Computer Science & Information Technology 93

Natarajan Meghanathan
Dhinaharan Nagamalai (Eds)

Computer Science & Information Technology

4th International Conference on Software Engineering (SOFE-2018)
September 29~30, 2018, Copenhagen, Denmark

AIRCC Publishing Corporation

Volume Editors

Natarajan Meghanathan,
Jackson State University, USA
E-mail: nmeghanathan@jsums.edu

Dhinaharan Nagamalai,
Wireilla Net Solutions, Australia
E-mail: dhinthia@yahoo.com

ISSN: 2231 - 5403
ISBN: 978-1-921987-91-5
DOI : 10.5121/csit.2018.81301 - 10.5121/csit.2018.81304

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

Preface

The 4th International Conference on Software Engineering (SOFE-2018) was held in Copenhagen, Denmark during September 29~30, 2018. The 4th International Conference on Advanced Computing (ADCOM-2018) and The 4th International Conference on Information Technology Convergence and Services (ITCSS-2018) was collocated with The 4th International Conference on Software Engineering (SOFE-2018). The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

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

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

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

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

Natarajan Meghanathan
Dhinaharan Nagamalai

Organization

General Chair

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

Program Committee Members

Abdellah Hadjadj	Avenue de l'Universite, France
Abdellatif Berkat	Abou-Bekr Belkadd University, Algeria
Abdelmonaime Lachkar	SMBA University Fez, Morocco
Abdessamad Belangour	Hassan II University, Morocco
Akram Abdelqader	AL-Zaytoonah University of Jordan, Jordan
Alaa Hamami	Princess Sumaya University for Technology, Jordan
Alex Afanasyev	Florida International University, USA
Ali Yazici	Atilim University, Turkey
Andy Rachman	Institut Teknologi Adhi Tama Surabaya, Indonesia
Ayad salhieh	Australian College, Kuwait
Ayman Sadig	Ahfad University for Women, Sudan
Baozhong Tian	University of Wisconsin , Wisconsin ,USA
Christian Mancas	Ovidius Univesrity, European Union
Denivaldo Lopes	Federal University of Maranhao, Brazil
Dipak Gade	IGATE Solutions Ltd, India
Dmitry A. Zaitsev	International Humanitarian University, Ukraine
Doru Florin Chiper	Technical University Gh. Asachi Iasi, Romania
Emad Awada	Applied Science University, Jordan
Enrique Herrera-Viedma	University of Granada, Spain
Ephzibah E.P	VIT University, Vellore, India
Fairouz Tchier	King Saud University, Saudi Arabia
Fatiha BOUBEKEUR	Mouloud Mammeri University of Tizi-Ouzou, Algeria
Franco Frattolillo	University of Sannio, Italy
Gabor Kiss	Obuda University, Hungary
Govardhan A	Jawaharlal Nehru Technological University, India
Hamid Alasadi	Basra University, Iraq
Hao Shi	Victoria University, Australia
Hao-En Chueh	Yuanpei University, Taiwan, R.O.C.
Hariharan S	J.J.College of Engineering, India
Hayet Mouss	Batna Univeristy, Algeria
Hazem El-Gendy	Ahram Canadian University, Egypt
Hazlina Haron	Universiti Utara Malaysia, Malaysia
Hemant G	Architecture Manager Intel, USA
Hongzhi	Harbin Institute of Technology, China
I-Ching Hsu	National Formosa University, Taiwan
Ivo Pierozzi Junior	Embrapa Agricultural Informatics, Brazil

Technically Sponsored by

Computer Science & Information Technology Community (CSITC)

Networks & Communications Community (NCC)

Soft Computing Community (SCC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

4th International Conference on Software Engineering (SOFE-2018)

4th International Conference on Advanced Computing (ADCOM-2018)

4th International Conference on Information Technology Convergence and Services (ITCSS-2018)

SOFTWARE ENGINEERING IN GLOBALLY DISTRIBUTED TEAMS

Manasés Jesús Galindo Bello

Department of Applied Computer Science, Hochschule Fulda, Germany

ABSTRACT

Software engineering principles are brought into practice by Information Technology companies all around the world. Software can be developed by local teams which members have different cultural backgrounds, as well as by teams distributed across countries. To save costs and be close to markets and customers, companies offshore or outsource the personnel. Although developing software in distributed teams offers multiple benefits, there are also stiff challenges that engineers and managers have to deal with, e.g. communication and collaboration may get affected because of geographic distance, different time zones and distinct cultural backgrounds among team members. If not addressed on time and effectively, these challenges generate misunderstanding and conflict among the team which eventually may impact the projects deadlines and quality of the software. This paper presents the most common software engineering practices, challenges and tools in global teams, as well as practical cases in the industrial and academic realms.

KEYWORDS

Collaboration, Management, Culture, Conflict, Challenges

1. INTRODUCTION

The Information Technology (IT) world is in constant change. Day after day more and more big companies as well as start-ups are relying on a geographically dispersed workforce to be able to quickly bring to the market innovative and enhanced products. Being agile to respond to the customer needs is key to succeed in the global economy nowadays. Therefore, companies build their teams in such a way that gathers the most skilled and qualified people from around the world and relying on the benefits of international diversity by bringing together high qualified personnel from different countries with diverse work experiences and perspectives.

Software development methodologies have their own well-known strengths and weaknesses [1], and over the years most of those methodologies have been improved. In software engineering, a software methodology refers [1] to the framework that is used to structure, plan and divide the software development work into distinct phases to improve design, product management and project management; and it is also known as software development life cycle. Each methodology has a different specification and it may include the creation of specific deliverables and artifacts to document, develop or maintain a system. One methodology that works for a project or a company, it is not necessarily suitable for another. The methodology shall be chosen based on different factors such as the nature of the project and technical, organizational and team considerations.

Natarajan Meghanathan et al. (Eds) : SOFE, ADCOM, ITCSS - 2018
pp. 01–16, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81301

In the latest years, companies have been evolving [11] from using traditional plan-driven software development to start experimenting with one or more agile frameworks, as well as opting to outsource or offshore the resources and collaborating with geographically dispersed teams. Diverse and numerous are the effects of physical distance [2] between software development teams. Proximity, in contrast to distance, promotes constant and face-to-face communication and the development of closer and more positive relations. In the same way, the continuous presence of colleagues improves feelings of familiarity and liking towards them, but on the contrary, physical distance avoids or significantly decreases affinity and fondness, both of which are oppositely connected to conflict [3]. The challenges by project managers leading globally distributed teams can turn to be complex and not easy to overcome. Collaborating successfully is not a simple task when every team member is local and even share the same working space; and when people come from different countries and cultural backgrounds, or when the team members are located in different time zones, communication can be inefficient and generate misunderstanding, cooperation can deteriorate, and at the end these factors create conflicts within the team members [4] and the project.

To communicate over physical distance, globally distributed teams have to rely heavily on computer and telecommunications technologies to mediate their interactions. Notwithstanding that such technologies enable remote team members to interact, it is practically impossible to transmit non-verbal language. Nevertheless, companies have migrated from using telephone conference bridges to video conferencing, which has earned its place amongst new online collaborative tools and it was recognized in 2016 among the top five collaborative trends [5].

One of the biggest challenges in globally distributed teams can be the communication between the team members. Their personal diversity of race, culture and backgrounds lead to some of these challenges that project managers encounter, and such diversity may create barriers that could not allow a sharp understanding of the objectives and tasks. English is commonly known as one of today’s global and most dominant languages, and it is primarily spoken by those for whom it is not a first language. In globally distributed teams, English is used as a communicative tool to interact within and across international borders by those who do not share a common first language [6]. Even though the primary communication language is the same, pronunciation, regionalisms, idiomatic expressions and/or lack of practice and fluency make it difficult to communicate clear ideas to other team members.

