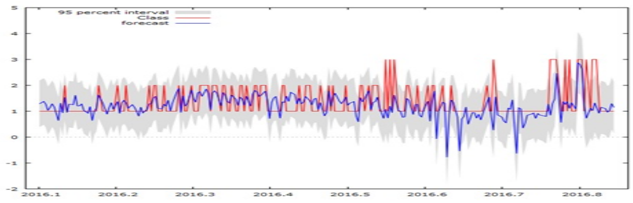


Fig. 6. Patient 2 Sample Prediction.

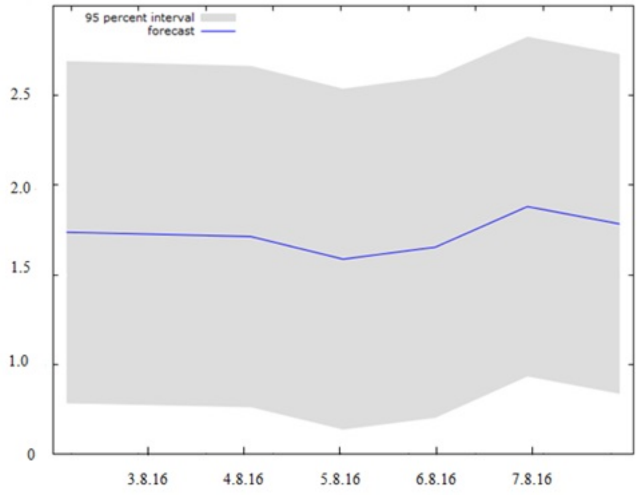


Fig. 7. Patient 2 Off-Sample Forecast Prediction.

Percentage Error = -22.59 and Mean Absolute Percentage Error = -22.59.
As the percentage error is 22.59 this means, the accuracy is 77.41%.

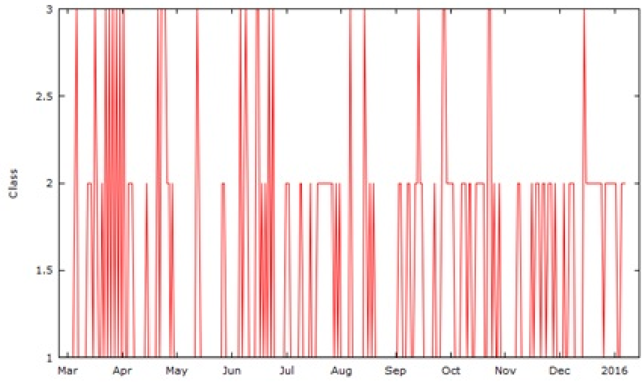


Fig. 8. Patient 3 Class Variable Graph.

The R-squared = 0.127255 and Adjusted R-squared = 0.039667 = 3.9667%, this in-sample forecast has the following results: Mean Absolute Error = 0.49991, Mean Percentage Error = -14.677 and Mean Absolute Percentage Error = 35.186.

- If we take the mean percentage error of 35.186% we note that the actual value and predicted value are 35.186% apart, which is considered unacceptable in clinical studies.

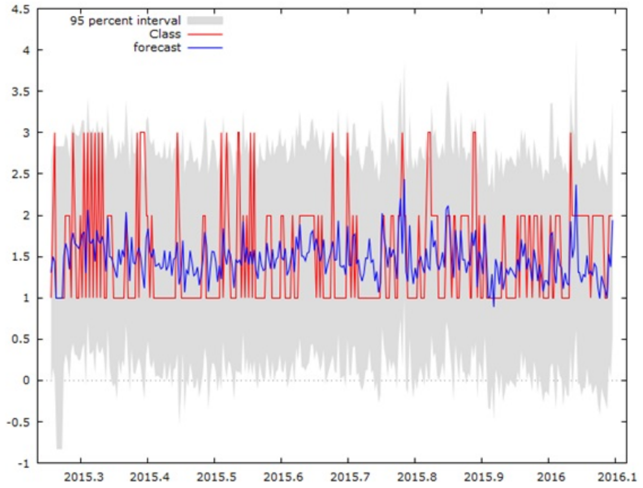


Fig. 9. Patient 3 In-Sample Prediction.

After resetting the sample to full range, and rerunning our linear regression model, we could now predict future observations with 95% confidence integral. We added 6 off-sample observations and the results are in fig. 10.

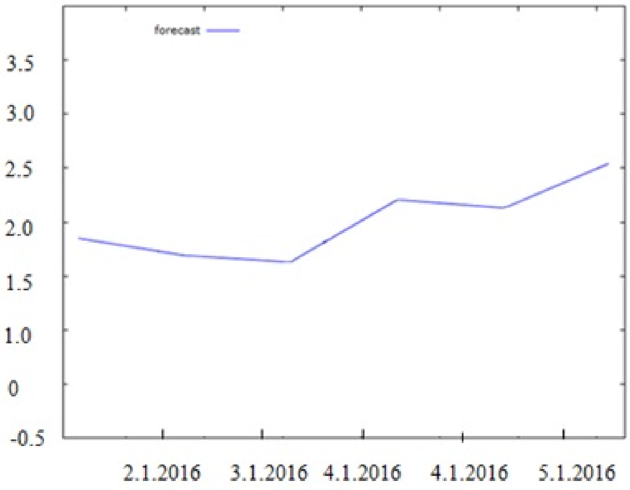


Fig. 10. Patient 3 Off-Sample Forecast.

For this patient the Mean Absolute Error = 0.49991, Mean Percentage Error = -14.677 and Mean Absolute Percentage Error = -22.59; as the Mean Percentage Error is 14.677 this means, the accuracy is 85.323%.

### *C. Results and Discussions*

- Gretl Results:
  The following table illustrates all the previously obtained results using Gretl.

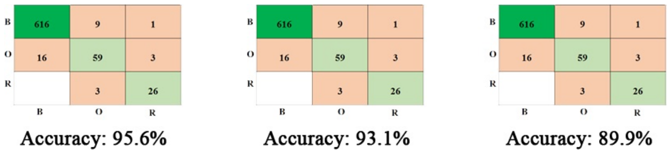
TABLE III
Patients Results Summary

| Patients | Accuracy |
|---|---|
| Patient 1 | 94.5779% |
| Patient 2 | 77.41% |
| Patient 3 | 94.8376% |

- MATLAB Results:
  On MATLAB, we used all available classification algorithms (in classification learner) to study the data, and generate confusion matrices. 5 folds were used for testing the model. The following table presents the most accurate algorithm for all 3 patients.
  The best classification algorithm is Complex Tree: The average accuracy is equal to 92.8667%

TABLE IV
MATLAB Results

| Algorithm | Accuracy |
|---|---|
| Medium and Complex tree | 95.6% |
| Medium and Complex tree | 93.1% |
| Medium and Complex tree | 89.9% |



| | B | O | R |
|---|---|---|---|
| B | 616 | 9 | 1 |
| O | 16 | 59 | 3 |
| R | | 3 | 26 |

Accuracy: 95.6%

| | B | O | R |
|---|---|---|---|
| B | 616 | 9 | 1 |
| O | 16 | 59 | 3 |
| R | | 3 | 26 |

Accuracy: 93.1%

| | B | O | R |
|---|---|---|---|
| B | 616 | 9 | 1 |
| O | 16 | 59 | 3 |
| R | | 3 | 26 |

Accuracy: 89.9%

Fig. 11. Generate Confusion Matrixes.

## V. Conclusion

This research highlighted the death risk of coronary heart disease and the main factors contributing to the patient's death, where we were able to implement a linear regression model capable of predicting which class can occur in the near future, based on multiple input samples. Also, when we look at this project, we find that it answers two main questions; the accuracy of the model and the prediction of a correct risk class, by having high efficiency and expandability; while including off-samples which are disregarded by other studies. Finally, recommendations for future studies should include larger forecast size and detecting what is/are the main factor(s) for risk evolvement. This will increase the relevance of the results and show even higher significance, thus we can cover more patients and reduce mortality rate.

## References

[1] B. G. Nordestgaard, M. J. Chapman, S. E. Humphries, H. N. Ginsberg, L. Masana, O. S. Descamps, O. Wiklund, R. A. Hegele, F. J. Raal, J. C. Defesche, A. Wiegman, R. D. Santos, G. F. Watts, K. G. Parhofer, G. K. Hovingh, P. T. Kovanen, C. Boileau, M. Averna, J. Born, E. Bruckert, A. L. Catapano, J. A. Kuivenhoven, P. Pajukanta, K. Ray, A. F. H. Stalenhoef, E. Stroes, M.-R. Taskinen, A. Tybjrg-Hansen, and for the European Atherosclerosis Society Consensus Panel, "Familial hypercholesterolaemia is underdiagnosed and undertreated in the general population: guidance for clinicians to prevent coronary heart disease consensus statement of the european atherosclerosis society," *European Heart Journal*, vol. 34, no. 45, pp. 3478–3490, 2013.

[2] C. J. McAloon, L. M. Boylan, T. Hamborg, N. Stallard, F. Osman, P. B. Lim, and S. A. Hayat, "The changing face of cardiovascular disease 2000–2012: An analysis of the world health organisation global health estimates data," *International journal of cardiology*, vol. 224, pp. 256–264, 2016.

[3] S.-L. Caroline, R. Lothar, F. Stephan, V. Mariuca, B. Martina, K. Ulrike, D. Stefnaie, and Z. Amdreas, "educed number of circulating endothelial progenitor cells predicts future cardiovascular events. proof of concept for the clinical importance of endogenous vascular repair.," vol. 16, pp. 2981–2987, 2005.

[4] A. Sharrett, C. Ballantyne, S. Coady, G. Heiss, P. Sorlie, and W. Patsch, "Coronary heart disease prediction from lipoprotein cholesterol levels, triglycerides, lipoprotein(a), apolipoproteins a-i and b, and hdl density subfractions.," vol. 10, pp. 1108–1113, 2001.

[5] X. Yanwei, W. Jie, and Z. Zhihong, "Combination data mining methods with new medical data to predicting outcome of coronary heart disease, convergence information.," pp. 868–872, 2007.

[6] R. B. Darlington and A. F. Hayes, *Regression analysis and linear models: Concepts, applications, and implementation*. Guilford Publications, 2016.

[7] J. J. Faraway, *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*, vol. 124. CRC press, 2016.

[8] C. Chatfield, *The analysis of time series: an introduction*. CRC press, 2016.

[9] J. C. Pickett, D. P. Reilly, and R. M. McIntyre, "How to select a most efficient ols model for a time series data," *THE JOURNAL OF BUSINESS*, vol. 11, 2005.

[10] R. Wolfinger, "Covariance structure selection in general mixed models," *Communications in statistics-Simulation and computation*, vol. 22, no. 4, pp. 1079–1106, 1993.

[11] https://www.mathworks.com/help/stats/classification-nearest-neighbors.html, n.d.

[12] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[13] R. Lucchetti *et al.*, "Who uses gretl? an analysis of the sourceforge download data," in *Econometrics with gretl. Proceedings of the gretl conference 2009, Bilbao, Spain*, pp. 45–55, 2009.

[14] S. Skrivanek, "The use of dummy variables in regression analysis," *More Steam, LLC*, 2009.

[15] J. Frost, "Multiple regression analysis: Use adjusted r-squared and predicted r-squared to include the correct number of variables," *Minitab Blog*, vol. 13, no. 06, 2013.

[16] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016.

# Techniques for Improving the Efficiency of Functional Programs

Francisco Ribeiro
*HASLab/INESC TEC*
*University of Minho*
Braga, Portugal
francisco.j.ribeiro@inesctec.pt

***Abstract*—Combining different programs or code fragments is a natural way to build larger programs. This allows programmers to better separate a complex problem into simple parts. Furthermore, by writing programs in a modular way, we increase code reusability.**

**However, these simple parts need to be connected somehow. These connections are done via intermediate structures that communicate results between the different components.**

**This review paper compiles different techniques used to remove intermediate structures and multiple traversals from programs written in functional languages.**

***Index Terms*—functional programming, deforestation, program fusion, circular programs, lazy evaluation**

## I. Introduction

Over the last several years, programming languages have evolved in order to provide powerful abstractions to programmers. Examples of such abstractions are models that represent code abstractions, powerful type systems and recursion patterns allowing the definition of functions that abstract the data type they traverse.

Examples of such recursion patterns are higher-order functions like *map* and *filter*. Composing operations like these ones makes it possible to express long, and sometimes complex, sequences of instructions with little effort.

However, if these mechanisms are not tuned appropriately, they may lead to efficiency problems, either by doing more traversals than necessary or by creating intermediate data structures. For example, the following concise Haskell function `all`,

```
all p xs = and (map p xs)
```

checks if all elements of a list `xs` satisfy a given predicate `p`. As we can see, it is expressed as a composition of functions `and` (conjunction of a list of booleans) and `map`. The `and` function is a fold on lists and, as a result, `all` is the composition of two higher-order functions.

```
all p xs = foldl (&&) True (map p xs)
```

In this definition, an intermediate list is created to communicate the results from one function to another. However, this program can be rewritten in a way which does not make use of an intermediate list.

```
all' p xs = h True xs
    where
        h b [] = b
        h b (x:xs) = h (b && p x) xs
```

Modifying the program's code in order to overcome these issues has the drawback of compromising readability and conciseness.

Furthermore, one does not wish to write programs in this style and, instead, prefers to use a more compositional style such as the first version of `all` provided that there are no performance penalties.

In addition, a more efficient implementation may not necessarily be the most natural solution to a problem, leading to increased difficulty during development and maintenance [1], [2].

In essence, programmers wish to write programs in the style they are most familiar with, not necessarily the most efficient one, and have them perform the best way possible. They want the best of both worlds.

Therefore, there's a need for techniques that automatically perform these kinds of optimizations automatically.

The remainder of this paper is structured in the following way: Sections II and III briefly describe lazy evaluation and deforestation, respectively. These two concepts are important to understand the basis of the techniques that will be explained subsequently. Sections IV, V, VI and VII describe specific fusion techniques used to optimize programs written in a functional style. In fact, these techniques are well known to the functional programming community and constitute key ideas behind important optimizations included in many compilers. Section VIII concludes the paper.

## II. Lazy Evaluation

As stated previously, intermediate lists connect the different parts that assemble a program. Therefore, with strict evaluation, a lot of intermediate structures are allocated along the way which do not take part in the final result [3], [4]. The problem with the memory usage of these structures can be overcome with lazy evaluation. This way, because elements are generated as they are needed, there is no requirement for loading the entirety of the intermediate lists.

However, even under this mechanism, each list element still has to be allocated, checked and de-allocated [3].

Therefore, lazy evaluation in itself is not enough to overcome all the disadvantages introduced by the use of intermediate lists.

As such, in order to address this issue, several techniques have been developed throughout the years with the aim of completely eliminating the creation of these intermediate structures.

## III. Deforestation

Deforestation is a technique allowing for the elimination of intermediate structures which are created and consumed soon afterward. Although the term may be used as a synonym to ”fusion” in general, it is generally used in order to refer to Philip Wadler's pioneer work [3], in which the author coins the term.

One of the first deforestation algorithms was presented by Philip Wadler [3] and, although it removed intermediate data structures, it had some disadvantages, as Gill et al. [2] state.

The major drawback of this kind of approach to the elimination of intermediate structures is the restriction imposed on the algorithm inputs. In his paper, Wadler presents what he calls a *treeless* form for defining functions which do not use any internal intermediate structures. The algorithm developed transforms a program composed by functions defined in *treeless* form into a single function, also defined in *treeless* form. As one can see, this is where one of the main disadvantages of this technique is evident. By limiting its application to functions defined in a restrictive form, the algorithm has a restricted range of inputs to operate on.

This places boundaries on the style of programming allowed to programmers, compromising code readability and conciseness.

A technique allowing the elimination of intermediate data structures, and thus creating a more efficient version, without sacrificing code clarity was needed.

## IV. Short-cut Fusion

Gill et al. [2] present a transformation technique to create more efficient versions of programs through the elimination of intermediate lists. The core idea behind this deforestation technique are the algebraic transformations performed on some functions.