Conflict in work groups is unavoidable and may happen in all hierarchy levels, between directives and/or team colleagues. Work groups can opt to see the conflict as a negative factor and act to solve it once it is present, or as a positive factor and use it in favor [7] to achieve beneficial changes to the team and the company. Therefore, conflict becomes a part of life and it can occur in almost any area, and it is of high importance to learn to manage it efficiently and effectively to reduce or avoid the stress levels. In every work group of every organization, each team member has different personalities, education, background, goals and opinions. Because of this, conflict in the workplace becomes inevitable. It is a normal, and even healthy, part of business relationships; after all, two individuals cannot be expected to agree on everything at all times. When conflict is not addressed and properly managed, it can deteriorate the team members’ communication or a business relationship, but when it is addressed in an intelligent, respectful and positive way, it creates an opportunity for growth and ultimately strengthening the relation between two parts. Learning how to manage conflict efficiently is an important skill for anyone in management and the key to preventing it from hindering people’s professional growth; and needless to say, it is extremely important to handle it efficiently to ensure the success of a global project and the well-being of the globally distributed team members.

The rest of this paper is organized as follows: Section 2 presents the related work on global software engineering. Section 3 briefly describes what global software engineering is and the state-of-the-art methodologies and collaboration tools. Section 4 mentions the major challenges of global software engineering followed by the author's empirical knowledge (Section 5) and the challenges faced in the industrial and academic realms. The conclusion and further research is presented in Section 6.

2. RELATED WORK

Researching about software engineering and software development in a global context is not new, but it has gained importance over the last decade due to the capitalization of the talent pool and resource usage wherever needed [8], the advantages of the different agile development methodologies that have emerged [9-10] and many companies opting to adopt such agile frameworks [11-12] and relaying in a geographically dispersed workforce to quickly kickoff projects and save costs by outsourcing or offshoring [13, 16]. Numerous articles and books have been written based on empirical investigation to demonstrate the advantages and disadvantages of global software development and the challenges that globally distributed teams may face [14-17]. In many occasions, researches have explored the particular challenges associated with global software engineering bringing into practice the agile methodologies to the industry [17-18] as well as to the academic realm [19-20].

In his book, Carmel [14] provides original "prescriptions" for dealing with cultural differences and collaborative teams. Another approach that works for projects of any size is presented by Ebert and it is based on his first-hand experience and expertise [16]. The book offers a proven and balanced framework for planning global development and provides best practices for managing a variety of software projects, coordinating the activities of several locations across the globe while accounting for cultural differences.

Holmstrom et al. [18] identified through their empirical investigation the particular challenges of global software development at three different USA based companies. They presented the problems associated to geographical and socio-cultural distance and their proposals to overcome or at least reduce distance problems. They discovered that companies work hard to stimulate knowledge sharing between teams' members, but in the end, it comes down to the capacity and interest of one understanding each other. In a more recent research [11] the authors presented how companies have been evolving from traditional waterfall development to agile development, and from there making a transition towards continuous integration and continuous deployment in what they call the "stairway to heaven". Based on interviews at different companies with globally distributed development teams, they presented the challenges in the transition of development methodologies as well as the actions needed to overcome them.

Oktaba et al. [21] investigated how Latin American software companies have tried to improve their software processes' capability as a fundamental step towards increasing the quality of their products. They found out that many of these companies implement good practices, models and standards proposed by the Software Engineering Institute (SEI), the Capability Maturity Model Institute (CMMI) or the International Organization for Standardization (ISO). Nevertheless, these companies have faced troublesome situations when trying to collaborate with geographically dispersed teams. It is mentioned in their research that cultural differences play an important role in the success of software processes improvement, and that an organization would reject a process if it does not match its culture, in a similar way as the human body would reject a mismatched transplanted organ.

In the academic realm, since Fall 2004 the Fulda University of Applied Sciences (HSF) and the San Francisco State University (SFSU) have been teaching a distributed and collaborative software engineering class [19-20] for the students to get acquainted with local and global software engineering. After five years of analyzing the outcomes of the classes, it was found out that both, local and global teams, were able to produce similar high-quality work. However, the biggest challenges that both teams faced were cultural differences and lack of development experience, and additionally time zone difference and physical distance for the global teams.

3. GLOBAL SOFTWARE ENGINEERING

Companies involved in large-scale software development tend to engineer and release systems used by a high number of users with different needs. The role of product management is to document these needs and to combine, merge and prioritize them to be able to prepare a roadmap with a set of requirements for the next release of the system. Since the adoption of agile methodologies, each release of the system is usually done within two to four weeks; and when more requirements arise or new projects are being negotiated, it may become hectic and hard to keep the agile development pace. Finding qualified workforce locally, in the same country or within the same time zone, it is indeed a difficult task and –in most of the cases– expensive. As an example, a software development company located in the USA would pay annually about $94,000 USD to a middle level software developer located also in the USA, whereas hiring a remote software developer in India would cost about $14,000 USD annually [22]. Besides salaries, companies also save costs by outsourcing the resources as they do not have to provide neither medical insurance nor an ergonomic work space to remote employees (also known as vendors or contractors).

Global software development is [18] when the distribution of the members of a distributed software development team exceeds the frontiers of a country. It is characterized by having distributed teams involved in IT projects and consisting of stakeholders from different cultures, nationalities, mother languages, geographic locations and probably different time zones too. Companies leverage skilled workforces in lower-cost economies thanks to high-speed Internet-based communication. India and China are well known [23] for offering workforces at greatly reduced cost compared with employment markets in the USA and western Europe. Other countries that have become skilled in software development in the last decade [24] and have come to compete in the market are Brazil, Ukraine, Belarus, Russia, Pakistan, Malaysia and Vietnam.

In the last decade, companies have opted to migrate from plan-driven software development approaches to agile methods because they are more flexible [25] when it comes to taking into consideration changes in the requirements in all phases of software development. Besides the methodologies, collaboration tools have been improved significantly and they help global teams in a daily basis. Version control, instant messaging and project management are some of the most common collaboration tools of global software development.

3.1. Common Methodologies

The waterfall methodology [1, 26] was the first established modern approach for building a system. It was originally defined by Winston W. Royce in 1970 and it quickly gained support from project managers because everything flows logically from the beginning through the end of a project. Since then, engineers, developers and researches improved and redefined the methodology to define the different plan-driven or traditional software engineering methods –which follow a linear framework type– and have been widely used in both large and small software engineering projects [26-27]. Notwithstanding the success of the waterfall methods with large and complex systems, it has disadvantages such as inflexibility in the face of changing requirements, punctilious processes irrespectively of the nature and size of the project, and the creation of a big amount of documentation [27] which could take more time than coding the system itself.

The waterfall approach provides a framework with five distinct stages [26]: requirements analysis and definition, system and software design, implementation and unit testing, integration and system testing, operation and maintenance. As it is a linear framework, a stage shall not start until the previous one has finished and the results have been validated and approved. Agile methodologies [10, 25] were developed to address the drawbacks presented in the traditional and linear frameworks. Agile methodologies deal with unstable and volatile requirements by using simple planning, short iterations, earlier releases and frequent customer feedback. These characteristics enable agile methods to deliver product releases in a shorter period of time compared to the waterfall approach. In contrast to traditional and linear frameworks, agile methodologies provide frameworks to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system (retrospective). Figure 1 presents a comparison of the stages of waterfall and agile life cycles.