With these algebraic transformations, the authors show it is possible to standardize the way lists are consumed and produced. Furthermore, this algorithm allows every program as input.

In Haskell, one could define the well known list data type as:

```
data List a = Nil | Cons a (List a)
```

`foldr` is a function which behaviour consists of processing a list with an operator and returning the value it constructed along the way (accumulated in an initial value).

This systematic consumption of a list can be thought of as replacing every occurrence of `Cons` with the provided operator and the `Nil` instance with the initial value.

Therefore, many functions that consume lists in a constant way like the one just described can be expressed in terms of `foldr`. That is because this higher-order function encloses that kind of systematic consumption of a list.

Some example implementations of pre-defined functions resorting to `foldr` are:

```
map f xs = foldr (\a b -> f a : b) [] xs

filter f xs = foldr (\a b -> if f a
                             then a:b
                             else b)
                    [] xs
```

However, this standardization of list consumption is not enough to achieve the desired program transformation, as the following example demonstrates.

Supposing a composition of functions like:

```
sum (map f ls)
```

where `map` applies function `f` to each element of `ls` and `sum` performs the addition of every element in the list.

One could modify this program and have:

```
foldr (+) 0 (foldr ((:).f) [] ls)
```

where `foldr` is a higher-order function which consumes a recursive data structure (in this case, a list) by applying a given combining function in a systematic way to all the constituent parts, building a return value in the end.

But there isn't a rule that simplifies occurrences of *foldr/-foldr*. A workaround for this, could be rewriting these kinds of programs in a more specific way, and have the above example transformed in:

```
foldr ((+).f) 0 ls
```

The problem with this approach is that it is not very general. More precisely, it is very difficult to be sure we have sufficient rules. When another combination of functions is encountered, a new rule would need to be written so that that particular case would get simplified.

In the example used to illustrate this, the `foldr` on the outside had no way to know how the `foldr` on the inside was producing its result list. As such, we also need a way to standardize list production.

The abstraction described for list consumption consists in the replacement of every `cons` with a function and the `nil` at the end with a given value. And `foldr` encapsulates this behaviour by receiving a function `f` and an initial value `acc`.

Therefore, if list production is abstracted in terms of `cons` and `nil`, it is possible to obtain `foldr`'s effect if this list-producing abstraction is applied to `f` and `acc`.

As such, a function `build` can be defined like:

```
build g = g (:) []
```

Following the line of thought just described, we come up with the *foldr/build* rule, which can be expressed as:

```
foldr f acc (build g) = g f acc
```

As an example, one can consider the `upto` function which, given two numbers, produces a list that starts from the first one and continues until the second one.

In a very straightforward way, one could define this function as:

```
upto x y = if x>y then []
           else x : upto (x+1) y
```

But, as stated before, we can try to abstract the production of the list in terms of `cons` and `nil`, and thus getting the following definition:

```
upto' x y =
    \cons nil →
            if x>y then nil
            else cons x (upto' (x+1) y
                                 cons nil)
```

Now, the function `upto` would be written like:

```
upto x y = build (upto' x y)
```

Deforestation is now possible if the list is produced using `build` and consumed using `foldr`:

```
mul (upto x y)
    = foldr (*) 1 (build(upto' x y))
    = upto' x y (*) 1
```

Applying the *foldr/build* rule (key elements highlighted inside red rectangles[1]) allows us to obtain a reduced form of the function `mul`, where no intermediate list is produced, which confirms the effect of deforestation.

## V. Circular Programs

Algorithms that perform multiple traversals on the same data structure can be expressed as a single traversal function through a technique called *Circular Program Calculation*.

This kind of approach, first explored by Bird [5], highlights the importance of the lazy evaluation mechanism in functional languages like Haskell. In fact, defining circular programs in this way only works because of lazy evaluation. A circular definition has the consequence of creating a function call containing an argument that is, simultaneously, a result of that same call. Under a strict evaluation mechanism, this can be a problem as an infinite cycle is created because values are demanded before they can be calculated, leading to non-termination.

On the other hand, lazy evaluation allows for the computation of such circular structures. With this strategy, the right evaluation order of the expression is determined at runtime. More specifically, only the elements of the expression to be computed that are necessary to continue are expanded.

Although circular programs avoid unnecessary multiple traversals, they are not necessarily more efficient than their more straightforward counterparts [6] and are even more difficult to write. In fact, even more experienced programmers find it hard to understand programs written in such a way. In his paper, Bird proposes deriving these circular programs from their less efficient (in terms of number of traversals), but more natural, equivalent solutions.

The example he uses is the function `repmin`, which has become a traditional example for being simple and a good assistant for the explanation of this particular technique.

The problem at hand is going to be the replacement of every leaf value in a tree with the original minimum value of the tree.

First of all, we must define a datatype for the tree. After that, we need a function `replace` and a function `tmin` to swap the tree's leaves for a given value and to calculate the minimum value of a tree, respectively.

With that, we can easily come up with a natural way of expressing the problem, which is implemented by the function `transform`.

```
data LeafTree = Leaf Int
              | Fork (LeafTree, LeafTree)

tmin :: LeafTree → Int
tmin (Leaf n) = n
tmin (Fork (l, r)) = min (tmin l) (tmin r)

replace :: (LeafTree, Int) → LeafTree
replace (Leaf _, m) = Leaf m
replace (Fork (l, r), m)
    = Fork (replace (l, m),
            replace (r, m))

transform :: LeafTree → LeafTree
transform t = replace (t, tmin t)
```

After having a straightforward solution to the problem, one can start applying Bird's proposed technique.

The first step consists of tupling. The functions `tmin` and `replace` both have a similar recursive pattern and operate on the same data structure. Therefore, a function `repmin` can be created by combining the results from the two previous functions in a tuple.

```
repmin (t, m) = (replace (t, m), tmin t)
```

Furthermore, a recursive definition of this function can be created, in which two cases need to be considered:

```
repmin (Leaf n, m)
    = (replace (Leaf n, m), tmin (Leaf n))
    = (Leaf m, n)

repmin (Fork (l, r), m)
    = (replace (Fork (l, r), m),
       tmin (Fork (l, r)))
    = (Fork (replace (l, m), replace (r, m)),
       min (tmin l) (tmin r))
    = (Fork (l', r'), min n1 n2)
```

[1] Colours assumed to be available

```
         where (l', n1) = repmin (l, m)
               (r', n2) = repmin (r, m)
```

The final step is where circular programming is used in order to put together the two elements forming the result of `repmin`.

Highlighted inside blue rectangles is the presence of circularity; `m` is being used simultaneously as an argument and a result of the same call.

```
transform :: LeafTree → LeafTree
transform t = nt
    where (nt,m) = repmin (t,m)
```

However, this method for deriving circular programs presents a drawback.

Although it allows the derivation of a circular program from a more natural and intuitive equivalent, removing the burden of having to come up with such a complicated implementation and creating a circular alternative which makes less traversals on the data structure, this technique does not guarantee termination. Fernandes et al. developed a different technique based on short-cut fusion for deriving circular programs [7]. Circular programs have also been the subject of study in other research works [8]–[11].

## VI. Stream Fusion

The work by Coutts et al. [1] in *Stream Fusion* consists of an automatic deforestation system that takes a different approach compared to more traditional short-cut fusion systems.

The approach taken by [2] with the *foldr/build* rule is to fuse functions that work directly over the original structure of the data, that is, lists.

In *Stream Fusion*, the operations over the original list structure are transformed in order to, instead, work over the co-structure of the list.

As Coutts et al. [1] state, the natural operation over a list is a *fold*, while on the other hand, the natural operation over a stream is an *unfold*. Therefore, a list's co-structure is a stream.

The Stream datatype encloses that unfolding behaviour. In order to achieve this, it wraps an initial state and a stepper function which specifies how elements are produced from the stream's state.

```
data Stream a
    = ∃s. Stream (s → Step a s) s
```

The stepper function produces a `Step` element, which permits three possibilities:

```
data Step a s = Done
              | Yield a s
              | Skip s
```

The `Step` datatype allows the co-structure to be non-recursive, thanks to the `Skip` data constructor. This is the key point of the stream fusion system. The `Skip` constructor is what allows the production of a new state without yielding a particular element and this is a crucial point as it permits every stepper function to be non-recursive.

The `Done` and `Yield` alternatives are quite simple as they pinpoint the end of a stream and carry an actual element together with a reference to the rest of the stream's state, respectively.

In order to convert list structures to streams and vice-versa, two functions are needed.

```
stream :: [a] → Stream a
stream xs0 = Stream next xs0
        where
            next [] = Done
            next (x:xs) = Yield x xs

unstream :: Stream a → [a]
unstream (Stream next0 s0) = unfold s0
    where
        unfold s = case next0 s of
            Done → []
            Skip s' → unfold s'
            Yield x s' → x : unfold s'
```

The function `stream` creates a Stream with:

- a stepper function `next` which is non-recursive and yields each element of the stream as it unfolds;
- a state, which consists of the list itself.

On the other hand, the function `unstream` creates a list by unfolding the given stream, repeatedly calling the stream's stepper function.

Implementing functions to perform operations over streams is quite simple. The function intended has to define the particular stepper function for the stream it is going to return as a result. Considering the simple and well known `map` example operating on lists, one would define its stream counterpart as:

```
map_s :: (a → b) → Stream a → Stream b
map_s f (Stream next0 s0) = Stream next s0
    where
        next s = case next0 s of
            Done → Done
            Skip s' → Skip s'
            Yield x s' → Yield (f x) s'
```

What $\texttt{map}_s$ does here is define a stepper function that applies the function given as a parameter of $\texttt{map}_s$ to every yielded element of the stream.

A very simple but important case where one can see the effect of the stream fusion approach is the function $\texttt{filter}_s$. Its implementation allows us to observe the true impact of this technique.

```
filter_s :: (a → Bool) → Stream a → Stream a
filter_s p (Stream next0 s0) = Stream next s0
    where
        next s = case next0 s of
            Done → Done
            Skip s' → Skip s'
            Yield x s' | p x → Yield x s'
                       | otherwise → Skip s'
```

The only way that the function $\texttt{filter}_s$ is non-recursive is because of `Skip`. This constructor, when put in place of the elements that should be removed from the stream, allows us to avoid the recursion otherwise necessary to process every stream element in order to find out which ones satisfy the given predicate.

More precisely, in the last line of the above implementation, `Skip` is introduced whenever an element does not pass the predicate's test.

This way, code can be better optimized by general purpose compiler optimizations.

In order to use the stream fusion approach on lists, one has to convert lists to streams and back again. This way, functions from the Stream setting (like the previous $\texttt{map}_s$ and $\texttt{filter}_s$ examples) can be applied, as they are intended to operate on streams. This is accomplished by using functions `stream` and `unstream`. As an example, function `map` on lists is specified in the following way:

```
map :: (a → b) → [a] → [b]
map f = unstream . maps f . stream
```

This way, each function has to construct a Stream, perform its task and then build a list. Doing this for every function in a stream pipeline would be very inefficient. Considering the example of composing a $\texttt{filter}_s$ and a $\texttt{map}_s$, the following is obtained:

```
filter p . map g =
     unstream . filters p . stream .
     unstream . maps g . stream
```

Communicating the results from $\texttt{map}_s$ to $\texttt{filter}_s$ builds an intermediate list (generated by `unstream`), which gets consumed right away (`stream`). But there is a chance to eliminate this intermediate list.

`stream . unstream` is the identity on streams and, as a result, it can be removed. Formalizing, this originates the *stream/unstream fusion* rule:

```
∀s . stream (unstream s) ↦ s
```

The Glasgow Haskell Compiler allows us to write rules that will then be used while compiling our programs. These "custom rules" can be expressed through pragmas, which are instructions that can be given to the compiler. Expressing the previous rule through these pragmas will make the example be transformed into:

```
unstream . filters p . maps g . stream
```

The *stream/unstream fusion* rule is not a traditional fusion rule, as it merely eliminates lists that got created while converting operations.

In fact, the method documented so far has a very curious and important implication. As previously stated, one can define pragmas in order to extend the compiler. Some algebraic transformations can be expressed through these pragmas. For example:

```
map f (map g xs) = map (f.g) xs
```

This algebraic rule expresses that a composition of `maps` is equivalent to the `map` of the composition of the two functions. This rule allows for the generated code to be more efficient.

However, there is a multitude of possible function combinations and, as a consequence, one could never be certain of the number of rules necessary to cover all cases.

This is a point where the work by Coutts et al. [1] plays an important role. As presented earlier, when writing different stream combinators (like $\texttt{map}_s$ and $\texttt{filter}_s$), the outcome of each stepper function that is defined depends on the outcome of the previous stream's stepper function.

This way, whenever a stepper function of a stream is called, every stepper function of the streams preceding the current one is going to be executed.

Therefore, functions are fused without the need to explicitly state the rules performing those transformations.

The main goal of program fusion is to eliminate intermediate data structures. However, *Stream Fusion* achieves that at the cost of introducing lots of intermediate `Step` values. These allocations are going to be responsible for a great amount of overhead. This situation is overcome thanks to several optimization techniques included in *GHC* (e.g. case-of-case transformation and constructor specialisation). Therefore, programs are automatically transformed and, in the end, the most efficient solution is obtained (where all the intermediate values mentioned have been eliminated, thus reducing unnecessary allocations).

## VII. Hylo System

Program calculation is what is behind the techniques described to transform programs into more efficient versions. These techniques are based on many existing transformation laws. However, these rules only allow us to work with programs by hand, therefore leaving the application of program transformations necessary to obtain more efficient versions to the programmer, and not to the computer. Fusion systems are, as a consequence, not automatic.

Algorithms need to be developed that construct programs based on those transformation laws. This is what the HYLO System by Onoue et al. [12] aims to be: a fusion system applying these transformations in a more universal and regular way than existing ones.

First of all, we need to understand that there are two possible approaches to fusion: search-based fusion and calculational fusion.

The first one, search-based fusion, unfolds recursive definitions of functions to find suitable places inside those expressions to perform folding operations. But to achieve this, this kind of method needs to keep track of the function calls so it can control the unfolding, in order to avoid an infinite process. As this introduces a great overhead, fusion cannot be practically implemented this way.

The work by Onoue et al. [12] focuses on the second kind of fusion, calculational fusion, which has been the object of a lot of investigation over the years.

This approach explores the recursive structure of each component of the program in order to apply fusion through existing transformation laws.

However, most of the proposed techniques for fusion have the slight inconvenient of forcing the programmer to express the functions in terms of the necessary recursive structure, so that the different transformations can be applied. This is impractical, as it leads the programmer away from more potentially readable and natural implementations.

With this in mind, when briefly explaining their approach, the authors of the HYLO System start by stating that the majority of recursive functions can be expressed in terms of a very specific recursive form: hylomorphism.

An hylomorphism is the composition of an anamorphism (list production) followed by a catamorphism (list consumption).

Consider the following Haskell implementation of the *Fibonacci* function:

```
fib 0 = 0
fib 1 = 1
fib n = fib (n-1) + fib (n-2)
```

The sequence of calls generated by this program could be generalized over a binary tree, which would then collapse in order to calculate the desired $n^{th}$ *Fibonacci* term.

This, in essence, is a hylomorphism, in which the anamorphism corresponds to the generation of the call tree and the catamorphism to its collapse.

Indeed, this program could be rewritten in terms of an `unfold` (anamorphism) followed by a `fold` (catamorphism).

```
fib'_T :: Integer → Integer
fib'_T = fold_T (+) id ∘ unfold_T g
    where
        g 0 = Left 0
        g 1 = Left 1
        g n = Right (n-1, n-2)
```

By expressing the program this way, if the two recursive patterns that compose the hylomorphism get fused, the creation of the intermediate structure (call tree) is avoided.

In order to rewrite the program's recursive components in terms of hylomorphisms, the authors of the HYLO System developed an algorithm to derive such general recursive structures from the recursive definitions of the program.

As such, for the previous example, the algorithm would derive the `fold ∘ unfold` definition from the original `fib` implementation.