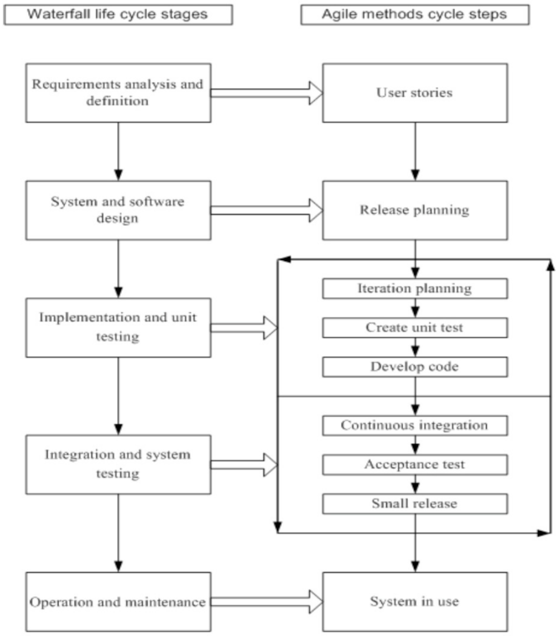
Waterfall life cycle stages
Agile methods cycle steps
Requirements analysis and definition
User stories
System and software design
Release planning
Implementation and unit testing
Iteration planning
Create unit test
Develop code
Integration and system testing
Continuous integration
Acceptance test
Small release
Operation and maintenance
System in use

Figure 1. Waterfall methodology vs Agile methodologies [25]

The most common agile methodologies are Scrum and Kanban [10]. They are widely used for well-known international companies all over the world, being the USA the country with the highest number of projects using Scrum [28]. Nevertheless, in the past five years Kanban has become more popular for the extra benefits and flexibility offered over Scrum [29-30].

Another framework that has become popular and quickly spread all the way through the technical community is DevOps, this because it understands the value of collaboration between development and operations staff through all stages of the software life cycle when creating and operating a service. One definition of DevOps is [31] the practice of development (Dev) and operations (Ops) engineers participating together in the entire service life cycle, from design through the development process to production support. DevOps has strong affinities with agile methodologies. The old view of operations tended towards the Dev-side being the “makers” and the Ops-side being the “people that deal with the creation after its birth” [31]. In this way, DevOps can be interpreted as an outgrowth of agile software development, which prescribes close collaboration of customers, product management and developers to rapidly iterate towards a better product. Furthermore, DevOps states that service delivery and how the application and systems interact are a fundamental part of the value proposition to the client too, therefore, the product team needs to include those concerns as a top-level item. From this perspective, DevOps [31] extends agile principles beyond the boundaries of "the code" to the entire delivered service.

3.2. Common Collaboration Tools

In order to succeed in a globally distributed software development project, teams need to have effective collaboration, which is one of the most common challenges of global software engineering and it is described in the next section. To achieve this collaboration, companies need to rely with full trust on different tools, and needless to say, on a high-speed internet connection. These collaboration tools can be grouped in six categories mentioned below, as well as the major service providers of each one [5, 31-33]. It is out of the scope of this research to mention the technical characteristics and benefits of each tool; therefore, each one has been just discretionary hyperlinked.

- Communication. Screen-sharing and instant messaging applications (chats), private social networks and wikis (internal to the companies), telephone and video conferencing, email applications and virtual boards are the tools that comprise this category, and the most common are Yammer, Rocket.Chat, IBM Sametime, Slack, Skype, Google Hangouts, GoToMeeting, WebEx, Zoom, IBM Verse, Outlook.

- Design. When team members need to prototype, brainstorm, create mockups and use virtual boards with sticky notes, the tools that can help on these tasks are Invision, Mural, Zeplin and MindMeister.

- Documentation. Tools designed to allow teams editing files at the same time and save their changes automatically. In this group Quip, Google Docs, Office Online and Zoho Projects are well known.

- File Sharing. When it comes to safe storing and sharing any type of files, Google Drive, Dropbox, Box, and OneDrive are the leading tools.

- Project Management. One of the most important tools while it comes to collaboration, and they can be used for all the members of a team. These tools have the ability to assign and prioritize tasks and help to identify what is critical versus what can wait. These tools keep teams up-to-date by informing the group of milestones and due dates for individual and team goals. Besides that, good management tools have the ability to provide quick feedback and score or measure performance. The leading tools are Asana, Jira Software, Trello, Basecamp, Nimble and Zoho Projects.

- Version Control. A source code repository is where developers can host their projects (either publicly or privately) to collaborate with other developers. At the time of this paper, GitHub is the most popular software repository on the web [34] followed by Bitbucket, SourceForge and GitLab.

Cloud-based applications have become popular and widely adopted because they excel in real-time team collaboration. However, when it comes to privacy and data security, a company may think twice before adopting a new technology or opt to implement it within its own infrastructure. In such cases, some service providers offer an enterprise version of their software for the companies to deploy and manage it in their own secure environment in order to keep their data private. Some of these providers are GitHub Enterprise, Trello Enterprise, Atlassian Enterprise, Slack Enterprise Grid and Box Enterprise. Often companies make public their paths on adopting a technology, e.g. IBM adopting Slack [35] even though the big company has its own cloud-based communication tool (IBM Same time). Sharing the tech stack that a company uses is beneficial to stay up-to-date with the top technologies in the industry and to discover new tools and services. With this information, developers can get to know which skills companies are looking for when searching for a job offer.

4. MAJOR CHALLENGES OF GLOBAL SOFTWARE ENGINEERING

Real life global software engineering projects have been widely analyzed and it has been pointed out the main problems that affect such environments, especially related to communication. Global collaboration creates significant coordination costs, e.g. incompatible schedules can lead to project delays, intercultural or linguistic misunderstandings can create non-billable rework, and technology failures can cause missed deadlines [14]. Working at a distance is complicated because it affects both the way one feels and thinks. It can lead to group the work colleagues into categories rather than see them as individuals. It also has bad consequences because one tends to view those with the same or similar culture/nation more positively, and those from a different culture/nation more negatively. Having an English-based video conference with a colleague in a different country, who speaks with an accent and exhibits different culture, makes that person seem a world apart [3]. If that was not bad enough, working at a distance also limits the amount of information we acquire about our colleagues and this can have negative effects for an effective collaboration. Besides this, other major challenges are overcoming the cultural differences, project management and the process of negotiation and decision-making.

4.1. Cultural Difference and Effective Collaboration

Cultural attributes can always be noticed in groups of people that share the same learned values, habits and behaviors; and these play a vital role in how one person performs different tasks at work based on individual patterns of thinking, feeling and acting [38]. Attributes that can be easily noticed by the naked eye are clothing, religious rituals, traditions, architecture or sports; while invisible attributes comprise of orientations to environment, time, communication, personal space, power, individualism, competitiveness, structure and mental frame. Cultural dimensions [38] are identified factors of great influence on the expected success of IT teams. These aspects must be contemplated by companies outsourcing or offshoring projects, as well as for local projects developed by teams with people coming from different countries, social backgrounds and perspectives.

When working in the same office with the colleagues it is possible to notice, interpret, incorporate and leverage an important amount of information as trying to make sense of one's daily experiences. Information that might not seem relevant (e.g. personal lives, moods or even the weather) also plays an important role. Having a bad mood due to weather conditions or being sleep-deprived, does affect the way one faces the daily activities. Unfortunately, barriers in the form of distance, time, culture, language and technology, all stand in the way of communicating such information, creating what Cramton [36] defined as The Mutual Knowledge Problem. A large percentage of the information is unknown when interacting with distant colleagues, and the less information is known about them, the more they are seen as "them". These outcomes are the exact opposite of a positive teamwork, but they happen so naturally that one may not usually identify such problems at the moment, especially when the colleagues are focused exclusively on results without regard for interpersonal relations. The problem with "us and them" thinking is focusing on the differences over the similarities among the colleagues, but one way to overcome this is to reverse the focus. Highlighting the things one has in common with the distant colleagues is the best way to reduce the problem [3].

Effective global collaborations have processes, procedures and good practices designed in a way that enforces information sharing across sites. Some of these are [37] scheduling regular meetings to share task-related information, taking time to share personal updates, equipping the team with the right resources and giving and taking a virtual tour to provide context. The more distant teammates know about each other's environment, the better to be able to understand one another's behavior.

4.2. Project Management

Functional, cross-functional and self-managing teams are the three different types of teams that can be found within a company [39], each one has its own specific goals and objectives, and they are created for both long term and short term interaction. This makes project management a more challenging discipline when it comes to initiate, plan, control and close the work of a globally distributed team. Project managers make project goals their own and use their skills and expertise to inspire a sense of shared purpose within the project team.

The Project Management Body of Knowledge [40] describes the core areas involved in the right management of a project and clearly identifies communication, human resources and stakeholders' management as core knowledge areas, but the dimension of intercultural teams plays an important role on the potential failure or success of project management for global teams. Cultural difference in project management is a stiff challenge also for local intercultural teams. These teams often face the problem of coping with local habits, cultural dimension conflicts, cross-cultural communications challenges, cultural shock and human resources issues. An intercultural project manager who works abroad, with people located in other countries and with local colleagues coming from different countries, has to take these cultural differences in religion, language, traditions, ethics, background, etc., into consideration and avoid adopting an ethnocentric perspective in order to succeed on the role as leader of the team.

4.3. Negotiation and Decision-Making

Since negotiation is strategic in nature, it necessarily represents a set of overlapping goals. For professional business negotiators [41], the job is to secure a settlement within their bargaining range and to trade interests wisely. Breakdowns in negotiations, when parties are from different cultures, many times are attributed to cultural differences. Furthermore, if culture has an effect on negotiation, the mental models of negotiators from one culture may not map on to the mental models of negotiators from another culture [41], making the specification of a single mental model problematic.

Project managers of global engineering projects many times have to assume the role of negotiators when making business and/or establishing relationships with other project managers in different companies and countries. Nevertheless, just because project managers can be from different cultures, it does not necessarily mean that their negotiation strategies will clash and their agreements will be suboptimal. Distinct social groups may have similar cultural values, and members may find the intercultural negotiation process trouble-free. In addition, not all members of a cultural group with a different profile believe and act consistently and strictly within the cultural norms [41]. When parties are motivated to reach an agreement, much can go on during the course of a negotiation to overcome individual, contextual and cultural differences. Three key factors leading to successful agreements that are affected by culture are [41] value for information sharing, means of searching for information and motivation to search for information. For long term positive outcomes, the goal should be to encourage a win-win situation in order to bring a set of agreements that are clearly acceptable and beneficial to all concerned parties. It is worth mentioning, the process of negotiation is difficult per se when it takes place among people of similar socioeconomic and cultural backgrounds, and it becomes even more complex when it takes place across countries and different cultures. Before starting any negotiation or during the decision-making process, it is very important to have in mind the assumption that individuals on the other side of the road do not necessarily think, act and feel the same way one does. Project managers who have been successful in the past and have worked with people from different cultures and nations, have a much better sense of the often hidden significance of the agenda based on the cultural values of their counterparts.

5. A POSTERIORI KNOWLEDGE

It is discussed in this section what the author has seen and experienced about software engineering by working for international companies, as well as during his academic career, and being involved in global software engineering projects.

5.1. Software Engineering vs. Software Development

Although the terms might seem similar and interrelated, they are not quite the same thing. The IEEE defines[1] Software Engineering as the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing and documentation of software. Based on this complete definition, it is possible to infer that software engineers are experienced computer scientists involved in the whole software development life cycle. Nevertheless, programmers, software developers and testers are not necessarily engineers, rather experts whom possess a particular set of skills in programming languages, frameworks or technologies, and they are highly reliable during the implementation and testing phases of a software development project. In the other hand, software engineers and software architects have broader skills that allow them to orchestrate the whole software engineering process.

Nowadays it is quite common to see in the news how software companies generate millions of revenues, and in particular those companies developing mobile applications and games. For these and other reasons, many people have decided to pursue a career as software developers; and there are some interesting cases of switching career paths in arts and music or aeronautics to become developers. There are plenty of options for those who want to pursue a career as programmers or software developers. Many courses are offered online on in-site by recognized universities and educational institutions related to programming languages, databases, apps development et al. In Mexico and other countries in Latin America, it is offered a technological high school program in Informatics with the focus to highly prepare the schoolers as programmers being able to work as soon as they graduate, but the salaries for such technicians in informatics are quite low due to the fact that companies value more experience and educational level. However, not all people are fond of studying and, therefore, they choose the technical career path knowing that their skills as well as salaries may improve over the years. In the other hand, to become a software engineer, the first step is to study engineering in a bachelor's or master's level. During the studies, one must necessarily be involved in software development projects; for that reason, in many university programs it is mandatory to make a professional residence or internship in a company, and it is precisely while being hands-on in a real project when students get the necessary practical knowledge and skills for what they are studying for.

5.2. Experience in the Industrial Realm

I have had the opportunity to work in several companies and being part of globally distributed and intercultural teams in Mexico. After graduating from B.Sc. in Computer Engineering, I got

[1] The Bureau of Labor Statistics - Systems and software engineering - Vocabulary, ISO/IEC/IEEE std 24765:2010(E), 2010.

the opportunity to start an intense training organized by two international companies: T-Systems and Volkswagen. The aim was to train 20 software engineers and take the certification exams from Solaris and Oracle. It was my first time in an international environment and I got to see how projects can be started in a short notice as well as getting cancelled. It was precisely what happened there, as we were notified that the project we had been trained and certified for was cancelled, but there was the opportunity to enroll to other training for a new project.

After that, I decided to check other options and I found opened positions with a Brazilian consultancy firm for a huge project within HSBC Retail Banking and Wealth Management. My professional experience was only the one I had from my internships and the training at T-Systems and Volkswagen. After personal, English language and technical interviews, this firm hired me and sold my services to the bank as a Senior specialist. There I got to meet many other people from other firms as it is a common practice for the bank to outsource the services at the beginning of a project. During this time, I faced the conflicts and challenges with cultural distance and effective collaboration as the teams were multicultural, locally and globally. I collaborated with people from China, Brazil, Argentina, Paraguay, Chile, USA, Canada, France, India and England. One big issue was time difference because often meetings were hold at 03:00 (Central Standard Time) at the offices of HSBC Mexico in order to coordinate with the teammates in China and India, and it was necessary to come back to a normal working day again at 09:00 in the morning.

A couple of years later, another consultancy came into the picture and offered a project within IBM in the USA. Time difference was one hour, so I decided to take the offer. I was the only one in Mexico joining that project whose managers and technical leaders were located in the USA, and it was allowed to all teammates to work from home in a regular basis. During this time, I got to collaborate in many projects as well as with many people located in China, India, Brazil, USA, Canada, Philippines, Denmark, Egypt, Slovakia, England and France. Something positive from IBM is that all the time new methodologies and technologies to improve the collaboration are implemented, and the career path is quite wide. As in my case, I worked in backend and frontend development before becoming the product owner and application architect of a project. Working by IBM helped me to get more familiar with intercultural teams. Once a month there was a team meeting where two team members randomly chosen would take five to ten minutes to share something about their countries and cultures. It was a very bold action from the managers as it helped the teammates to get to know each other better and reduce the cultural distance between them.

Both companies have well defined coding standards, good development practices and defined project management methodologies are followed. They are an example of the companies that have migrated from traditional to agile methodologies. Some of the tools used by HSBC [42] and IBM [43] for effective collaboration are precisely those mentioned in Section 3.2.

5.3. Experience in the Academic Realm

During my studies of Computer Engineering I had several courses of programming languages, databases, software development and software engineering. Most of the theory, methodologies and best practices were learnt there, but it was until the time working in the industry when everything was put fully into practice. Unfortunately, at the time of this paper, universities in Mexico do not offer collaborative courses with other universities in a bachelor's level.

The first time I faced global collaboration in an academic field was while studying a M.Sc. in Global Software Development at the Fulda University of Applied Sciences in Germany. As briefly mentioned in Section 2, since Fall 2004 the HSF and the SFSU have been teaching a distributed and collaborative software engineering class for the students to get acquainted with local and global software development. The goal of the class is not only to polish the students' hard skills, i.e. programming and architecting a software, but also to cover the soft skills such as leadership and effective teamwork which are key to succeed in software development projects.