Following that, schemes for data production and consumption need to be captured so that the *Acid Rain Theorem* can be applied to hylomorphisms, in order to fuse them.

The final step consists of inlining the resulting hylomorphism into a normal recursive definition, in which the intermediate structures have been eliminated.

All in all, the HYLO System allows programs to be written without the concern of expressing them in terms of specific and more generic recursive structures, as these are derived by an automatic algorithm. Thus, fusion laws can still be applied, leading to more efficient programs without sacrificing so much code readability and without forcing programmers to express functions under certain recursive patterns. This system was incorporated into the Haskell compiler.

## VIII. Conclusion

Throughout the years, programming languages have come up with new mechanisms that allow programmers to abstract more complex ideas into simple instructions. However, these abstractions may lead to performance issues. Chaining several *higher order functions* can cause a program to perform extra unnecessary traversals and operations if optimization techniques like *fusion* are not implemented.

Techniques like *deforestation* and *short-cut fusion* aimed to eliminate intermediate structures that were inevitably created as a way to ”glue” different functions together. Other approaches, like *circular program calculation*, focused on converting algorithms which performed multiple traversals to programs performing a single one.

Ultimately, *Stream Fusion* accomplishes both. By rewriting Haskell's List library functions in order to adapt them to the *Stream* setting and, together with that, integrating with an existing set of compiler optimization rules, this approach accomplishes some kind of automation when it comes to fusion. Automating this final step is something that previous techniques have missing. In a similar way, the *HYLO System* also tries to perform these transformations automatically by extracting the recursive structure of programs and performing fusion through the application of transformation laws.

In recent years, many languages such as C++, C# and Java started adopting functional constructs as a form of enriching the way they allow people to write programs. As some of these constructs include many of the *higher-order functions* presented previously, the techniques discussed in this paper are of utter importance as the introduction of this ”functional flavour” demands the inner workings of these languages to cope with the introduced overhead by somehow mimicking the formerly described optimizations.

Other areas, that are somehow related to functional programming, also benefit from the kind of mechanisms described throughout this paper. Attribute grammars is an example of such an area in which fusion techniques play a very important role [13].

## References

[1] D. Coutts, R. Leshchinskiy, and D. Stewart, “Stream fusion: From lists to streams to nothing at all,” in *Proceedings of the 12th ACM SIGPLAN International Conference on Functional Programming*, ser. ICFP '07. New York, NY, USA: ACM, 2007, pp. 315–326. [Online]. Available: http://doi.acm.org/10.1145/1291151.1291199

[2] A. Gill, J. Launchbury, and S. L. Peyton Jones, “A short cut to deforestation,” in *Proceedings of the Conference on Functional Programming Languages and Computer Architecture*, ser. FPCA '93. New York, NY, USA: ACM, 1993, pp. 223–232. [Online]. Available: http://doi.acm.org/10.1145/165180.165214

[3] P. Wadler, “Deforestation: transforming programs to eliminate trees,” *Theoretical Computer Science*, vol. 73, no. 2, pp. 231 – 248, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/030439759090147A

[4] ——, “Listlessness is better than laziness: Lazy evaluation and garbage collection at compile-time,” in *Proceedings of the 1984 ACM Symposium on LISP and Functional Programming*, ser. LFP ’84. New York, NY, USA: ACM, 1984, pp. 45–52. [Online]. Available: http://doi.acm.org/10.1145/800055.802020

[5] R. S. Bird, “Using circular programs to eliminate multiple traversals of data,” *Acta Informatica*, vol. 21, no. 3, pp. 239–250, Oct 1984. [Online]. Available: https://doi.org/10.1007/BF00264249

[6] J. P. Fernandes, J. Saraiva, D. Seidel, and J. Voigtländer, “Strictification of circular programs,” in *Proceedings of the 20th ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*, ser. PEPM ’11. New York, NY, USA: ACM, 2011, pp. 131–140. [Online]. Available: http://doi.acm.org/10.1145/1929501.1929526

[7] J. P. Fernandes, A. Pardo, and J. Saraiva, “A shortcut fusion rule for circular program calculation,” in *Proceedings of the ACM SIGPLAN Workshop on Haskell Workshop*, ser. Haskell ’07. New York, NY, USA: ACM, 2007, pp. 95–106. [Online]. Available: http://doi.acm.org/10.1145/1291201.1291216

[8] P. Martins, J. P. Fernandes, and J. Saraiva, *Zipper-Based Modular and Deforested Computations*. Cham: Springer International Publishing, 2015, pp. 407–427. [Online]. Available: https://doi.org/10.1007/978-3-319-15940-9_10

[9] A. Pardo, J. P. Fernandes, and J. Saraiva, “Shortcut fusion rules for the derivation of circular and higher-order monadic programs,” in *Proceedings of the 2009 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-based Program Manipulation, PEPM 2009, Savannah, GA, USA, January 19-20, 2009*, G. Puebla and G. Vidal, Eds. ACM, 2009, pp. 81–90. [Online]. Available: http://doi.acm.org/10.1145/1480945.1480958

[10] ——, “Shortcut fusion rules for the derivation of circular and higher-order programs,” *Higher-Order and Symbolic Computation*, vol. 24, no. 1, pp. 115–149, Jun 2011. [Online]. Available: https://doi.org/10.1007/s10990-011-9076-x

[11] ——, “Multiple intermediate structure deforestation by shortcut fusion,” *Science of Computer Programming*, vol. 132, pp. 77 – 95, 2016, selected and extended papers from SBLP 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167642316300880

[12] Y. Onoue, Z. Hu, M. Takeichi, and H. Iwasaki, “A calculational fusion system hylo,” in *Proceedings of the IFIP TC 2 WG 2.1 International Workshop on Algorithmic Languages and Calculi*. London, UK, UK: Chapman & Hall, Ltd., 1997, pp. 76–106. [Online]. Available: http://dl.acm.org/citation.cfm?id=265779.265797

[13] J. Saraiva and D. Swierstra, “Data Structure Free Compilation,” in *8th International Conference on Compiler Construction, CC/ETAPS’99*, ser. LNCS, Stefan Jähnichen, Ed., vol. 1575. Springer-Verlag, March 1999, pp. 1–16.

# GreenSource: Repository tailored for Green Software Analysis

Rui Rua
Departamento de Informática
Universidade of Minho
R. da Universidade, 4710-057 Braga
rui.a.rua@inesctec.pt

*Abstract*— **Energy consumption analysis and energy-aware development have won the attention of both developers and researchers over the past years. The interest is becoming more notorious due to the proliferation of mobile devices, where saving energy is a key concern.**

**In the last years, a considerable number of studies aiming at analyzing the energy consumption emerged, with objectives such as measuring/estimating the energy consumed by an application or code block, or even detecting energy-expensive coding patterns. However, when it comes to actually improving the energy efficiency of an application, the amount of information provided about source code energy consumption that can be used by developers to reduce it in the development phase, is still very low.**

**In this paper we present GreenSource, a publicly available repository containing more than 600 Android Projects extracted from open-source repositories. This infrastructure contains static and dynamic metrics obtained from the execution of stress and unit tests over the projects' applications. The results were obtained using a tool developed in this work context, the AnaDroid framework. This tool uses testing frameworks and an energy profiler to instrument, build and execute tests over applications in a physical device, while monitoring its energy and resources consumption/usage.**

**Processing each one of this projects is a time-consuming task, due to the lack of tools capable of gather all this information and the large size and complexity of the projects. With this work, we intend to openly provide the resultant metrics and metadata obtained from the process of analyzing the projects and its execution. The queryable and minable data provided by the GreenSource can be used for further studies and researches, helping developers community to reason about energy consumption in software and relate it to source code.**

## I. Introduction

With the advancement of the technological age, the software engineering community has been focusing on how software is developed and continually progressed in this direction. Efforts in this regard have been made at various levels, from hardware [1] level to compilers [2], programming languages [3] or integrated development environments (IDE's). These intended to increase the productivity of software development, abstracting inherent development processes, allowing developers to focus on the most essential functional aspects of their software product.

However, since the beginning of the century we have witnessed a revolution in the computer systems portability. The portability factor became much valued by users, and thereafter also for its manufacturers. Consequently, given the limited capacity of the battery of such devices, the optimization of energy consumption for these has proved to be a crucial aspect for producers of these, as well as for the developers of software for these platforms.

Accompanying the mobile market growing, the Android ecosystem keeps evolving at an impressive pace as well. Since this operative system can run on a wide variety of devices, from smartphones, tablets or weareables, its widespread usage in the last decade was significantly notorious. This is the most used operative system for mobile devices, having in 2018 around 84,8% devices running its platform[4].

Over the last few years, the interest in analyzing energy consumption of the Android platform and respective applications has been increasing significantly. Energy-greedy mobile apps that drain the battery of devices are perceived as being of poor quality by users [5]. As a consequence, users are likely to uninstall an energy-inefficient app, and sometimes are even recommend to do so.

Due to this recent interest, in the last decade several works in this sense appeared. These aimed at analyzing the energy consumption in multiple ways, such as measuring/estimating the energy consumed by an application or block of code [6], [7], or even detecting energy expensive coding patterns [8] or API's [9]. In order to perform optimization in terms of energy at software level, we face a whole new challenge, which can only be achieved through source code improvements that can take advantage of energy saving techniques. Nonetheless, in order to identify energy-greedy code and propose techniques/solutions to avoid it, a significant and characterizing amount of information regarding the code energy consumption has to be analyzed.

The versatility and continuous evolution of the Android platform, with a constantly changing architectural and functional environment, leads to the increasing challenge of gather characterizing information about its energy and resources consumption. Since this platform runs in a countless number of devices [10], with different hardware components and architectures, running different versions of the system, its almost unfeasible to find solutions with satisfactory results for all configurations.

In this paper, we present an approach to gather useful metrics and data about the execution of portions of applications' source code in physical devices. We reused the GreenDroid [11] concept to instrument and monitor the execution of source code portions, providing an extensible framework that can be used by developers to estimate energy consumption of application. We executed this framework over more than 600 Android applications, extracted from the MUSE repository[1]. The results of the execution of the extracted applications was then centralized in an open repository. This infrastructure was designed to store metrics and metadata relatively to executed code of Android applications. The obtained information is related to the characteristic of the executed application and respective platform and device.

To summarize, the developed work involves essentially two main artifacts:

- The **Anadroid** framework: Tool that resulted from the evolution of GreenDroid, which consisted in one of the starting points to carry out the energy consumption analysis of source code in Android. This framework was conceived in order to have ability to instrument the source code of any Android project, generate the respective APK (Android PacKage) and monitor its execution. The execution of the applications is done through stress or unit tests.
- **GreenSource** infraestructure: To demonstrate the power of the AnaDroid framework, we executed it over hundreds of Android projects. Having access to a large number of applications and a powerful tool like the AnaDroid , it was decided to build an infrastructure capable of store and organize information of all executions. As such, the GreenSource was built. It is a repository containing data and metrics related to the structure and performance of applications that can be related to the energy consumption of its source code.

The information that the resultant infrastructure contains is openly available for consultation (`http://greenlab.di.uminho.pt/greensource/`). The main goal of this work is to offer a relevant scientific contribution that can be significant and characterizing the Android platform, being able to be reused in later studies. In order to achieve these objectives, we intend to continue to populate this repository with information regarding more applications, tests and devices. The contained data can be subjected to analyzes and studies (e.g. Data Science/Machine-Learning) that can allow to correlate factors that can have a significant impact on energy consumption and obtain relevant conclusions about of the applications code.

The remaining of this paper is organized as follows. In Section 2 we introduce the main artifacts and methodologies followed to obtain the GreenSource infrastructure and the information contained in it. We then present the results of the experiment in Section 3. Section 4 presents the threats to the validity of this work. Finally, in Section 5 we present our conclusions and future work directions.

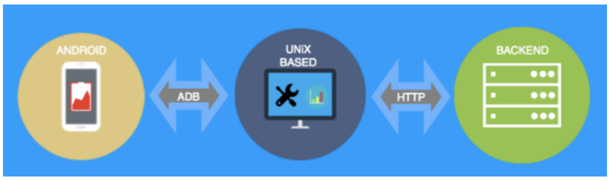


Fig. 1: GreenSource high-level components

## II. GreenSource

### A. Data provenance

In order to obtain diverse and relevant material for the accomplishment of studies that could reach significant magnitude, we needed to gather a significant number of Android Projects. The goal was to collect projects from a wide variety of sources in order to obtain a diverse set and whose projects/source code was openly accessible. Excepting the alternative of developing a tool that analyzes open-source repositories, which identifies Android projects and extracts that content, we reused previous works that had the same goal.

In order so obtain such corpus of Android projects, We took advantage of the work done during the development of GreenDroid[11], whose goal was to extract Android projects from the MUSE repository, an extension of the sourcerer repository[12]. These projects were collected from other open repositories and were developed in Java, the current leading development language for the Android platform.

Among the thousands of projects contained in the repository, we selected those that we could identify as Android projects, by executing queries on the repository database. Of all identified projects, we selected a subset that we identified as functional (i.e. compiled, built and executed without errors). Excluding all the problematic apps, we obtained a set containing more than 600 functional projects, which allow us to build applications that can be installed and run on Android devices. This set represents the starting point for the creation of the repository, already having a considerable size and minimally representative, and can be increased in future updates.

### B. AnaDroid Framework

The AnaDroid tool was developed to offer a generic way of integrating the ability to measure the energy consumption of an Android application. This tool can be used during its development process, as well as to automate the procedure of executing it over a large set of applications. This framework comes as an evolution of the GreenDroid framework [11], making it more accurate, current and complete. Its workflow is quite similar, from the instrumentation phase to the test execution phase. Several changes were made to how GreenDroid performed the instrumentation, exercised and analyzed the code and energy consumption of applications. Its concept has been extended to be able to interact with more testing frameworks, as well as new energy profilers, such as Trepn Profiler. With the inclusion of these new tools and with changes made to its workflow and how it analyzed the

[1] Muse repository: `https://opencatalog.darpa.mil/MUSE.html`

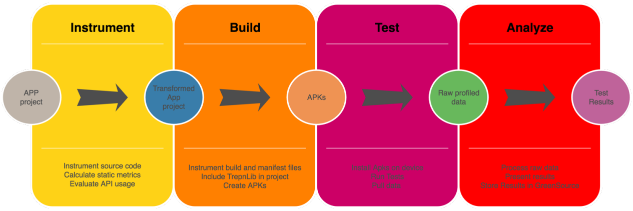


Fig. 2: AnaDroid Workflow

application code statically, it was possible to extract more information that can be associated and justify the energy performance of the applications.

The AnaDroid workflow is showed in the figure 2. It starts by instrumenting an Android project, both at the source code and building scripts level. This step is needed, in order to delimit the code execution interval and make calls to the energy profiler. In addition, during the instrumentation, it also collects static metrics and metadata about application methods and classes (Some of them are present in table **??**. Te next steps consists in generate the APK and install it on a physical device, using the ADB tool (Android Debug Bridge), allowing to perform and manage these tasks from the development machine. According to the intended testing framework, the tests are then executed on the device, the resultant data is collected and the results for each of the tests are generated. Before and after each test run, information about the resources (CPU, free memory and number of running processes) and status of the device which may interfere with the tests results are collected . In addition, in the final stage (Analyze phase), the AnaDroid can analyze and process the results obtained for each test executed. Then, it can send the obtained metrics and data to the GreenSource backend, in order to centralize results and contribute to the growth of knowledge regarding the power consumption and features of Android code.

### *C. TrepnLib and Trepn*

Trepn Profiler[2] is a software-based artifact developed by Qualcomm that works on devices with Snapdragon chipset-based Android devices. It is a diagnostic tool designed for expert consumers, such as Android developers. It can be used to profile hardware usage (like GPS, WiFi and others), resources usage (memory, CPU) and power consumption of the system or standalone Android applications. This tool doesn't need external (hardware) tools, as it gets its power readings from the Power Management IC (PMIC) and the battery fuel gauge software.