At the beginning of the Winter semester, Professor Todtenhoefer (HSF) informed during the first class that everybody would work in teams for a Software Engineering project and collaborating remotely on the same project with other team from the SFSU, but only one group would work in a local project. The teams were made randomly by the professor based on the skills and expertise of each student. I was selected to work in a global project with other three students from Pakistan and they chose me as the team leader. Both teams, HSF and SFSU, had four members and each one with different development responsibilities, i.e. front-end or back-end, besides the roles of team lead and tech lead. The team from SFSU was more multicultural as the members were from Germany (team lead), Czech Republic (tech lead), China and the USA. I was very confident at the beginning and thought that we would not face or practically avoid all issues due to everybody in my team had more than three years of experience in the industry and also in global teams. We joined the SFSU team about two months after the project was started, this because in the USA the study programs have different schedules. We discovered that there was no organization whatsoever and the collaboration tools were not set up. As professionals, we asked to have a kick-off meeting but the SFSU team refused by saying that it was not necessary and email communication was enough. After two weeks, the goals and tasks for HSF team were not clear. Our professor played the role of Project Manager and Negotiator. We did bring all the details to him (escalating the issue), so he could talk with the other professor playing the role of Project Manager of SFSU team. It was until this point when SFSU team started to cooperate and we had the first video call and we communicated only with the two leads using Skype. We were very limited by the professors on the tools we could use, so everything was kept in minutes of the meetings using emails and all the tasks were specified on the minutes. A very rudimentary process, but we put a lot of effort for bringing to both teams harmony and organization. The teams had weekly meetings with their respective Project Managers to check progress and the planned tasks for next week, and also to talk about strategies and actions in order to improve and have a more harmonic, happy and effective collaboration. During the last month of developing a web application using software engineering principles and effective collaboration, we managed to create a peaceful and harmonic working environment, even though the project management techniques were rudimentary. Both professors graded very high our developed software and team collaboration, so the outcomes turned to be copacetic.

5.4. Conflicts and Challenges

Conflict is like water: too much causes damage to people and buildings; too little creates a dry, barren landscape denuded of life and color. It is a fact that one needs water to survive as well as an appropriate level of conflict to develop and grow one's character [4]. The way to manage natural resources of water through dams and reservoirs is key to achieve the balance necessary for life, and likewise with conflict management; a balance must be stablished between opposing forces and competing interests. In order to well manage conflict, members of globally distributed teams must develop and polish a broad set of soft skills, i.e. leadership, poise, tolerance, proactivity, intercultural knowledge, organization, customer relationship and management. In particular, project managers have to be able to use this set of skills to create an environment in which the project reaches its immediate goals and also functions successfully in terms of meeting the expectations, feelings and needs of its team members.

It is known that sleep deprivation affects people's mood, quality of life and level of happiness. Project managers know –or at least they should– that happy employees are more productive [7]. Base on these facts, companies and managers should be more flexible when working in distributed teams and implementing practices that would benefit everybody in the team, e.g. scheduling meeting in reasonable times, letting the employees work from home if they had to be working overnight and provide good equipment for remote work and state-of-the-art collaboration tools. Something extremely useful to overcome the challenges and cope with a multicultural team is a training in intercultural communication. Unfortunately, this is something that is not practiced in small software development companies and only some big international corporations have considered giving such trainings to their employees. Being the world so into globalization and software engineering projects being outsourced and offshored, it is of high importance for the companies to consider giving the employees interpersonal and intercultural trainings for everyone to be ready to face, understand and successfully collaborate with other people with different backgrounds and way of thinking.

5.5. The Dark Side of Global Software Engineering

Working remotely creates distance with other teammates that could probably never be reduced. Project managers may become more task-focused than people-focused and caring more about weekly or monthly results and performance rather than the well-being of the team. As already mentioned, happy employees are more productive, but this is something that global managers do not consider in many occasions and specially in organizations that are shaped like a pyramid.

Another negative side of global and intercultural collaboration is the constant stereotyping. A team member can be stereotyped or be less accepted based on the country he or she is coming from. This creates a distance even in local teams. Team members may search for those who share the same or a similar culture when needing help related to the project or just to have someone to talk to, but when this is not found, they may also become task-oriented employees avoiding talking with other team members and eventually having a not happy working environment.

And last but not least, the bad practices that consultancy firms still do while hiring with low salaries recently graduated engineers or developers with less than one year of experience, and selling their services in a Senior level to big corporations abroad; and as the projects are being outsourced, such corporations primarily care about results. When it comes to offshoring, also big corporations have been pointed out for their deliberately malicious business practices. Such is the case of IBM [44, 45] making millions of dollars by selling services to governments in different countries and delivering software that does not work because it was offshored in India with low skilled specialists [45].

6. CONCLUSION AND FUTURE WORK

Bruce Lee once said *"Empty your mind. Be formless, shapeless, like water. When you put water into a cup, it becomes the cup. You put it into a bottle, it becomes the bottle. You put it into a teapot, it becomes the teapot. Water can flow or it can crash. Be water, my friend"*. Running water is never steady, it continues flowing; when it reaches an obstacle, it continues pushing until it can flow again. In a similar way, one will always face conflicts and challenges in both personal and professional life, and it is important to face them and move on by being flexible and open-minded to be able to understand that every head is a different world and people will not always think in the same way one does. There are no easy answers or a standard and unique way on how a manager from Canada, Mexico, Brazil, Germany or Australia should manage conflicts that are inevitable while working in global teams or in negotiations across cultures. Understanding the cultural differences and gathering the maximum information about the culture of the countries, helps to understand what type of people are in the team, how to deal with them and what are their expected reactions or behaviors. Collaboration in globally distributed teams is and always will be a difficult thing to do, and it is important to be aware that issues and conflicts may arise. There are a wide range of useful skills for handling conflicts, and one very important is assertiveness. An individual needs to be able to express his/hers views clearly and firmly, but without aggression. It is important to emphasize that dealing with conflict in an early stage is generally easier because positions are not so entrenched and others are less likely to have started to take sides. The best way to address a conflict in its early stages is through negotiation between the participants.

Bachelors and Masters programs in universities may prepare good software engineers, but still it is until one gets involved in real life projects when those important soft skills are well developed. Learning about Conflict Resolution and Emotional Intelligence helps to deal in a more professional way with the multiple issues that one may face while working in an intercultural and/or globally distributed team.

The information presented in this paper is based mainly on experience and informal interviews carried out with former colleagues as well as students and professors of software engineering classes at universities in Mexico, Germany and the USA. It was observed that students as well as professionals have faced similar conflicts when collaborating with distributed teams and they did not have a proper preparation in social and cultural aspects that could have been useful to cope with those kind of conflicts. It is therefore, as plan of the future work, to conduct a case study adopting a qualitative research approach to analyze how software engineers deal with such conflicts when collaborating with global teams and which techniques or soft skills would be useful in their daily work. Furthermore, it is the plan to invite more universities offering software engineering programs to organize and collaborate with other universities, and especially those located in non-English speaking countries and not highly developed like Germany or the USA.

REFERENCES

[1] Paul Fisher, James McDaniel and Peter Hughes. 2008. System Development Life Cycle Models and Methodologies. Canadian Society for International Health Certificate Course in Health Information Systems, Module 3: System Analysis & Database Development, Part 3: Life Cycle Models and Methodologies.

[2] Sara Kiesler and Jonathon N. Cummings. 2002. What do we know about proximity and distance in work groups? In P. J. Hinds & S. Kiesler (Eds.). Distributed work. Cambridge. MA: MIT Press. 77-81.

[3] Mark Mortensen and Pamela J. Hinds, (2001). Conflict And Shared Identity In Geographically Distributed Teams. International Journal of Conflict Management. Vol. 12, Issue 3 (2001), 212-238. DOI: https://doi.org/10.1108/eb022856

[4] Craig E. Runde and Tim A. Flanagan. 2008. Building Conflict Competent Teams (1st. ed.). John Wiley & Sons, San Francisco, CA, USA.

[5] Rob Marvin. 2015. 5 Collaboration Trends to Expect in 2016. PC Mag Middle East. (October 2015). Retrieved March 7, 2018 from http://me.pcmag.com/your-business/4249/feature/5-collaboration-trends-to-expect-in-2016

[6] Alan Firth. 1995. The Discourse of Negotiation: Studies of Language in the Workplace (1st. ed.). Pergamon, UK.