Trepn can be used as an standalone application, or as a service (an unix-like daemon in Android), which allows invocations via source code or from the ADB tool. This versatility makes this profiler easy to integrate in Android-based tools and applications, in purpose of measure and profile portions or entire applications. It provides the capability of pin data points (application states) while monitoring, which can be used to log and mark specific events during the profiling timeline.

Given the capability of Trepn of being invoked via (Java) source code, we had to find a way to easily integrate his calls in Android Applications, abstracting the calls to the Trepn Service. We developed a Android Library (TrepnLib) for this purpose, providing an API that allows to isolate and profile portions/code blocks (like methods,loops or Activity's lifecycle) of any Java class present in the application source code. Instrumenting the source code with the API provided by the TrepnLib, it is possible to estimate the power consumption and profile the isolated block, as well log other relevant events, like the start/end of methods, identify recursive calls, and others. To provide all this capabilities, we designed the TrepnLib taking into account his use cases. We reached the conclusion that the most common blocks/portions of code that are more relevant to isolate in terms of debugging and development process were methods and (unit) test cases. The capability of estimate power consumption of Java code at instruction/line level is not reliable using Trepn Profiler, since his sample rate is never lower than 100 ms, difficulting the task of associate samples at an specific rate with executions of instructions that take only a few milliseconds to run.

Furthermore, we provided functions to start and stop the profiling process, given the type of instrumentation (method or test oriented), that start and stop the Trepn Service, as well creates auxiliary files that are used to manage several runs,states and contexts. Methods to trace usage of methods and log states/events are also provided.

### *D. GreenSource Backend*

In order to give a greater purpose to the Anadroid framework, it has been integrated into the GreenSource repository. The main function of GreenSource's backend is to store and manage the information gathered through AnaDroid tool executions over Android projects. As such, the GreenSource contains a database within, having the function of providing an uniform way of communicating with it. In this way, mechanisms can be put in place that eases the processes of management, validation and manipulation of data at a higher architectural level, obtaining an abstraction level independent of the database engine used. The communication interface chosen consists in a RESTful API, which enables a uniform form of communication that provides the ability to consult, insert, change and delete data through HTTP requests. The database has been carefully designed to be expandable for future refinements and expansions of the AnaDroid tool. This database is a relational database and its schema consists of 21 tables, which refer to the elements that compose the application, as well as metadata and metrics related to the execution and analysis of the ones made on them.

The way the database was structured and developed, allows it to accompany the expansion of Anadroid, being easily extensible to support different test frameworks, devices and energy profilers.

[2] `https://play.google.com/store/apps/details?id=com.quicinc.trepn`

## III. Results

This section demonstrates some results obtained with the help of the AnaDroid tool, which were stored in the database of the GreenSource backend. Several types of results were selected for the execution of application tests with the UI/Application Exerciser Monkey test framework. These results allow to compare tests, applications and portions of these, as well its executions.

The process of running AnaDroid on a wide range of applications is an extremely costly process over time. This time is influenced by both the performance of the development machine and the Android device on which the tests are performed. Until the writing of this paper, this framework was successfully executed over a total of 352 Android projects. The features and specifications of the device in which the applications and tests were executed are described in table I.

| Feature | Details |
|---|---|
| Chipset | Snapdragon 400 Qualcomm MSM8226 |
| CPU | 1.2 GHz Quad Core |
| GPU | Adreno 305 |
| RAM | 1 GB |
| Mem | 8 GB |
| Screen | IPS LCD 720 x 1280 pixel 16M colors |
| Wifi | 802.11b/g/n |
| Bluetooth | 4.0 com A2DP/LE |
| GPS | A-GPS/GLONASS |
| Battery | 2070 mAh |

TABLE I: Android device specifications

We tried to run tests until we reached a relevant method coverage, approximately equal to or greater than 60%. These were done using the framework UI Application Exerciser Monkey, since its tests reach much higher values of method coverage than those obtained with the JUnit tests present in some projects. In order to reach this level of method coverage, 20 equal tests (generated from the same seeds) were carried out for each one of these applications. If this level of method coverage was not reached after 20 tests, the process would continue for more 30 tests. These tests were executed using the same seeds, in order to generate the same sequence of events for every application. The workflow of the test execution is represented in figure 3

In order to prevent the pseudo-random events generated by the Exerciser Monkey from turning on/off system resources or invoking other applications, some precautions had to be taken. The first consisted of using an auxiliary application called Simiasque[3], which hides status bar under an overlay mask, preventing monkey tests from clicking it. The second was to prevent Exerciser Monkey from generating system-events (pressing the Home, Back, Start Call, End Call, or Volume buttons) i to prevent generated events from being made outside the running application interface or prevent the phone from rebooting.

We then repeated this process for the 352 applications and analyzed the obtained results. By obtaining this type of results for a large set of applications, it is possible to compare applications for each one of the obtained metrics, in order to be able to correlate them with the (energy) performance of these. To illustrate examples of the comparisons and conclusions that can be made (in the future, with much more confidence, when a more significant number of different applications were analyzed), we have selected the following applications:

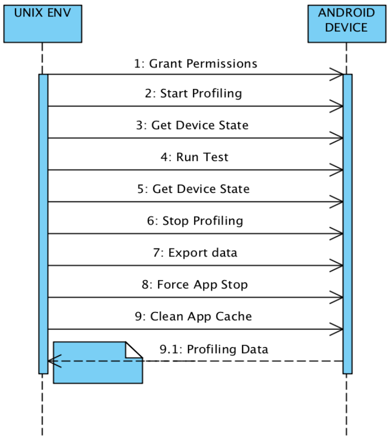


Fig. 3: Test Execution workflow

- Android DisplayingBitmaps: due to being the application with more methods invoked during the execution of the tests.
- PkTest[4]: Application that has achieved considerable test runtime and uses above average amount of sensors/hardware usage.
- Material Library: It obtained a total execution time of tests quite similar to the PkTest application.

The nature of the results allow to visualize and compare applications according to the executed tests. For the Android DisplayingBitmaps, the table II shows some of the results obtained for each executed test.

| Test Number | Consumption (J) | Time (ms) | Coverage (%) | Avg Mem Usage (MB) | Avg GPU Load (%) | Avg CPU Load (%) |
|---|---|---|---|---|---|---|
| 89160419 | 74.919 | 33515 | 69.69 | 836987.535 | 0.88 | 61.789 |
| 11 | 48.04278 | 21551 | 40.40 | 843599.578 | 5.912 | 53.960 |
| 435986 | 90.456 | 24204 | 69.69 | 822303.591 | 7.407 | 56.781 |
| 40201 | 71.834 | 29017 | 69.69 | 825611.297 | 4.766 | 56.546 |
| 16 | 76.4522 | 27988 | 69.19 | 808640.368 | 4.689 | 54.929 |
| 231251 | 51.927 | 18337 | 69.69 | 820401.516 | 3.446 | 50.508 |
| 927139 | 58.230 | 26049 | 69.69 | 815680.168 | 5.056 | 59.879 |
| 123456789 | 60.152 | 25338 | 69.69 | 826982.464 | 5.305 | 55.934 |
| 256773292 | 59.510 | 23190 | 69.69 | 827095.791 | 5.625 | 54.002 |
| 330101 | 98.010 | 35165 | 69.69 | 827265.815 | 5.118 | 61.424 |
| 12131145 | 50.336 | 24695 | 69.69 | 833746.1977 | 1.9199 | 53.0411 |
| 1986 | 69.578 | 30640 | 63.63 | 811824.113 | 3.877 | 56.746 |
| 2018 | 49.380 | 22700 | 69.69 | 814691.094 | 2.554 | 52.995 |
| 1893 | 60.794 | 29015 | 62.12 | 847369.156 | 3.937 | 57.120 |
| 8913489 | 79.76 | 28309 | 69.69 | 830391.543 | 6.125 | 55.557 |
| 72929123 | 58.72 | 25635 | 69.69 | 821123.176 | 3.570 | 54.953 |
| 236236 | 68.39 | 25603 | 72.22 | 838037.96 | 4.640 | 56.984 |
| 37666 | 39.376 | 19059 | 69.69 | 827088.545 | 5.640 | 53.141 |
| 8894018411 | 57.80 | 24832 | 69.69 | 820214.612 | 3.927 | 53.182 |
| 5637 | 53.105 | 27954 | 69.69 | 820144.307 | 5.6738 | 56.569 |
| Total coverage |  |  | 72.22 |  |  |  |

TABLE II: Some test results obtained for Android DisplayingBitmaps app

For instance, in figure 4 we can conclude that the PkTest

[3] https://github.com/Orange-OpenSource/simiasque

[4] https://github.com/zubietaroberto/AndroidKeyStoreTest

application has a lower (but similar) execution time performance than the Material Library application. However, it has a higher energy consumption, even invoking only 6280 methods during the execution of the test, well below the 311,236 invoked by the latter. The hardware usage values (CPU, GPU, Memory) are higher for PkTest, and for GPU, the average usage percentage value (5.44 %) of this feature is 777.24% higher than the registered for the Material Library application. In this we can conclude that the use of this type of hardware also can have a considerable impact on the energy performance of an application.

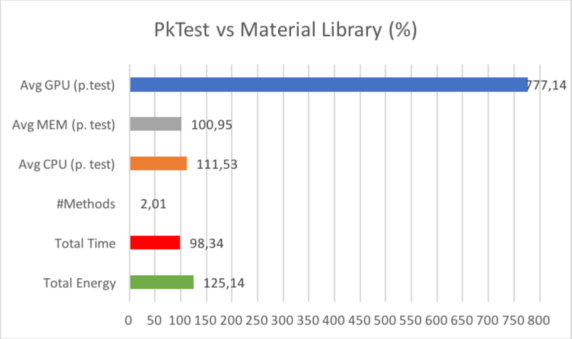


Fig. 4: Comparison between PkTest and Material Library

## IV. Threats to validity

Measuring the energy consumption of a mobile device is complex [13]. This is mostly due to the fact that it is quite difficult to fully isolate the code or application under measurement.

Today's operating systems, such as Android, have the ability to run multiple processes and applications simultaneously. Due to the difficulty of ensuring that during the execution of each test, only the intended application is running on the device and having an impact on its consumption, we register the state of the device before and after each test Following this approach, information about the status of the device is collected, which may interfere with the performance of the tests, like the number of processes running, percentage of CPU used and memory available.

Moreover, we executed the application in a factory-reseted device, with the lowest brightness level, to ensure the energy consumed by the display was as low as possible. We didn't considered testing in a root device, since we wanted to emulate a more realistic environment of execution. However, in order to avoid interferences, we did not provide Google account credentials, in order to avoid minimize the computations of Google services, like checking for updates.

Finding an adequate tool for energy profiling for the Android environment was also a challenge. The Android platform still lacks tools that allow developers to quickly and reliably monitor power consumption, as well locate energy hotspots in their code. Trepn is an accurate tool[14], capable of profile hardware usage (like GPS, WiFi and others), resources usage (memory, CPU) and power consumption of the system or even standalone Android applications, gets its power readings from the power management Integrated Circuit (PMIC) and the battery fuel gauge software. The main limitation of this profiler is that only gets accurate battery power readings from chipsets developed by Qualcomm, and the sampling rate can't be adjusted to less than 100 ms. However, we consider that is still the best free software-based alternative for this purpose, since this company dominates the smartphone SoC (System on a Chip) market due to date [15]. Another issue that we needed to solve was finding an approach to properly compare metrics applications of different types and domains. In order to avoid labeling and compare applications according to its domain and functionalities, we decided to exercise the User Interface of every applications and compare them by the respective obtained consumption. The tests were executed with the Application Exerciser Monkey, that allows to simulate user interaction and I/O events. In order to reach a relevant amount of method coverage to fairly compare executions, 20 equal tests (generated from the same seeds) were carried out for each one of the applications. If the defined level of method coverage was not reached after 20 tests, the process would continue for 30 more tests. These tests were executed using the same seeds, in order to generate the same sequence of events for every application.

## V. Conclusions

The main contributions of this work go from a tool capable of gathering relevant metrics and metadata to justify the consumption of code blocks of Android applications, to the development of an infrastructure capable of automating and gathering executions of this tool. We successfully implemented our methodology, resulting in a global infrastructure containing so far more than 600 Android applications and results from over 6,000 tests executed over some of these.

We were able to extend the GreenDroid framework to be capable of gather more information about the source code structure and become more expandable and precise. With the capability of easily integrate new testing frameworks and new energy profilers, like the Trepn Profiler, it became a tool that can be used for generically process Android Projects. It can be integrated in the testing phase of the development lifecycle of Android applications, helping developers to observe the energy and resources consumption, relating it to metrics obtained from dynamic and static analysis.

As a form of providing and share the results obtained, as well to prove and take advantage of the power of the AnaDroid, an open repository was developed. It contains hundreds of Android applications and respective results and metrics obtained with the execution of them (or portions) in a physical device. By agglomerating a high number of results, we pretend to obtain a set of information characterizing the Android development paradigm. This will allow to relate consumption with levels resource usage, to energetically compare different applications and devices and to obtain quality metrics of tests and software. In addition, it is hoped that the information retrieved from this repository may be (re)used in further works and researches.

## References

[1] J. W. Yoo and K. H. Park, "A cooperative clustering protocol for energy saving of mobile devices with wlan and bluetooth interfaces," *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 491–504, April 2011.

[2] M. Pedram, "Power minimization in ic design: Principles and applications," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 1, no. 1, pp. 3–56, Jan. 1996. [Online]. Available: http://doi.acm.org/10.1145/225871.225877

[3] *WOLFHPC '14: Proceedings of the Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing*. Piscataway, NJ, USA: IEEE Press, 2014.

[4] (2018) Smartphone market share. [Online]. Available: https://www.idc.com/promo/smartphone-market-share/os

[5] (2018) Top frustrations that lead to bad mobile app reviews. [Online]. Available: https://bit.ly/2QLoDTA

[6] J. C. J. P. F. Tiago Carção, Marco Couto and J. Saraiva, "Detecting anomalous energy consumption in android applications," 2014.

[7] D. D. Nucci, F. Palomba, A. Prota, A. Panichella, A. Zaidman, and A. D. Lucia, "Petra: A software-based tool for estimating the energy profile of android applications," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, May 2017, pp. 3–6.

[8] M. Linares-Vásquez, G. Bavota, C. Bernal-Cárdenas, R. Oliveto, M. Di Penta, and D. Poshyvanyk, "Mining energy-greedy api usage patterns in android apps: An empirical study," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: ACM, 2014, pp. 2–11. [Online]. Available: http://doi.acm.org/10.1145/2597073.2597085

[9] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang, "Fine-grained power modeling for smartphones using system call tracing," in *Proceedings of the Sixth Conference on Computer Systems*, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 153–168. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966460

[10] (2018) There are now more than 24,000 different android devices. [Online]. Available: https://bit.ly/2NNfPHQ

[11] M. Couto, J. Cunha, J. P. Fernandes, R. Pereira, and J. Saraiva, "Greendroid: A tool for analysing power consumption in the android ecosystem," in *2015 IEEE 13th International Scientific Conference on Informatics*, Nov 2015, pp. 73–78.

[12] J. Ossher, S. Bajracharya, E. Linstead, P. Baldi, and C. Lopes, "Sourcererdb: An aggregated repository of statically analyzed and cross-linked open source java projects," in *2009 6th IEEE International Working Conference on Mining Software Repositories*, May 2009, pp. 183–186.

[13] A. Banerjee and A. Roychoudhury, "Future of mobile software for smartphones and drones: Energy and performance," in *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems*. IEEE Press, 2017, pp. 1–12.

[14] A. R. Bakker, "Comparing energy profilers for android," in *Proceedings of 21st Twente student conference on IT, Enschede, The Netherlands*, 2014.