[7] Cathy A. Costantino and Christina S. Merchant. 1995. Designing Conflict Management Systems: A Guide to Creating Productive Healthy Organizations (1st. ed.). Jossey-Bass, San Francisco, CA, USA.

[8] Andrew Begel and Nachiappan Nagappan. 2008. Global Software Development: Who Does It? In Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on Global Software Engineering. DOI: https://doi.org/10.1109/ICGSE.2008.17

[9] Samireh Jalali and Claes Wohlin. 2011. Global software engineering and agile practices: a systematic review. Journal of Software: Evolution and Process. Vol. 24, Issue 6 (August 2011), 643-659. DOI: https://doi.org/10.1002/smr.561

[10] Smartsheet. 2018. What's the Difference? Agile vs Scrum vs Waterfall vs Kanban. (February 2018). Retrieved March 8, 2018 from https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban

[11] Helena Holmstrom, Hiva Alahyari and Jan Bosch. 2012. Climbing the "Stairway to Heaven"—a Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software. Proc. 38th EUROMICRO Conf. Software Eng. and Advanced Applications (SEAA 12). 2012, 392–399. DOI: https://doi.org/10.1109/SEAA.2012.54

[12] Jan Bosch. 2016. Speed, Data and Ecosystems: The Future of Software Engineering. IEEE Software, Vol. 33, Issue 1 (2016), 82–88.

[13] Hannah Whittenly. 2016. Ways Your Business Can Save Money by Outsourcing. (December 2016). Retrieved March 10, 2018 from http://www.leadershipgirl.com/5-ways-business-save-money-outsourcing

[14] Erran Carmel. 2011. Global Software Teams: Colloborating Across Borders and Time Zones (1st. ed.). Prentice Hall.

[15] Jutta Eckstein. 2010. Agile Software Development with Distributed Teams: Staying Agile in a Global World (1st. ed.). Dorset House Publishing.

[16] Christof Ebert. 2011. Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing (1st. ed.). Wiley-IEEE Computer Society Press, Hoboken, New Jersey, USA.

[17] Helena Holmstrom, Anna Sandberg, Jan Bosch and Hiva Alahyari. 2014. Scale and Responsiveness in Large-Scale Software Development. IEEE Software. Vol. 31, Issue 5 (October 2014), 87-93. DOI: https://doi.org/10.1109/MS.2013.139

[18] Helena Holmstrom, Eoin O. Conchuir, Par J. Agerfalk and Brian Fitzgerald. 2006. Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. International Conference on Global Software Engineering (ICGSE2006), Costão do Santinho, Florianópolis, Brazil, 2006. DOI: https://doi.org/10.1109/ICGSE.2006.261210

[19] Shihong Huang, Dragutin Petkovic, Gary D. Thompson and Rainer Todtenhoefer. 2010. Teaching and Assessment for Global and Collaborative Software Engineering Course.

[20] Gary D. Thompson, Dragutin Petkovic and Shihong Huang. 2009. Teaching Distributed Collaborative Development Techniques in a Software Engineering Class Setting.

[21] Hanna Oktaba, Félix García, Mario Piattini, Francisco Ruiz, Francisco J. Pino and Claudia Alquicira. 2007. Software Process Improvement: The Competisoft Project. Computer, IEEE Computer Society. Vol. 40, Issue 10 (October 2007), 21-28. DOI: https://doi.org/10.1109/MC.2007.361

[22] Visually, Inc. 2012. Salaries of web developers in India, the Philippines, USA and around the world. Retrieved March 10, 2018 from https://visual.ly/community/infographic/business/salaries-web-developers-india-philippines-usa-and-around-world

[23] Human Engineers. 2015. China and India: Comparative HR Advantages. Retrieved March 10, 2018 from http://www.humanengineers.com/hr_library/general/china-and-india-comparative-hr-advantages

[24] Pär J. Ågerfalk, Brian Fitzgerald, Helena Holmström and Eoin Ó Conchúir. 2008. Benefits of Global Software Development: The Known and Unknown. In Q. Wang, D. Pfahl, and D.M. Raffo (Eds.): Making Globally Distributed Software a Success Story, ICSP 2008, LNCS 5007, 1-9. Springer-Verlag Berlin Heidelberg.

[25] Ming Huo, June Verner, Liming Zhu, Muhammad Ali Babar. 2004. Software Quality and Agile Methods. Proceedings of the 28th Annual International Computer Software and Applications Conference. IEEE Xplore. (October 2004). DOI: https://doi.org/10.1109/CMPSAC.2004.1342889

[26] Ian Sommerville. 2015. Software Engineering (10th. ed.). Pearson, Harlow, England.

[27] Mukesh Jain. 2008. Delivering Successful Projects With TSP and Six Sigma: A Practical Guide to Implementing Team Software Process (1st. ed.). Auerbach Publications, Boca Raton, FL, USA.

[28] Scrum Alliance. 2013. The State of Scrum: Benchmarks and Guidelines. Retrieved March 17, 2018 from https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2013-State-of-Scrum-Report_062713_final.pdf

[29] Kanbanize. 2016. Kanban VS Scrum. Retrieved March 17, 2018 from https://kanbanize.com/blog/kanban-vs-scrum-infographic

[30] Kanbanize. 2018. Top Reasons Why Companies Consider Using Kanban. Retrieved March 17, 2018 from https://kanbanize.com/blog/why-use-kanban-infographic

[31] Jennifer Davis and Katherine Daniels. 2016. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale (1st. ed.). O'Reilly, UK.

[32] Nikoletta Bika. 2017. 14 collaboration tools for productive teams. Retrieved March 17, 2018 from https://resources.workable.com/tutorial/collaboration-tools

[33] Jagina McIntyre. 2017. Agile Collaboration: Tools, Techniques & Games. Study.com. Retrieved March 17, 2018 from https://study.com/academy/lesson/agile-collaboration-tools-techniques-games.html

[34] Wikipedia. 2018. Comparison of source code hosting facilities. Retrieved March 17, 2018 from https://en.wikipedia.org/wiki/Comparison_of_source_code_hosting_facilities

[35] Bill Higgins. 2017. Listen to the wild ducks: How IBM adopted Slack. Retrieved March 18, 2018 from https://medium.com/design-ibm/listen-to-the-wild-ducks-how-ibm-adopted-slack-2bcfd3732680

[36] Catherine D. Cramton. 2001. The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration. Organization Science. Vol. 12, Issue 3 (June 2001), 346 - 371. DOI: https://doi.org/10.1287/orsc.12.3.346.10098

[37] Heidi K. Gardner. 2017. Smart Collaboration: How Professionals and Their Firms Succeed by Breaking Down Silos (1st. ed.). Harvard Business Review Press, USA.

[38] Germinal Isern. 2014. Intercultural Communication and Management Factors and Their Impact to the Process of Global Software Development for Virtual and Non-Virtual Teams. Journal of Intercultural Management. Vol. 6, Issue 1 (January 2014), 5–16. DOI: https://doi.org/10.2478/joim-2014-0001.

[39] Susan M. Heathfield. 2016. How to Build Powerfully Successful Work Teams. Retrieved March 18, 2018 from https://www.thebalance.com/how-to-build-powerfully-successful-work-teams-1918510

[40] Project Management Institute. 2013. A guide to the Project Management Body of Knowledge (PMBOK guide) (5th. ed.). Project Management Institute.

[41] Michele J. Gelfand and Jeanne M. Brett. 2004. The Handbook of Negotiation and Culture (1st. ed.). Stanford Business Books, Stanford, USA.