[15] (2018) Global smartphone system-on-chip (soc) revenue share by vendor. [Online]. Available: https://bit.ly/2yt4jMe

# Appendix

| Class | Method | Times invoked | CC | LoC | AndroidAPIs | N args |
|---|---|---|---|---|---|---|
| FlagsActivity | onCreate | 198 | 2 | 6 | 2 | 1 |
| BaseFlagFragment | validate | 1071 | 6 | 18 | 0 | 0 |
| VerifyPhoneFragment | onCreateView | 198 | 1 | 4 | 5 | 3 |
| CustomPhoneNumberFormattingTextWatcher | hasSeparator | 781 | 4 | 8 | 0 | 3 |
| BaseFlagFragment | onPostExecute | 135 | 0 | 1 | 0 | 0 |
| FlagsActivity | onOptionsItemSelected | 19 | 3 | 6 | 4 | 1 |
| BaseFlagFragment | onPhoneChanged | 715 | 0 | 1 | 0 | 0 |
| BaseFlagFragment | initCodes | 198 | 1 | 2 | 1 | 1 |
| CustomPhoneNumberFormattingTextWatcher | reformat | 715 | 6 | 24 | 0 | 2 |
| CountryAdapter | getView | 4224 | 2 | 8 | 7 | 3 |
| VerifyPhoneFragment | onActivityCreated | 198 | 1 | 3 | 1 | 1 |
| Country | getCountryCode | 87757 | 1 | 2 | 0 | 0 |
| CustomPhoneNumberFormattingTextWatcher | stopFormatting | 67 | 1 | 3 | 0 | 0 |
| CustomPhoneNumberFormattingTextWatcher | onTextChanged | 2155 | 4 | 7 | 0 | 4 |
| BaseFlagFragment | initUI | 198 | 1 | 38 | 16 | 1 |
| BaseFlagFragment | hideKeyboard | 1071 | 1 | 3 | 9 | 1 |
| Country | getCountryCodeStr | 44 | 1 | 2 | 0 | 0 |
| FlagsActivity | onCreateOptionsMenu | 198 | 1 | 3 | 1 | 1 |
| BaseFlagFragment | onItemSelected | 44 | 0 | 1 | 0 | 0 |
| Country | getPriority | 61 | 1 | 2 | 0 | 0 |
| BaseFlagFragment | doInBackground | 198 | 0 | 1 | 0 | 0 |
| CustomPhoneNumberFormattingTextWatcher | afterTextChanged | 2155 | 11 | 28 | 10 | 1 |
| CustomPhoneNumberFormattingTextWatcher | beforeTextChanged | 2155 | 4 | 7 | 0 | 4 |
| Country | getResId | 4177 | 1 | 2 | 0 | 0 |
| CustomPhoneNumberFormattingTextWatcher | getFormattedNumber | 2700 | 1 | 2 | 0 | 2 |
| VerifyPhoneFragment | send | 1071 | 3 | 10 | 4 | 0 |

TABLE III: Static metrics obtained for each invoked method during a test.

| Metric | Unit | Description |
|---|---|---|
| Consumption | J | Test or method the total consumption. |
| Time | ms | Test or method run time. |
| Method Coverage | % | For test-driven instrumentation, coverage at the method level is shown. |
| Wifi | 0-1 | If Wifi was used during the execution of the monitored block. |
| Mobile Data | 0-1 | If mobile data was used during the execution of the monitored block. |
| Screen State | 0-1 | If there was interaction with the screen. |
| Battery Charging | 0-1 | If the device was charging during execution. |
| Avg RSSI Level | dBm | average level of RSSI obtained. |
| Avg Memory Usage | B | Arithmetic mean of memory consumed. |
| Top Memory Usage | B | Peak of memory consumed. |
| Bluetooth | 0-1 | If Bluetooth was used during the monitored block execution. |
| Avg GPU Load | % | Average percentage of GPU usage. |
| Avg CPU Load | % | Average percentage of CPU utilization. |
| Top CPU Load | % | Max percentage of CPU utilization. |
| GPS | 0-1 | If GPS was used during the execution of the monitored block. |

TABLE IV: All dynamic metrics obtained for each test.

# The Future Architecture for Internet of Things- A Review

Akilu Rilwan Muhammad
*Instituto de Telecomunicações*
*Universidade do Porto*
Porto, Portugal
up201402359@fe.up.pt

***Abstract*—The Internet is probably the most successful computing innovations that forms the bedrock to most computing applications and services nowadays. Build on top of the Internet are several technologies bridging both geographical differences and time barriers. Cloud computing and the Internet of Things (IoT) are among such technologies delivering computing resources and services that would normally be difficult if not impossible to acquire, by a start-up organization, yet serve as a middle layer for developers to build domain-specific applications. FIWARE platform, an initiative of the European Commission evolves to aid developers minimize development time of domain-specific applications while exploiting state-of-the-art computing resources and artifacts. This paper presents a review of innovations in the Internet and Internet of Things paradigms, and exploits the potentials of FIWARE ecosystem for IoT architecture.**

***Keywords— Cloud computing, Internet of Things, FIWARE, Generic Enablers***

## I. Introduction

With the advances in Internet technology, wireless sensor networks, wearable & mobile computing gadgets and the adoption of Cloud computing technologies, present day Internet is faced with immense challenges that beyond any doubt call for its transformation and redesign [1] for Future Internet to succeed in the delivery of contents, services and scalability to meet the demand of technology consumers.

Whereas the Internet facilitates internetworking of computers around the globe eliminating geographical and time barriers, Cloud computing builds on top to provide pool of sophisticated computing resources to consumers. The integration of the aforementioned development today creates numerous opportunities such as the Internet of Things (IoT), facilitating internet connectivity for embedded devices and sensors. The present day Internet is heavily characterised by human-to-human communications means, the IoT is envisioned to move towards the realisation of machine-to-machine communications [2] connecting heterogeneous devices, leading to immense pressure for development of Future Internet standards. Cloud computing and Internet of Things are the leading research perspectives in the Future Internet landscape [3].

The European Commission thus established the Future Internet (FI-WARE) funded project, aimed to realise the Future Internet mission. This paper begins with a review of technological innovations that form the bedrock of IoT applications (Section II and III), presents a discussion on the FIWARE ecosystem and FIWARE components or Generic Enablers (GEs) to support IoT architecture (Section IV) and Section V concludes this paper.

## II. Cloud computing

The notion behind cloud computing is the provision of computing services and infrastructure to subscribing consumers as a service, against been supplied as a resources, [4] thereby promoting five distinctive features:

- *On-demand self-service:* entails the provision of self-managed computing resources to consumers.
- *Broad network access:* so applications are accessed through heterogeneous platforms (PCs, mobile devices, PDAs etc).
- *Resource pooling:* location independent pooling of computing resources to serve multiple cloud consumers in a multi-tenant fashion.
- *Rapid elasticity* of these resources and services provided to *quickly scale-out* and released to *quickly scale-in,* and
- *Metered services:* automatic control and optimisation of resource usage.

Cloud computing has long been envisioned to be the next generation of computing paradigm towards the provision of basic level computing services [5]. Cloud computing delivers encapsulated task as *"service"* to consumers in three different models: *Software as a Service, Platform as a Service* and *Infrastructure as a Service*, and open-up opportunities to application developers overcoming barriers to resource limitations. FIWARE platform results from the reap benefits of cloud computing innovations.

## III. Wireless sensor networks

Wireless Sensor Networks (WSN) specifically describes the representation of spatially dispersed sensors deployed to observe and record environmental conditions such as temperature, air quality, object tracking/monitoring [6] and send recorded measurement over the network to a central back-end. This is essential as it facilitates analysing recorded data set, visualisation as well as promote effective decision-making process and response to sudden environmental changes: air pollution for instance.

Wireless Sensor Networks technology has been acknowledged in recent studies as promising development towards achieving reliable and cost-effective remote monitoring [7] making it a good choice nowadays in the industry for such operations as industrial process monitoring and control, and machine health monitoring [8]. Sensor *nodes* in WSN are interconnected to one and often many other sensors, equipped with a *microcontroller* circuit to connect with other sensors, a *radio transceiver* with internal or sometimes external antenna, and a *power source*, such as a battery or an ambient power (e.g. solar powered).

WSN deployment has been embraced in diverse areas of application domain. Research [9] suggests that future objects would have embedded sensors in them such that they become smart objects. Thus, effective utilisation of the Future Internet and the realisation of envisioned IoT ecosystem requires the deployment of large-scale Wireless Sensor Networks infrastructure as its bedrock [10], Several domain areas of human endeavour applications already touched by the IoT revolutions: both personally at home and the Enterprise, Table 1 shows some areas of IoT applications.

TABLE I. IOT APPLICATIONS BY DOMAIN AREA

| Domain Area | Features | Examples |
|---|---|---|
| Agriculture | -Monitoring farm productivity<br>-Animal tracking<br>-Farm registration | Testbed [30] |
| Environment (Smart City) | -Pollution control<br>-Traffic management<br>-Waste disposal<br>-Tourism | StreetLamp [28]<br>SmartPort [13] |
| Energy | -Load balancing & energy management<br>-Operational decision-making | Smart Grid [7]<br>FINSENY [24]<br>SGAM [25] |
| Health | -Medical equipment monitoring<br>-Remote health service delivery.<br>-Care giver assistance | RPM [26]<br>iTaaS [29]<br>Doukas & Maglogia [27] |
| Industry | -Industrial process monitoring<br>-Equipment maintenance | Industry 4.0 [30] |

## IV. FIWARE ENABLERS FOR IOT DEVELOPMENT

Building an IoT architecture is largely a complicated process, partly due to the heterogeneity [17] of connected devices in terms of communication protocols supported and the constrained nature of device processing capability (e.g. limited memory and battery). Thus, development of a framework that abstracts device-specific peculiarities while on the other hand reducing application development time therefore becomes necessary. Fortunuátely, FIWARE ecosystem offers these framework in a way that IoT developers, service providers, enterprises and other organisations can develop products that satisfy their requirements. This section describes an architecture comprising different FIWARE generic enablers for IoT development.

### *A. The Orion Context Broker*

The Orion Context Broker is the core component of the FIWARE, the publish/subscribe reference implementation from the Data/Context Management GEs category, charged with the responsibility to gather context data [21] about entities and their attributes within the FIWARE components or services subscribed to it. An implementation of the NGSI9 and NGSI10 (*Next Generation Service Interface)* publish/subscribe REST API, Orion provides NGSI9 (*context availability*) and NGSI10 (*information exchange*) interfaces for clients [15] to perform such operations as:

- *Context provider registration* - (sensors for instance, to report temperature measures).
- *Context information updates* - (read and send temperature value).
- *Observe /notify* - inform Orion when changes occur.

Thus, Orion CB stores information about context (entities) registered and can provide updated information about the entity queried. Orion usually stores entity information as a MongoDB collection, a NoSQL document-oriented database system.

### *B. The IDAS Generic Enabler*

Device management in the FIWARE ecosystem is handled by the IDAS GE from IoT Service Enablement Chapter. An IoT Agent Gateway enabler for inter-networking and protocol conversion functionalities between devices and the IoT Backend GEs. Context information exchange within FIWARE platform is done using NGSI interface only, connecting objects or things therefore requires resolving the difference in FIWARE-device communication model. IDAS thus, provides support for several device-specific protocols to simplify device management and integration. Currently, IDAS provides the following agents for IoT devices [22]:

- *Ultralight 2.0 (UL2.0)* - a text-based messaging protocol IoT agent for low bandwidth devices with limited memory.
- *Lightweight M2M* - FIWARE standard IoT agent bridging communication between devices running OMA Lightweight M2M protocol and FIWARE components (NGSI).
- *JSON* - bridges HTTP/MQTT messaging for communication with devices using simple JSON protocol.
- *LoRaWAN* - facilitates information exchange and command with devices using LoRaWAN protocol.

### *C. The Cygnus Generic Enabler*

The Orion CB maintains current state context information only. For persisting historical data however, FIWARE provides the Cygnus GE. Cygnus in turn implements connectors supporting context data from Orion intended for persisting historical data third-party in storage engines such as PostgreSQL ,STH Comet, MySQL databases and HDFS back-end. The Cygnus simply connects FIWARE Orion with thirty-party persistence storage engine.

### *D. The Cosmos Generic Enabler*

FIWARE provides Cosmos (the Big Data GE) deployed to facilitates means for Big Data processing and analysis (batch and stream data). Whereas stream data is processed in near real-time, batch data is stored and processed at a later time.

### *E. Short Term Historic (STH) Comet Generic Enabler*

This a FIWARE component intended for storage and retrieval of time series aggregated historic data about entities and their attributes within the Orion CB.

As FIWARE runs within Docker container [23], an operating system-level lightweight virtualization technology alternative to hypervisor-based, these components are usually containerised to run and managed via Docker commands. Figure 1 shows a simplified architecture of a FIWARE-based IoT system.

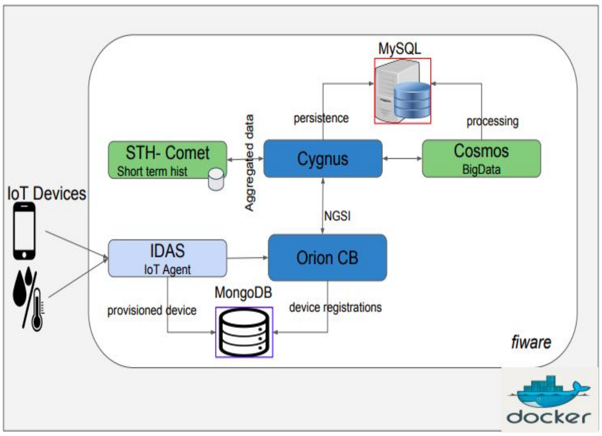


Fig. 1. Simplified FIWARE-powered IoT Architecture with MySQL persistence.

## V. CONCLUSION

The overall goal of this paper is to review the trends in Internet and Internet-based technologies, and discusses the potential of using the FIWARE ecosystem for the development of Internet of Things (IoT) applications. FIWARE ecosystem is a broad and complex platform with an immense collection of tools and services that require time and effort to understand, this paper serves to provide a guideline about most commonly used enablers and the purposes they serve. With the current trend in the Internet of Things market and widespread of connected devices, FIWARE platform has proven to be the next generation of IoT applications development. The paper also describes the bedrock technologies that give birth to the FIWARE platform and discusses major generic enablers used in IoT ecosystem.

## REFERENCES

[1] Tsai Chun-Wei, Lai Chin-Feng, and Athanasios V. Vasilakos. “Future Internet of Things: open issues and challenges”. Wireless Networks. November 2014, 20(8), pp. 2201–2217.

[2] Khan R, Khan SU, Zaheer R, Khan S. Future Internet: “The Internet of Things Architecture, Possible Applications and Key Challenges”. 2012 10th International Conference on Frontiers of Information Technology, 2012. doi:10.1109/fit.2012.53.

[3] Sotiriadis S, Stravoskoufos K, Petrakis EGM. Future Internet Systems Design and Implementation: Cloud and IoT Services Based on IoT-A and FIWARE. Designing, Developing, and Facilitating Smart Cities, 2016, p. 193–207.

[4] Marston S, Li Z, Bandyopadhyay S, Ghalsasi A. Cloud Computing - The Business Perspective. 2011 44th Hawaii International Conference on System Sciences, 2011. doi:10.1109/hicss.2011.102.

[5] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener Comput Syst 2009;25:599–616.

[6] Han G, Jiang J, Zhang C, Duong TQ, Guizani M, Karagiannidis GK. A Survey on Mobile Anchor Node Assisted Localization in Wireless Sensor Networks. IEEE Communications Surveys & Tutorials 2016;18:2220–43.

[7] Fadel E, Gungor VC, Nassef L, Akkari N, Abbas Malik MG, Almasri S, et al. A survey on wireless sensor networks for smart grid. Comput Commun 2015;71:22–33.

[8] Mini S, Udgata SK, Sabat SL. Sensor Deployment and Scheduling for Target Coverage Problem in Wireless Sensor Networks. IEEE Sens J 2014;14:636–44.

[9] Rawat P, Singh KD, Chaouchi H, Bonnin JM. Wireless sensor networks: A survey on recent developments and potential synergies. Journal of Supercomputing 2013;68:1–48.

[10] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Gener Comput Syst 2013;29:1645–60.

[11] About Us - FIWARE. FIWARE n.d. https://www.fiware.org/about-us/ (accessed November 10, 2018).

[12] Fernández P, Santana J, Ortega S, Trujillo A, Suárez J, Domínguez C, et al. SmartPort: A Platform for Sensor Data Monitoring in a Seaport Based on FIWARE. Sensors 2016;16:417.

[13] FIWARE Architecture - FIWARE Forge Wiki https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE_Architecture (accessed December 6, 2018).

[14] Rosangela de Fatima Pereira Marquesone, Tereza Cristina M. B Carvalho, Lucas Batista Guimaraes, and Eduardo Mario Dias. A FIWARE-Based Component for Data Analysis in Smart Mobility Context. 2017 IEEE First Summer School on Smart Cities (S3C), 2017. doi:10.1109/s3c.2017.8501373.

[15] Salhofer P. Evaluating the FIWARE Platform. Proceedings of the 51st Hawaii International Conference on System Sciences, 2018. doi:10.24251/hicss.2018.726.

[16] Alvarez F. FIWARE overview: examples of usage of big data, and the benefits for smart cities. 5th EU-Japan Symposium on ICT Research and Innovation, 2014.

[17] Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M. Internet of Things for Smart Cities. IEEE Internet of Things Journal 2014;1:22–32.

[18] Zahariadis T, Papadimitriou D, Tschofenig H, Haller S, Daras P, Stamoulis GD, et al. Towards a Future Internet Architecture. Lecture Notes in Computer Science, 2011, p. 7–18.

[19] Santos IL, Alves MP, Flavia C, Li W, Y ZA, Khan SU. A System Architecture for Cloud of Sensors. IEEE 16th Int. Conf. on Dependable, Autonomic & Secure Comp., 4th Int. Conf. on Big Data Intelligence & Comp, 2018, p. 666–72.

[20] Hernández-Muñoz JM, Vercher JB, Muñoz L, Galache JA, Presser M, Hernández Gómez LA, et al. Smart Cities at the Forefront of the Future Internet. Lecture Notes in Computer Science, 2011, p. 447–62.

[21] Bellabas A, Ramparany F, Arndt M. Fiware Infrastructure for Smart Home Applications. Communications in Computer and Information Science, 2013, p. 308–12.

[22] Fiware. Fiware/tutorials.IoT-Agent. GitHub n.d. https://github.com/Fiware/tutorials.IoT-Agent (accessed December 16, 2018).

[23] Morabito R. A performance evaluation of container technologies on Internet of Things devices. 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016. doi:10.1109/infcomw.2016.7562228.

[24] FINSENY | Future Internet for Smart Energy http://www.fi-ppp-finseny.eu/ (accessed December 17, 2018).

[25] Smart Grid Architecture Model (SGAM). Smart Grid Reference Architecture, November 2012. https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf

[26] Fazio M, Celesti A, Marquez FG, Glikson A, Villari M. Exploiting the FIWARE cloud platform to develop a remote patient monitoring system. 2015 IEEE Symposium on Computers and Communication (ISCC), 2015. doi:10.1109/iscc.2015.7405526.

[27] Doukas C, Maglogiannis I. Bringing IoT and Cloud Computing towards Pervasive Healthcare. 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2012. doi:10.1109/imis.2012.26.

[28] Ramparany F, Marquez FG, Soriano J, Elsaleh T. Handling smart environment devices, data and services at the semantic level with the FI-WARE core platform. 2014 IEEE International Conference on Big Data (Big Data), 2014. doi:10.1109/bigdata.2014.7004417.

[29] Martínez R, Pastor JÁ, Álvarez B, Iborra A. A Testbed to Evaluate the FIWARE-Based IoT Platform in the Domain of Precision Agriculture. Sensors 2016;16. doi:10.3390/s16111979.

[30] Lu,Yang. "Industry 4.0: A survey on technologies, applications and open research issues". Journal of Industrial Information Integration. 2017; 6: 1-10.

# An overview of Sentiment Analysis Approaches

Shamsuddeen Hassan Muhammad
*Department of Software Engineering*
*Bayero University , Kano*
Kano, Nigeria
shmuhammad.csc@buk.edu.ng

***Abstract*—Sentiment analysis is a relatively new field of study at the intersection of computer science and linguistics that aims to find an opinion expressed in a text. It has received a swell of interest in both academia and industry. This paper provides an overview of the basic approaches for sentiment analysis task: machine learning-based approach and lexicon-based approach. The machine learning approach is based on training models on corpora annotated with polarity information and the lexicon-based approach is based on using sentiment lexicon. Recently, a hybrid approach is employed to leverage the strength of both two approaches**

***Index Terms*—sentiment analysis, opinion mining, lexicon-based, domain-specific, machine learning**

## I. Introduction

The exponential growth of social media and dynamic websites has changed how people use an Internet from read-only participation to read-write participation. Participants now actively contribute their opinion rather than reading the Web content passively [1].This digital revolution and paradigm shift allows users to express their opinion and sentiment on many areas such as government, commerce, politics, education, health and many entities [2]. For example, on average, 6,000 tweets per-second are made on Twitter and 91.8 million blog posts are published every month on Wordpress only [2]. These raise the need to find user's sentiments through such a medium.

Traditionally, people, businesses and government use approaches such as survey to find feedback or opinion on a particular subject. For example, if people want to buy a new mobile device, they consult their friends, relatives or acquaintance who had bought a similar product or service for an opinion and recommendation which can be positive, negative or neutral.They use the feedback received to determine the worthiness of the product to prevent disappointment. However, single or few opinions may be biased. In the same way, businesses conduct survey and opinion poll to find users' opinion on their product with a view to improve customer services and marketing strategy. Also, government uses a survey to find people reaction and acceptance towards new and existing policies. However, with the rapid increase of user-generated and opinionated text, the classical tools such as survey and traditional Natural Language Processing techniques(NLP) for analysing and understanding users sentiment or opinion are sub-optimal [3] [4]. To this end, an efficient way of finding user sentiment from text is needed [5].

Sentiment analysis (SA) is a study that aims to find sentiment, opinion, emotion, attitude computationally from written text [6]. It is an offshoot of natural language processing. Depending on the domain ,it is often referred to with a different nomenclature such as: *opinion mining, opinion analysis, opinion extraction, sentiment mining, sentiment extraction, subjectivity analysis, emotion analysis, review mining* and many more terms are evolving. Two most widely used names that appear in the academic are sentiment analysis and opinion mining. In contrast, only the term sentiment analysis is widely used in industry [7].

Early research on textual information processing focused on mining and retrieval of factual information, such as information retrieval, text classification or text clustering. Research in sentiment analysis started relatively in the year 2001 [13] [8] and the phrase *"opinion mining"* was first use in 2003 [14]. In [15], they reported that 99% of all the research on sentiment analysis have been published after the year 2004. Fig. 1 highlights most prominent areas of research in the area of sentiment analysis.

Pang and Lee [8] identify three factors which triggered interest in sentiment analysis research. First, the rise of machine learning methods in natural language processing and information retrieval; Second, the availability of datasets for machine learning algorithms to be trained on. Thirdly, the realization of the fascinating intellectual challenges and intelligence applications that the area offers.

Sentiment analysis has been successfully applied in many domain and applications such as recommender systems, user reviews and politics [8].Businesses also use sentiment analysis to find consumer opinion on product and services to improve their business and service delivery [9].

There are two basic approaches for sentiment analysis; lexicon-based approach and machine Learning-based approach.The machine learning approach is based on training models with corpora annotated with polarity information. The Lexicon-based approach is based on using polarity of lexicons and it provides better accuracy [10]. Recently, a hybrid approach has been proposed and it exploit the strength of two or more techniques to offer better performance [11].

This paper aims to provides an overview of the sentiment analysis approaches. Section II presents different levels of sentiment analysis. Section III discusses the three approaches of sentiment analysis. Finally, section IV concludes the paper.

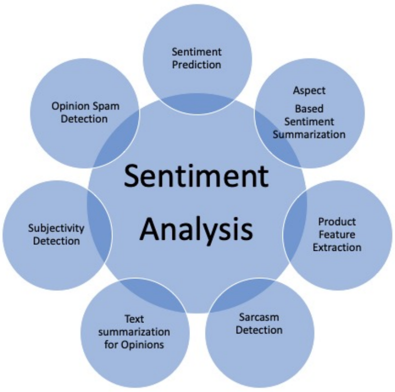


Fig. 1. Areas of research in sentiment analysis

## II. Levels of Sentiment Analysis

According to [6], based on the levels of granularity, the sentiment analysis has been investigated at three different levels: document level, sentence level and aspect level. In contrast, Kumar & Sebastian [17] reported four different levels, with the addition of word level as depicted in Fig. 2.

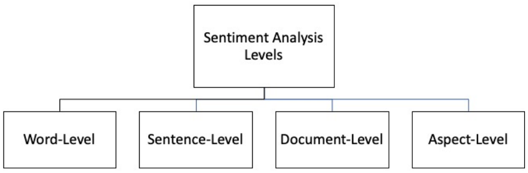


Fig. 2. Four Levels of sentiment analysis

*1) Word Level:* Sentiment analysis at word level involves finding adjective as a source of sentiment indicator. In the same way, other part-of-speech such as a noun, verb and adverb sometimes indicates subjectivity and opinion [16].

*2) Sentence level:* Sentiment analysis at sentence level deals with an opinion expressed in each sentence within a document. It finds the polarity of each sentence as positive, negative or neutral. Subjectivity classification is closely related to this concept; it deals with categorising sentences as either subjective sentences or objective sentences. However, subjectivity is different from sentiment because some objective statement may imply opinions e.g. ”I bought a new computer yesterday and the battery does not last long” [6].

*3) Document Level:* In document level, the whole contents of the document are summarized to a single opinion. Therefore, it is assumed that each document contains an opinion on a single entity, thus, sentiment analysis at this level is not practicable for a document that contains sentiment on multiple entities [6].

*4) Aspect level:* This level of analysis is more difficult than sentence level and document level analysis. It is sometimes called feature level or feature-based opinion mining and summarization [17]. This level identified that any opinion without targets is meaningless and each opinion contains a target and sentiment. Therefore, the aim of the aspect level is to find sentiments on entities and/or their aspect. For example, the sentence *”The iPhones call quality is good, but its battery life is short ”* evaluates two aspects, call quality and battery life of iPhone (entity). The sentiment on iPhones call quality is positive, but the sentiment on its battery life is negative. The call quality and battery life of iPhone are the opinion targets [6].

## III. Approaches to Sentiment Analysis

There are two basic approaches to perform sentiment analysis. Machine learning-based and lexicon-based approach as shown in Fig. 3. Recently, hybrid approach of sentiment analysis has been explored to leverage the advantages of both machine learning and lexicon-base approaches [3] [11].

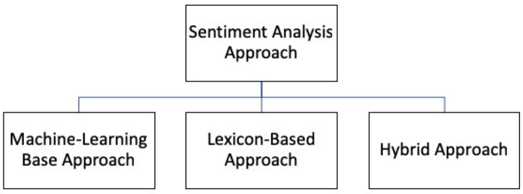


Fig. 3. Three approaches two sentiment analysis

### *A. Machine Learning Approach for Sentiment Analysis*

Considerable research in sentiment analysis uses machine learning approach to perform sentiment classification. We briefly expound both the two approaches of supervised and unsupervised machine learning methods.

*1) Supervised sentiment classification :* Text classification has long been an existing research field and the task of sentiment classification is similar to the text classification. Text classification classifies text base on topics such as politics, religion and sports while sentiment classification classifies text to categories (classes) such as excellent, good neutral, bad and very bad. Some studies use numeric sentiment polarity values [7].

Similar to text classification method, supervised sentiment classification method uses a learning algorithm trained with sentiment-labelled data to classify an unseen document. The typical process of sentiment classification is shown in Fig. 4. First, standard text pre-processing, feature engineering and vector-space representation are applied to the training and test documents drawn from a problem domain. After that, a machine learning algorithm is employed to learn prediction model during a training phase. The model is then used in the testing phase to do classification (or regression) of unseen documents. One of the most important steps in the sentiment classification process outline is feature engineering. Feature engineering uses existing knowledge from the problem domain and create features that make machine learning more effective.The feature engineering process involves three phases of activates: feature discovery, feature selection and feature weighting.

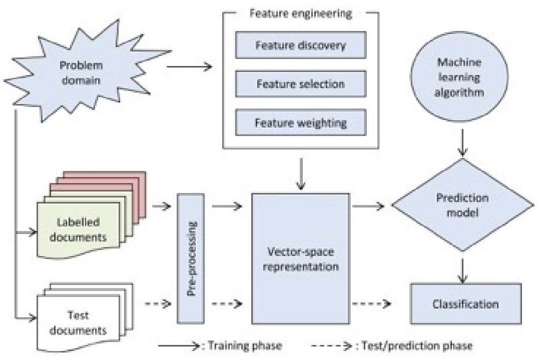


Fig. 4. Supervised sentiment classification method [33]

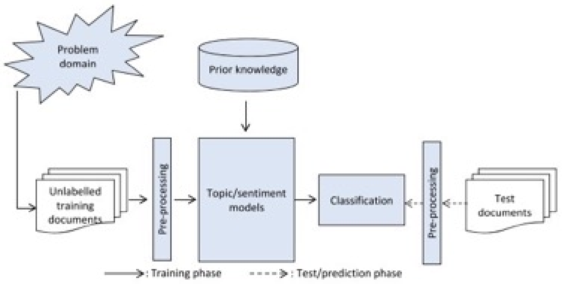


Fig. 5. Unsupervised sentiment classification method [33]

Previous research on sentiment analysis focus on using the standard machine learning algorithms such as *Naïve Bayes, Maximum Entropy and Support Vector Machine*. One of the pioneer study that experiments the three techniques performance on sentiment analysis task [13] reported that standard machine learning techniques perform better than human-produced baselines. However, the three machine learning methods performed poorly on sentiment classification compare to traditional topic-based categorization. The sub-optimal performance indicates that sentiment classification task is more difficult than topic classification. This is because topic can be easily identify by keyword alone while sentiment can be expressed in a subtler manner. For instance, *How could anyone sit through this movie?* contains no single word that is obviously negative. Hence fine-grain analysis is required with sentiment classification.

Recently, dedicated supervised methods for sentiment classification has been developed to improve accuracy. One of the techniques use score function [14]. The approach starts by training a classifier using a corpus of self-tagged reviews drawn from websites. Thereafter, the same corpus is then employed to improve their classifier before applying it to sentences obtain from web searches. Authors experimental result accuracy outperforms the traditional machine learning algorithms approaches.

*2) Unsupervised sentiment classification:* The unsupervised sentiment classification process is shown in Fig. 5. At the training phase, unlabelled documents are pre-processed and probabilistic topic modelling methods are employed to detect both topic and sentiment. Prior knowledge in the form of seed sentiment-bearing terms is required to guide the process. Consequently, the sentiment class of a text document can be determined based on the topic used to compose the document. Standard topic modelling approaches assume a three-layered hierarchical framework, where topics are associated with documents, and words are associated with topics. For sentiment detection, this framework is extended with an additional sentiment layer in between documents and topics or with sentiment classes as an additional topic model [26].

One of the pioneering unsupervised learning methods was proposed by Turney [13]. It is a simple unsupervised learning algorithm that classifies reviews base on the average semantic orientation of the phrases that contain adjectives and verbs. It classifies review as recommended if positive and not recommended if negative. The accuracy for machine learning-based approaches for sentiment analysis is not up to the mark compare to the lexicon-based approach [4].