[42] FeaturedCustomers. Business Software used by HSBC Bank. Retrieved March 19, 2018 from https://www.featuredcustomers.com/customer/hsbc-bank/reviews

[43] IBM. 2018. IBM Whitewater tools and services. Retrieved March 19, 2018 from https://status.whitewater.ibm.com

[44] Tristan Yates. 2005. How IBM Conned My Execs Out Of Millions. Retrieved August 04, 2018 from http://atdt.freeshell.org/k5/story_2005_9_27_95759_4240.html

[45] Alec Kinnear. 2018. A billion reasons never to buy IBM services. Retrieved March 22, 2018 from https://foliovision.com/2018/03/why-not-buy-ibm

AUTHOR

Manasés Jesús Galindo Bello completed a M.Sc. in Global Software Development (Germany) and a Master in Mobile Business (Spain) in addition to a B.Sc. in Computer Engineering (Mexico). He has broad experience in software engineering, has led IT projects at HSBC, HP and IBM and trained software engineers in different areas and companies.

USING CONCEPT ALGEBRA FOR MAPPING SOFTWARE PRACTICES TO ESSENCE FRAMEWORK

Murat Pasa Uysal

Department of Management Information Systems,
Baskent University, Ankara, Turkey

ABSTRACT

As a relatively new framework suggested for core problems of software development, one important issue for Essence Framework (EF) is mapping software development practices to the EF's conceptual domain. There are several works describing systematic procedures, however, a review of literature cannot suggest a study using formal method(s). In this paper, a software practice mapping method is proposed, which adopts and employs Concept Algebra principles in a Scrum case. The results are promising, however, more empirical evidences are needed to support the solution.

KEYWORDS

Software Engineering Practice, Essence Framework, Concept Algebra

1. INTRODUCTION

The Essence Framework (EF) is proposed for addressing the core problems of software development (SD) and its application [1]. Existence of plenty of development methods, which are: (a) hard to compare, (b) lacking of sound experimental method evaluations and/or validations; and (c) the increase of gap between practical application and academic research would be some of these problems. EF Kernel and Language Specification describes its key features and how it supports practitioners and method engineers. A set of elements for forming a common ground and describing a software engineering (SE) endeavour is defined as the kernel. Therefore, EF allows "people to describe the essentials of their existing and future methods and practices so that they can be compared, evaluated, tailored and re-used by practitioners as well as academics and researchers [2]".

By applying the principle of separation of concerns, and separating the "what" of SD from the "how", EF provides a common base and enables method building with the composition of various practices. Thus, a practice is defined as "a repeatable approach to doing something with a specific objective in mind [2]". It includes the necessary elements that exist in every software endeavour, such as, team work, requirements analysis/specification, development, test etc. Therefore, a method is built by the composition of a set of practices and using Kernel specifications.

EF includes a layered architecture with three discrete areas of concern. Each focuses on core and specific aspects of SE practices: (a) Customer, (b) Solution, and (c) Endeavour areas as depicted by Figure 1. In fact, the much of focus is given on the SD and practice use for compositing SD methods. The Alpha(s) of EF and the agile approach adopted enable capturing the key SE

Natarajan Meghanathan et al. (Eds) : SOFE, ADCOM, ITCSS - 2018
pp. 17–23, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81302

concepts. On this common ground, they allow monitoring the health and progress of SE endeavours and their associated artefacts. One of the key features of EF is that it allows a project team to assemble the methods according to their needs and experiences by the composition of various practices. However, an important issue has been how to map a SE practice to EF knowledge domain.

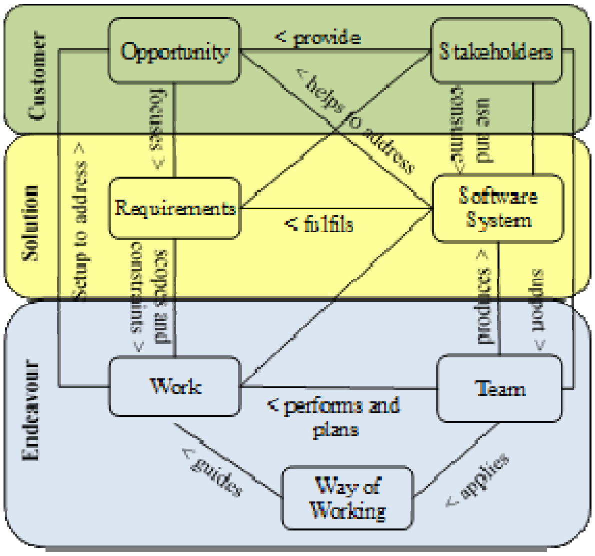

Figure 1. Essence Framework [1].

There are several works describing or proposing systematic translation of SE practices to EF-based descriptions. Essence Specification Document [2] includes several practice definitions, however, it has only a limited number and it is mainly for descriptive purposes. Park et al. base their mapping procedures on activity spaces, and thus, they propose an activity-state mapping algorithm, and present it in an Essence-powered Scrum practice [3]. Both Park [4] and Giray et al. [5] proposes an ontology-based systematic method for mapping SD to the EF. It is also explained how method engineering can help resolve some of the mapping issues [5]. In another study, Genetic Algorithms are introduced to generate candidate Essence Kernel replacements based on empirical data rather than human experience and judgement [6]. However, a review of literature on EF cannot suggest a formal method that guides mapping a SE practice to Essence-based definitions [7].

In this study, therefore, we propose a formal method for mapping SE practices to EF based on Concept Algebra definitions [8]. The next parts include theoretical foundations, sample case and conclusion sections of the paper respectively.

2. THEORETICAL FOUNDATIONS

2.1. Concept Algebra

Modelling is a kind of knowledge representation, and thus, conceptual mapping and semantic evaluations usually require formal methods. Since mapping the SD concepts of any SE practices to the EF concepts cannot be straightforward, thus, core concepts from Essence are initially extracted, and then, the mapping is conducted based on the formal definitions of Concept Algebra (CA) [8]. This algebra is "an abstract mathematical structure for the formal treatment of concepts and their algebraic relations, operations, and associative rules for composing complex concepts

[8]". It mainly provides denotational mathematics principles for algebraic manipulations of concepts.

A concept is defined as "a cognitive unit to identify and/or model a real-world concrete entity or a perceived-world abstract subject [8]". Accordingly, a concept connotes attributes or properties, and it denotes members or instances. Compositional and relational operations are the two main operations of CA. Thus, problems of various knowledge domains, such as, software and system engineering, can be identified, manipulated and modelled by using CA operations. In this study, the relational operations are used for comparing and mapping the corresponding abstract concepts of a SE practice to the semantic context of EF (Θ).

Given that Θ is a semantic context, the main conceptual definitions are as follows:

$$\Theta = (O, A, R) \tag{1}$$

Where, the symbol O denotes a finite/infinite nonempty set of objects, A is a finite/infinite nonempty set of attributes, and R is a set of relations between O and A. The general structured model of an abstract concept is illustrated in Figure 2.

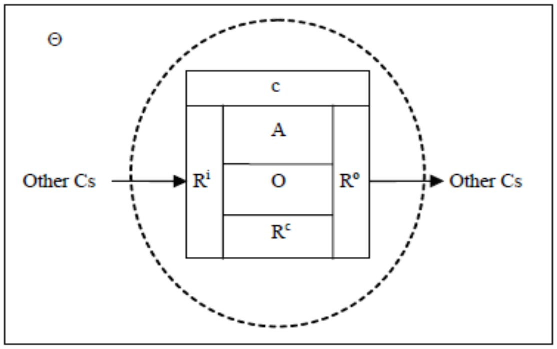

Figure 2. The structured model for an abstract concept [7]

An abstract Essence concept is regarded as the composition of different elements. Thus, an EF concept, with its attributes and objects, internal and external relations, can be defined as follows:

$$C_{EF} = (O_{EF}, A_{EF}, R_{EF}{}^{c}, R_{EF}{}^{i}, R_{EF}{}^{o}) \tag{2}$$

Where,

- C_{EF} is a concept in Essence,
- O_{EF} is a non-empty set of objects extended from this Essence concept, $O_{EF} = \{o_1, o_2, \ldots, o_m\}$,
- A_{EF} is a non-empty set of attributes of EF objects, $A_{EF} = \{a_1, a_2, \ldots, a_n\}$,
- $R_{EF}{}^{c} = O_{EF} \times A_{EF}$ is a set of internal relations of the Essence concept,
- $R_{EF}{}^{i} \subseteq C' \times C_{EF}$ is a set of input relations of the Essence concept and where C' is a set of external concepts,
- $R_{EF}{}^{o} \subseteq C_{EF} \times C'$ is a set of output relations.