### B. Lexicon-based approach (Linguistic Approach)

The lexicon-based approach is based on using sentiment lexicon generated from either corpus or dictionary. It is sometimes called corpus-based approach if it uses lexicons generated from corpus or dictionary-based approach if it uses lexicons generated from a dictionary. It is workflow is shown in Fig. 6. The first step is the creation of sentiment lexicon or adoption of an existing one (which is mostly the case by many researchers). The next step is to pre-process the document to be classified and each word in the document is assigned the corresponding prior polarity from the sentiment lexicon. Finally, the prior polarities are adjusted to reflect contextual polarities (contextual analysis) and sum-up to find the sentiment orientation of the document. The sentiment orientation of the document is classified as either positive if the sum is positive or negative if the sum is negative and neutral if the final sum is 0. Variation of this exist and the difference is mainly based on what value is assigned to sentiment words in sentiment lexicon, how negation is handled etc., with the rapid increase of automatic generation of domain-specific lexicon, the lexicon-based approach is now leverage to provide better accuracy [27].

*1) Sentiment Lexicon:* Sentiment lexicon (lexical resource) is a dictionary of a lexical item with corresponding semantic orientation. It plays a significant role in sentiment analysis task. The lexical item conveys a single meaning and it can be words (e.g. good and bad), word senses, phrases (I am over the moon, it arrived) and idiomatic expression. The semantic orientation can be in several forms such as words (positive, negative or neutral) and phrases (strongly positive, mildly positive and strongly negative ). A specific range of values is also used to indicate a ranking of the sentiment strength. For example, using 1 to 5 ranking with 1 has least ranking strength and 5 has the highest ranking strength. In this scenario, 3 being the middle is considered neutral [18]. Also, the accuracy

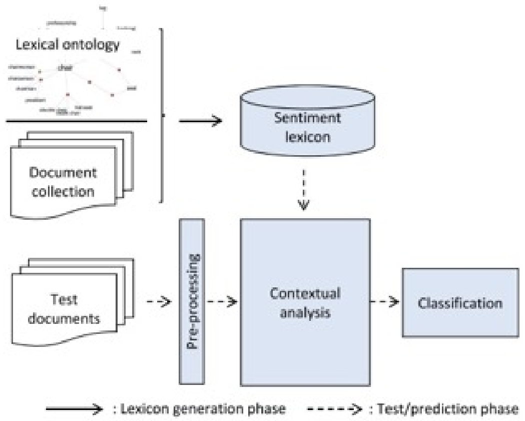


Fig. 6. Lexicon-based method for sentiment analysis [33]

of lexicon-base approach depends on the type of sentiment lexicons use.

*a) General Purpose Lexicon:* These are lexicons develop and use in sentiment analysis without any relation between the domain in question and the lexicon; they are nonspecific and can be used to find an opinion across domains such as a movie, social media, health and government. However, with the advantage of wide coverage, the lexicon losses accuracy because sentiment words are often domain-dependent [28]. A single word in one domain may contain positive polarity and contain negative polarity in another domain. Hence, the choice of word orientation is define by the domain in which the sentiment word is used. For example, the word *"suck"* appears to have negative and positive orientation in the following sentences: *"the camera we bought sucks,"*, the word *"suck"* have negative orientation in this sentence, but it can be used with positive orientation as in *"The vacuum cleaner we bought really sucks"*.

To exacerbate this problem, sentiment lexicons may have different sentiment orientation within the same domain. This makes the task of sentiment analysis even more difficult. For example, in camera domain, the word "long" have a different orientation in the following sentences. *"The battery life is long and It takes a long time to focus"*. The first sentence indicates positive opinion while the latter indicates negative opinion [24]. Consequently, several researchers use domain-specific lexicon for better accuracy.

*b) Domain-Specific Lexicon:* Due to the increase of application of sentiment analysis in several domains couple with need for accuracy , domain-specific lexicons are leveraged. They are generated from general purpose lexicon or a new one is generated from scratch. They increase the accuracy of sentiment analysis task. However, manual generation of the domain-specific lexicon for each domain is a laborious and mind-numbing task. To this effect, automatic methods for generation of domain-specific has been explored [12], [31], [32].

*2) Sentiment Lexicon Generation Methods:*

*a) Manual Generation of Sentiment Lexicon:* This approach consists of using an existing dictionary or corpus and manually select lexical items that have sentiment orientation. Thereafter, the lexical items are annotated manually with predefined sentiment strength.For example, 5-class sentiment strength is (-2 to +2). Because humans rather than machine annotate each lexical item , this method provides accuracy. Hence, sentiment analysis with manually generated lexicon achieve better performance. However, the method has drawback. It is time-consuming and daunting due to the inherent nature of manual activity involve. Also, it is limited to small coverage compared to automatically generated lexicon [18].

*b) Automatic Generation of Sentiment Lexicon:* In this approach, sentiment lexicon is generated automatically , therefore it eliminates time spend in manual approach. One of the popular automatic approach uses a seed word from which other sentiment words are generated automatically. Bootstrap approach is also employ and automatically ranks words based on a similarity measures [19]

*3) Examples of Sentiment Lexicon:* Some of the widely adopted sentiment lexicons are briefly explain in this section.

*a) WordNet:* This is an online English General lexical resource database. It contains adjectives, nouns and verbs group into synonyms set through semantic relation [20].

*b) SentiWordNet:* SentiWordNet is a lexical resource with a high level of coverage developed by Esuli and Sebastiani [21]. Positive, negative and neutral are three sentiment orientation used for each synset. It was developed from the *WordNet*. In SentiWordNet, words may contain different meaning and therefore different polarity. For example, "cold" may mean having a low temperature as in cold beer or without human warmth or emotion as in cold person. SentiWord uses glosses for each word entry to distinguish one from another [21].

*c) WordNet-Affect:* WordNet-Affect lexicon was created originally from WordNet synsets [22]. It consists of "Affective Knowledge" which describes moods, feelings and attitude. WordNet-Affect is one of the widely use lexicons because it is not limited to single-word concepts.

*d) SenticNet:* SenticNet is publicly available lexical resource for concept-level sentiment analysis. The lexicon includes both semantic and affective lexical unit. It provides over 30,000 multi-word expressions to enable fine-grain analysis of natural language opinion. It uses sentiment orientation between -1 and 1(-1 being extremely negative and +1 extremely positive) [23].

*e) General Inquirer:* This is a General Lexical system developed at Harvard for content analysis research in the behavioural sciences. The system uses two dictionaries: psycho-sociological dictionary and an anthropological dictionary used for studying themes in the folktales of many traditions and culture. The two dictionaries contain category of words. During sentence analysis, General Inquirer look-up these dictionaries and find in which category, the word belongs if it exists [24].

*f) Bing Lius Opinion Lexicon:* This is freely available sentiment lexicon developed by Bin Liu. It consists of English opinion lexicon being developed continuously. The lexicon contains a list of positive and negative words close to 6800 [25].

*4) Domain Specific Lexicon-based Sentiment Analysis:* The lexicon-based sentiment analysis approach performance reaches an optimal expectation when domain-specific lexicon is leverage. On the other hand, it gives sub-optimal performance when general purpose lexicon is leverage. To this effect, many research work on lexicon-based approach leverage domain-specific lexicons for better accuracy [12], [31], [32]. The approach is sometimes called: *Domain-dependent, Context-dependent, Domain-Oriented, Domain-based or Target specific lexicon-based approach*.

In [12], domain-specific lexicon from movie review corpora is automatically created and experimental results performance shows improvement over the manually created sentiment lexicons. Their method involves two steps, they generate corpus-based lexicon and each sentiment-bearing word is labelled with both positive or negative and polarity weight. Secondly, the lexicon is used in sentiment classification which shows improvement. Their approach is domain agnostic , therefore, very useful in creating domain-specific lexicons in many domains.

In the same way [29] proposed an approach for generating domain-specific lexicon through double propagation. Firstly, the technique uses a seed word to extract sentiment-bearing words and features. The extracted lexical items are then used iteratively to find new sentiment word and features until sentiment-bearing words are exhausted. They also proposed a method that assign polarity level to the sentiment words identified during sentiment extraction. Both proposed approaches provide satisfactory performance.

The study [11] devised a novel approach that exploits the idea of context coherency to automatically build a domain-focused lexicon for sentiment analysis. The context coherency is a phenomenon which explain that words with same polarity seems to always appear adjacent within context. The reported accuracy of this approach is 94% and proved to be effective and can be easily adopted in a different domain

A recent study [30] introduced a new domain-specific generation method from unlabelled review data. This approach is divided into two part, the first task is labelling the training reviews with polar values (negative and positive) and lexical unit with the higher ranking score are selected and used as training data. The second task uses the selected training data to obtain new domain-focused sentiment lexicon. The approach offered better performance when compared with other domain-specific lexicon base approach that uses SentMI and SenProf lexicon

### C. Hybrid Approach

Until recently, a hybrid approach for sentiment analysis has been explored by researchers. They combine the strength of sentiment analysis approach for optimal accuracy. In their work [28], they leverage the strength of rule-based classification and supervised learning . The combined approach achieved higher accuracy when experiments on movie reviews, product reviews and Myspace comments. In [3], they perform Twitter sentiment analysis with a combination of lexicon-based approach and trained classifier. They claimed their result performs better than state-of-the-art baseline.

## IV. Conclusion

We discussed an overview of sentiment analysis and the approaches to performing it in this paper. It is a sub-field of natural language processing that finds an opinion on human written text. It has been employ in different areas such as business and government. At a basic level, there are two approaches to perform sentiment analysis. Machine learning-based approach and lexicon-based approach. Until recently, a hybrid approach has been explored that combine the strength of two or more methods. The lexicon-based approach performance has been shown to outperform machine-learning approach when domain specific lexicons are employed. But, creating the domain-specific lexicon is a tedious and boring task. Therefore, as a solution to a manual generation of sentiment lexicon, novel approaches for automatic construction of domain-specific lexicon methods has been explored from recent literature

## References

[1] A. Darwish, K. L. J. O. A. I. I. technology, 2011, The impact of the new Web 2.0 technologies in communication, development, and revolutions of societies, Citeseer

[2] K leen 121 Amazing Social Media Statistics and Facts.

[3] Combining Lexicon- and Learning-based Approaches for Improved Performance and Convenience in Sentiment Classification, pp. 150, May 2015.

[4] A. C. Forte and P. B. Brazdil, Determining the Level of Clients Dissatisfaction from Their Commentaries, in Computational Processing of the Portuguese Language, vol. 9727, no. 2, Cham: Springer, Cham, 2016, pp. 7485.

[5] B. L.2010, Sentiment Analysis and Subjectivity., Handbook of Natural Language Processing.

[6] B. Liu, Sentiment Analysis. Cambridge: Cambridge University Press, 2015, pp. 1384.

[7] B. Liu and L. Zhang, A Survey of Opinion Mining and Sentiment Analysis, in Mining Text Data, no. 13, Boston, MA: Springer, Boston, MA, 2012, pp. 415463.

[8] B. Pang and L. Lee, Opinion Mining and Sentiment Analysis, FNT in Information Retrieval, vol. 2, no. 1, pp. 1135, Jul. 2008.

[9] A Practical Guide to Sentiment Analysis, pp. 1199, Apr. 2017.

[10] J. Smith, Contextual Lexicon-based Sentiment Analysis for Social Media, pp. 1147, May 2016.

[11] A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level, pp. 16, May 2018.

[12] S. Almatarneh and P. Gamallo, Automatic Construction of Domain-Specific Sentiment Lexicons for Polarity Classification, in Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS 2017, vol. 619, no. 4, Cham: Springer, Cham, 2017, pp. 175182.

[13] B. Pang, L. Lee, and S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in Proceedings of the 2002 conference on Emperical Methods in Natural Language Processing, 2002, pp. 7986.

[14] K. Dave, S. Lawrence, and D. M. Pennock, Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews, in Word Journal Of The International Linguistic Association, 2003, vol. 17, no. 5, pp. 519528.

[15] M. V. Mntyl, D. Graziotin, and M. Kuutila, The evolution of sentiment analysisA review of research topics, venues, and top cited papers, Comput. Sci. Rev., vol. 27, pp. 1632, 2018.

[16] A. Kumar, T. S. I. J. O. I. Systems, 2012, Sentiment analysis: A perspective on its past, present and future, mecs-press.net.

[17] B. Liu, Sentiment Analysis and Opinion Mining, Synth. Lect. Hum. Lang. Technol., vol. 5, no. 1, pp. 1167, 2012.

[18] S. Ahire, A Survey of Sentiment Lexicons, 2000.

[19] C. Banea, R. Mihalcea, J. W. LREC, 2008, A Bootstrapping Method for Building Subjectivity Lexicons for Languages with Scarce Resources., digital.library.unt.edu.

[20] G. A. Miller, WordNet: a lexical database for English, Communications of the ACM, vol. 38, no. 11, pp. 3941, Nov. 1995.

[21] S. Baccianella, A. Esuli, and F. Sebastiani, SentiWordNet 3 . 0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining SentiWordNet, Analysis, pp. 112, 2010.

[22] C. Strapparava and A. Valitutti, WordNet-Affect: an affective extension of WordNet, Proc. 4th Int. Conf. Lang. Resour. Eval., 2004.

[23] SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-Driven Sentiment Analysis.

[24] M. S. Smith, D. M. Ogilvia, P. J. Stone, D. C. Dunphy, and J. J. Hartman, The General Inquirer: A Computer Approach to Content Analysis., American Sociological Review.

[25] M. Hu and B. Liu, Mining and summarizing customer reviews. New York, New York, USA: ACM, 2004, pp. 168177.

[26] E. Cambria, D. Das, S. Bandyopadhyay, and A. (Editors) Feraco, A Practical Guide to Sentiment Analysis (Socio-Affecting Computing 5). 2017.

[27] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, Lexicon-Based Methods for Sentiment Analysis, vol. 37, no. 2, pp. 267307, May 2011.

[28] R. Prabowo, M. T. J. O. Informetrics, 2009, Sentiment analysis: A combined approach, Elsevier

[29] G. Qiu, B. L. 0001, J. Bu, and C. Chen, Expanding Domain Sentiment Lexicon through Double Propagation., IJCAI, pp. 11991204, 2009.

[30] H. Han, J. Zhang, J. Yang, Y. Shen, and Y. Zhang, Generate domain-specific sentiment lexicon for review sentiment analysis, Multimedia Tools and Applications, vol. 77, no. 16, pp. 2126521280, Aug. 2018.

[31] H. Kanayama, T. N. P. O. T. 2. C. on, 2006, Fully automatic lexicon expansion for domain-oriented sentiment analysis, dl.acm.org.

[32] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai, Automatic construction of a context-aware sentiment lexicon: an optimization approach. New York, New York, USA: ACM, 2011, pp. 347356.

[33] MUHAMMAD, A.B. 2016. Contextual lexicon-based sentiment analysis for social media. Robert Gordon University, PhD thesis.

# Survey on Well-known User-interface Design Rules

Amir Rastegar Lari

Department of Computer Science, University of Porto, Porto, Portugal

**Abstract— In this article, we investigate some well-known User-Interface (UI) design rules. User interface design or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. User-Interface design is not a straightforward process. Design rules often are based on goals in compare to actions. Usually, it is not possible to satisfy all the goals, a tradeoff is needed. Most of the rules are based on users and their goals. Some of them use funny approaches like gamification to engage using application more. Designing user-interface well is the main approach in developing a system and an application in Human-Computer interaction field and to assess how easy user interface design is to use, usability testing could be applied.**

***Keywords- User-Interface Design, Users, Tasks***

## I. Introduction

The goal of user interface design(UI) is to design user interfaces for different software platforms such as mobile applications, desktop software, and web applications with the aim of maximizing User-Experience (UX), usability and user engagement. UX refers to a person's emotions and attitudes about using a particular product, system or service. It includes the practical, experiential, effective, meaningful and valuable aspects of human-computer interaction and product ownership. Additionally, it includes a person’s perceptions of system aspects such as utility, ease of use and efficiency. User experience may be considered subjective in nature to the degree that it is about individual perception and thought with respect to the system. User experience is dynamic as it is constantly modified over time due to changing usage circumstances and changes to individual systems as well as the wider usage context in which they can be found. In the end, the user experience is about how the user interacts with and experiences the product. Usability is the ease of use and learnability of a human-made object such as a tool or device.[1]

In software engineering, usability is the degree to which software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.[2]

The object of use can be a software application, website, book, tool, machine, process, vehicle, or anything a human interacts with. A usability study may be conducted as a primary job function by a usability analyst or as a secondary job function by designers, technical writers, marketing personnel, and others. It is widely used in consumer electronics, communication, and knowledge transfer objects (such as a cookbook, a document or online help) and mechanical objects such as a door handle or a hammer. Usability includes methods of measuring usability, such as needs analysis and the study of the principles behind an object's perceived efficiency or elegance. In human-computer interaction and computer science, usability studies the elegance and clarity with which the interaction with a computer program or a web site (web usability) is designed. Usability considers user satisfaction and utility as quality components, and aims to improve user experience through iterative design.

The main objective of user interface design is to simple human computer interaction that user could easily achieve the goal easily with minimum waste of effort. In the good user interface design, user could do the job without noticing software. To make better usability, different factors could be considered such as typography and graphic design. These could affect on when the user is doing certain interaction and could make better or worse the potential of the user to do certain interaction. With the aim of having a usable system that could addapt to user needs, two factors such as psychology visual elements and technical functionality should be considered. As user interface design involve different platforms, disigners should be expertise and have enough skills in their platforms. Also, in user interface design, user needs should be understand well. Each user interface design could have several steps and processes that some of them may be related to each other [1][2].

In this paper, we are going to review some well-known user interface design rules be used by most companies.

## II. BEN SHNEIDERMAN UI RULES

In 1987, Ben Shneiderman, distinguished professor in university of maryland, introduced eight golden rules for designing productive and disappointment free user interface. Apple, Google and Microsoft are such as companies that used shneiderman UI rules for their successful products. Considering these eight rules while designing could help to have better design.[3]

### *A. Strive for consistency*

In order to design consistent interface, the same design patterns and the same sequences of actions should be used for similar situations. Using the familiar icons, menu hierarchy, right color, typography, terminology in prompt screens, commands could be included but not limited. In this way, users could do their goals more easy and they do not nead to learn new representation of the same action with consistent interface.[3]

### *B. Enable frequent users to use shortcuts*

It is needed for quicker methods when using the systems becomes more. Using UI rules as shortcuts will help them to get advantage of shortcuts. In particular, it could be more useful when it is needed to do the same job more often. Such shortcuts like Abbreviations, Function Keys, Hidden Commands, Macro Facilities could be useful for expert users. As an example, In Windows and Mac users could use keyboard shortcuts for copying and pasting, so as the user becomes more experienced, they can interact with the user interface more quickly and automatically as they get more experience.[3]

### *C. Offer informative feedback*

There should be some system feedback for every user action in reasonable amount of time. The feedback should be understood easily by human and the user should understand where they are and what is happening. For usuall and minor actions, the response can be selfconsious, while for unusuall and main actions, the response should be more significant. As a bad example when we often face with an error code instead of a human-readable and relevant message[3].

### *D. Design dialog to yield closure*

It should be used beginning, middle, and end to organize sequence of actions. Users should not guess their actions. They should know what their actions will led to them. By completing each group of actions some feeback should be sent to users. This could provide a satisfaction of accomplishment. The informative feedback at the completion of a group of actions gives the users a satisfaction of accomplishment with sense of relief and a signal to drop possibility plans. It also shows that the way for next group of actions is clear. For example, users will receive a “Thank You” message and a proof of purchase receipt when they’ve purchased online.[3]

### *E. Offer simple error handling*

Try to design a system with least user interface errors in order to avoiding the user for making a serious error.

Generally, no one likes to be told with errors. In case of error happening, the system should detect the error and offer simple, step by step uderstandable mechanisms for handling the error quickly and painlessly. For example, when the users forgot to provide input in an online form, bold the text fields.[3]

### *F. Permit easy reversal of actions*

The designers should offer users easy and obvious way to reverse their actions. This feature is about feeling of worry. As the user knows that errors can be undone, this encourages them seeking of unfamiliar options. Reversibility could be some single actions, a data entry, or a complete group of actions.[3]

### *G. Support internal locus of control*

Charging the system is strongly desirable for experienced users that they are initiator of actions and that the system responds to their actions in full control of events. Making users the initiators of actions rather than the responders should be considered in design systems.[3]

### *H. Reduce short-term memory load*

The human attention is limited and could keep a small items in short-term memory at a while. Information processing in short-term memory requires that interfaces be kept simple with proper information hierarchy. Multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes and sequences of actions.[3]

Specific client needs will dictate how to prioritize the usage of these rules. The client should never be taken out of the equation, and it might be the case that one or more of these rules results in superfluous and obtrusive design (as well as an additional development expense). Where simplicity is one of the main guiding principles in UI design, we should not make use of more guidelines than are necessary to successfully accomplish our task. This is, of course, not to discard the importance of these rules, but rather to reconsider their usage contextually; this will ensure that they are applied toward the success of a design, and not otherwise.

Specifically, let’s take as an example rule B. While enabling frequent users to use shortcuts may be helpful for some desktop applications, this would almost certainly not be the case on the application’s smartphone counterpart. Yet another example, this time considering rule 8. Advanced applications for power-users would need a vast number of features and therefore a more complex interface; although we would be adding to the short-term memory load, in this particular case, such a design decision might be appropriate.[3]

## III. NIELSEN NORMAN GROUP USER INTERFACE DESIGN RULES

Nielsen Norman group is one of the world leaders group in UI and UX. In 1995, they introduced ten usability heuristics for user interface design. It has ten principles for interaction design. They are called heuristics because they are not specific guidelines for usability and expectation, they are broad rules of thumb. Their heuristics are used in many companies products such as Adobe, Apple, and Google and they could improve usability, utility, and desirability of designs.[4][5]

### *A. Visibility of system status*

The users should always know about what it is going on the system by some feedback through the system inappropriate time. They should easily understand the system status on screen with a reasonable amount of time.[4][5]

### B. Match between system and real world

Designers should endeavor the users' language which used for speaking with the system. Concepts, words, and phrases should be easily understood by the user, rather than system-oriented terms. Making information should appear in a natural and logical order as a real-world convention. This will make systems easier to use.[4][5]

### C. User control and Freedom

Users should easily control their functions and have enough freedom to leave unorder state without obligating to through a complicated dialogue. Backward steps should be considered such as undo and redo as the users usually do mistake when working with systems.[4][5]

### D. Error prevention

It is better to design carefully to prevent errors in the first place rather than good error messages. Checking error-prone condition and remove or control them to present users with a confirmation option before they commit to the action. The potential errors should be kept minimum because users are not interested to detect and remedy problems. Also, flagging actions could be considered as a mean of error prevention.[4][5]

### E. Recognition rather than recall

Usually, recognition is easier for users than recall. The cognitive load should be at least by maintaining task-relevant information within the display while users explore the interface. It should be tried to make objects, actions, and options visible to minimize the user's memory load. The user should not have to memorize information from different parts to another. Using of the system should be visible or easily retrievable whenever appropriate this is because of limitation on human attention. We could only maintain around five items in our short-term memory at once. As a result, designers should be certain that users can simply employ recognize instead of recall. For example, doing a test on multiple choices is easier for us to answer questions on a test because it just needs to recognize the answer rather than recall it from our memory.[4][5]

### F. Flexibility and efficiency of use

As using system increase, it is demanded faster navigation and less interaction. For this aim, some issues such as abbreviation, hidden commands and function keys could be used. Accelerators that new users have not seen may help to speed up the interaction for the expert user such that the system can provide to both inexperienced and experienced users. It assists users to do frequent actions.[4][5]

### G. Aesthetic and minimalist design

Dialogues should include useful information and irrelevant or rarely needed information should not be used. Extra information in a dialogue participates with the relevant units of information and decreases their relative visibility. Unrelated information should be minimal because they will consume a user's limited attentional resources, which could inhibit a user's memory retrieval of relevant information. As a result, the display must include only the necessary components for the current tasks, whilst providing clearly visible and meaningful of navigating to other content.[4][5]

### *H. Help users recognize, diagnose, and recover from errors.*

Designers should assume users could not understand technical terminology, therefore, Error messages should almost always be expressed in plain language to ensure nothing gets lost in translation.[4][5]

### *I. Provide online documentation and help*

It could be ideal if the users could use the system without any helping or documentation. However, it may be needed to provide some help and documentation. The helping system should be easily searched, focused on the user's tasks, define steps for doing tasks and not to be confused and large.[4][5]

## IV. JOHNSON USER INTERFACE DESIGN

### *A. Focus on the users and their tasks, not on the technology*

When designing, it is critical to understand the users and their needs. The tasks that they want to do with the system should be considered. Another important issue is the context in which the software will function. It should not mostly focus on technology rather than the user.[6]

### *B. Conform to the user's view of the task*

In designing, try to strive for naturalness and use users' vocabulary not your own as a designer. Program internal must be concealed from the user view inside the program. Also, try the correct point on the power/complexity tradeoff and user should conform to view the task easily.[6]

### *C. Design for the common case*

Common results should be easily achieved by each user. Usually , there are two types of common, more often and less often. When designing, it is important to design for core cases and does not sweat edge cases.[6]

### *D. Do not complicate the user's task*

The user task should be simple and could be easily understood. Don't give users extra and sophisticated problems and do not make users reason by elimination.[6]

### *E. Facilitate learning*

In design, provide a low-risk environment that should be consistency. Provide an environment by designing that user could easily learn and interact less.[6]

### *F. Deliver information, not just data*

The design should display carefully and if needed get some professional help from others. The screen belongs to the user to provide information, not just data and preserve display inertia.[6]

### *G. Design for responsiveness*

Every task that the user do, acknowledge user actions instantly. It could be helpful to provide some way in order to let users know when software is busy and when it is not. While users are waiting for doing some part, free users to do other things. The movement should be animated smoothly and clearly. Users could be allowed to leave lengthy operations they don't want an estimate how much time operations will take. Letting users set their own workspace could be helpful.[6]

### *H. Try it out on users then fix it*

Test your design result on users. Sometimes it could surprise even experienced designers. The designer should schedule a time to correct problems found by tests. Usually, testing has two goals: informational and social. Every time and purpose could have different tests.[6]

## V. GAMIFICATION IN USER INTERFACE DESIGN

Gamification is the application of game-design elements and game principles in non-game contexts to improve user engagement, organizational productivity, flow, learning, crowdsourcing, employee recruitment and evaluation, ease of use, and more. The key here is to use elements of game design (fun, motivation, reward) to get users to do something that is in their benefit (and deepens your business goals). LinkedIn pioneered the UI pattern of profile completeness (and just-in-time tips and prompts) as a way to prompt users to share more information. This is now a default design pattern in web applications.[7][8][9]

Gamification, as a recent phenomenon for UI and UX designers , is a powerful approach for designing UI in order to increase user engagement and good experience. Firstly, you use this tool to enter fun elements in UI applications and systems. Users face challenges, whether challenging themselves or trying to win awards. Generally, they enjoy it. Secondly, the dynamics designers incorporate in successful gamification serve as effective intrinsic motivation, themselves – meaning users engage with the system because they want to. For instance, Foursquare/Swarm promotes users to “Mayors” of establishments after so many visits, enabling them to vie for a top place while enjoying meals, shopping, movies, etc.[7][8][9]

There are a variety of gamification techniques that can be considered in user interfaces, alongside other fundamental observations of what makes a gaming experience enjoyable & rewarding. By designing a product with these themes in mind, the aim is to create positive experiences that hook users to your product. Some of the gamification techniques for UI design are:

### *A.Badges / Medals*

Badges or Medals are a simple way of providing feedback to your user while bringing a sense of reward for completing a task. This reward creates motivation to continue and can also help offset any negative experiences during the task itself.[7][8][9]

### *B. Delightful interactions / instant gratification*

Adding small pieces of magic throughout your product makes the experience fun and enjoyable to interact with. The core functionality of the application must be reliable and usable before elements of delight can be added, as an interface that is unreliable or does not work as expected will create feelings of frustration and friction that overpower any touch of delight or fun.[7][8][9]

### *C. Fluid navigation / experience*

When navigation of video game user interfaces is done right, they bring fluid and smooth experience to the game, and when done perfectly — bring no friction to the players’ experience. Whether you need to switch items, update an option setting, or simply pause the game — when nothing disrupts the user the experience is seamless. Bringing this to product design translates to providing fluid navigation of your product, where transitions between pages & forms, for example, should give the user clear feedback & understanding of what is happening at all times.[7][8][9]

### *D. Leaderboards*

Depending on your product, leaderboards could be a feature to consider adding a sense of competition for your users, where you incentivize them to compete with each other, to increase engagement and fulfillment of completing tasks, etc.[7][8][9]

### *E. Progress bars*

As with badges & medals, progress bars, as the name implies, give the user a sense of progress and accomplishment whilst completing tasks.[7][8][9]

## VI. UI DESIGN ASSESSMENT BY USABILITY TESTING

Usability is a quality attribute that assesses how easy user interfaces are to use. On the Web, usability is a necessary condition for survival. If a website is difficult to use, people leave. If the homepage fails to clearly state what a company offers and what users can do on the site, people leave. The word "usability" also refers to methods for improving ease-of-use during the design process. Usability is defined by 5 quality components:

*Learnability:* How easy is it for users to accomplish basic tasks the first time they encounter the design?

*Efficiency:* Once users have learned the design, how quickly can they perform tasks?

*Memorability:* When users return to the design after a period of not using it, how easily can they reestablish proficiency?

*Errors:* How many errors do users make, how severe are these errors, and how easily can they recover from the errors?

*Satisfaction:* How pleasant is it to use the design?

There are many methods for studying usability, but the most basic and useful is user testing, which has 3 components: First, get holds of some representative users, such as customers for an e-commerce site or employees for an intranet (in the latter case, they should work outside your department). Second, ask the users to perform representative tasks with the design. Third, observe what the users do, where they succeed, and where they have difficulties with the user interface. Shut up and let the users do the talking.

It's important to test users individually and let them solve any problems on their own. If you help them or direct their attention to any particular part of the screen, you have contaminated the test results.[4][5]

## CONCLUSION

In this paper, we reviewed four well-known user interface design rules for application systems. Both Shneiderman and Nielsen start with a rule calling for consistency in design. Both lists have a rule about preventing errors. Shneiderman rule to “permit easy reversal of actions” is about the same in Nielsen rule to “help users recognize, diagnose, and recover from errors". "make users feel they are in control" is referring the same content with “user control and freedom”. Johnson mostly focuses on designing for users needs and tasks. Gamification is a recent approach that uses in most popular software user interfaces. It is barely on increasing user engagement on software by giving them scores or playing games. The main of all of these approaches is to design an user interface such that users could easily do their jobs and engage to applications. Some of them have similar methods that the reason could be just later authors were influenced by earlier ones. Every UI design could be evaluated by usability testing in order to understand how users could easily work with systems.

## REFERENCES

[1] https://en.wikipedia.org/wiki/User_interface_design

[2]hneiderman B (1999), "Jefferson's Laptop: User interfaces for universal creativity", Educom Review., May/Jun., 1999. Vol. 34, pp.34-36

[3] Scholtz J and Shneiderman B (1999), "Introduction to Special Issue on Usability Engineering", Empirical Software Engineering. Hingham, MA, USA, Mar., 1999. Vol. 4(1), pp. 5-10. Kluwer Academic Publishers.

[4] https://www.nngroup.com/

[5] M. Young, The Technical Writer's Handbook. Mill Valley, CA:University Science, 1989.

[6] Johnson, J. (2014). Designing with the mind in mind: simple guide to understanding user interface design guidelines. ISBN: 9780124079144, Morgan Kaufmann.

[7]https://tubikstudio.com/gamification-in-ux-increasing-user-engagement/

[8] K. Robson, K. Plangger, J. H. Kietzmann, I. McCarthy, and L. Pitt, "Is it all a game? Understanding the principles of gamification," Bus. Horiz.,

vol. 58, no. 4, pp. 411–420, 2015.

[9]https://www.interactiondesign.org/literature/topics/gamificationK.Elissa, "Title of paper if known," unpublished

# Author Index