A corresponding abstract SE Practice (SEP) concept, C_{SEP}, can be defined by adopting the same approach:

$$C_{SEP} = (O_{SEP}, A_{SEP}, R_{SEP}{}^{c}, R_{SEP}{}^{i}, R_{SEP}{}^{o}) \tag{3}$$

The relational operations in CA are defined as "related", "independent", "sub-concept", "super-concept", "equivalent", "consistent", "comparison", and "definition"; and they are represented by the $\{\leftrightarrow, \nleftrightarrow, \prec, \succ, =, \cong, \sim, \triangleq\}$ symbols respectively. Thus, the relationships between two concepts in the knowledge domains of EF and SEP are determined by the relations of their set of attributes A and the set of objects O. As being a dynamic mathematical structure, it is important to note that an abstract concept can adapt and interrelate itself to other concepts via input relations R^i and output relations R^o. In this study, these are $R_{SEP}{}^{i}$-$R_{SEP}{}^{o}$ and $R_{EF}{}^{i}$-$R_{EF}{}^{o}$ respectively.

2.2. Definitions

Take the concept c_1 from EF Θ and the concept c_2 from a SEP Θ. Suppose that they have the sets of attributes ($A_{1,}$ A_2) and the sets of objects (O_1, O_2). The following definitions are used when finding the similarity of two concepts in SEP and EF:

Definition 1: See whether the related concepts c_1 and c_2 share some common attributes in A_1 and A_2, which are denoted by:

$$c_1 \nleftrightarrow c_2 \Rightarrow A_{1,} \cap A_2 \neq \varnothing \tag{4}$$

Definition 2: Compare c_1 and c_2 and determine their equivalency or similarity levels as below:

$$c_1 \sim c_2 \Rightarrow \frac{\#(A1 \cap A2)}{\#(A1 \cup A2)} * 100\% \tag{5}$$

Where # means the cardinal operator giving the number of elements in a given set, and thus, 0% means no similarity whereas 100% means a full similarity.

Definition 3: Assume the equivalent concepts as follows:

$$c_1 = c_2 \Rightarrow (A_1 = A_2) \wedge (O_1 = O_2) \tag{6}$$

Which means that these two concepts have similar attributes ($A_1 = A_2$) and their instances are identical ($O_1 = O_2$).

3. SAMPLE CASE

One of the well-known practices is the illustration of how Scrum [10, 11] can be modelled in the Essence Kernel and Language Specification [2]. In this document, "Product Backlog" concept of Scrum is associated with the "Requirements" alpha concept of EF without specifying conceptual details. Note that a comprehensive comparison of concepts exists in Scrum, and mapping them to EF is beyond the scope of this paper. However, it is thought that even in a simple and clear case, such as "Requirements" and "Product Backlog", it is possible to miss or neglect some important conceptual details. Therefore, the below section shows how the formal mapping is applied:

- The theoretical background of mapping is based on Concept Algebra principles and definitions.

- A content analysis for the EF specification document and resources related to Scrum Practice [2, 7, 8, 9] is conducted.

- An attribute comparison list is created, which includes two sets of core attributes for the "Requirements" concept and "Product Backlog" concept (Table 1).

- Note that a concept in linguistics is assumed as a noun or noun-phrase, which serves as the subject of a to-be statement [8]. By using a Linguistic Typological Analysis (LTA) (assuming that a simple sentence is made of "subject", "predicate" and "object" parts), an initial similarity level is determined on a scale ranging from 0 to 3.

- "0" level indicates no-typological similarity where none of the parts of two attributes is similar. "1" indicates that one similar part exists. "2" means that two of linguistic parts are similar. Finally, 3 points out a full linguistic similarity where both of the sentences have similar "subject", "predicate" and "object" parts. Note that the level 2 or 3 is regarded as satisfactory for EF mapping procedures in this study.

- By using the definition (4), $A_{\mathrm{EF},} \cap A_{\mathrm{SEP}} \neq \varnothing$, we find that two concepts share some common attributes (a_3-b_3; a_4-b_4; a_6-b_6). LTA also shows that these three attributes have a linguistic similarity level at 2 (Table 1)

Table 1. Requirements and Product Backlog Attribute Sets

Set of attributes for "Requirements" concept of EF	**Set of attributes for "Product Backlog" concept of Scrum**	**Linguistic similarity level (0 to 3)**
$A_{EF} = \{a_1, a_2, \ldots, a_n\}$	$A_{SEP} = \{b_1, b_2, \ldots, b_n\}$	
a_1 = are the definition of what needs to be achieved	b_1 = is a prioritized list of desired product functionality	1
a_2= must address opportunity and satisfy stakeholders	b_2 = is required to meet the product owner's vision	1
a_3 = mechanisms for managing /accepting requirements need to be established	b_3 = product owner is responsible for determining and managing requirements	2
a_4 = progress through six states: conceived, bounded, coherent, acceptable, addressed, fulfilled	b_4 = the definition of ready and the definition of done are two major states of product backlog items (PBIs)	2
a_5 = must be bounded as a whole and stay within the bounds of original concept	b_5 = provides shared understanding of (a) what to build and (b) the order of what to build.	1
a_6 = continue to evolve as more is learned.	b_6 = Grooming is important and it refers to creating, refining, estimating and prioritizing PBIs continually.	2

- By using the definition (5):

$$\frac{\#(\{a1,a2,a3,a4,a5,a6,\} \cap \{b1,b2,b3,b4,b5,b6\})}{\#(\{a1,a2,a3,a4,a5,a6\} \cup \{b1,b2,b3,b4,b5,b6\})} * 100$$

The conceptual similarity level is consequently found as

$$= \frac{3}{9} * 100 = 33.3$$

This finding indicates a result, which may be regarded as different from the specifications or Essence-based Scrum practice definitions mentioned in the Essence Literature. Such that the "Requirements" and "Product Backlog" concepts are not conceptually equal as it is claimed or specified.

At first glance, the most of experts on both Scrum and EF may not object to association of "Requirements" and "Product Backlog". However, the result is substantially different in our sample case. It is thought that the primary reason would be the human experience and informal judgement, which is usually adopted in mapping procedures in the literature. For example, in [4] and [5], an ontology of terms, commitments and metamodeling techniques guide the mapping processes. However, their classifications of SE practice terms into a list of corresponding EF concepts, such as, work products, activities, roles, which again employ subjective expert judgements. In another study proposing an algorithm [4], the assignment of SE practice activities to EF activity spaces, specifying their alpha states and checklists are also dependent on personal experience and subjective expert evaluations.

Concepts are important for carrying certain meanings in thinking, reasoning and system modelling [8]. By using CA, SE practices and EF can be modelled as dynamic and abstract mathematical structures that encapsulate objects as well as their attributes and relations. This study shows that CA can provide the formal and generic knowledge manipulation means required for complex software and knowledge structures.

4. CONCLUSIONS

As a relatively new framework proposed for the core problems of SE methods, one important issue for EF has been the mapping a SE practice to the EF's conceptual domain. Thus, the main argument of this paper is that formal methods can provide more accurate transformations as well as they can enable application of more systematic mapping procedures. In this study, therefore, a formal method based on CA definitions is proposed as a solution. This is applied in a simple Scrum case and the results interestingly differ from the ones exist in the EF literature. However, the research limitations confine us within presenting only theoretical foundations, a generic case and initial observations. More empirical evidences are also needed to support the proposed method. Therefore, the paper concludes with an invitation to future studies aiming to address these research limitations.

REFERENCES

[1] Park S., Jacobson I., Myburgh B., Johnson P. and McMahon P.E. 2014. SEMAT yesterday, today and tomorrow, SEMAT, retrieved from http://semat.org.

[2] OMG 2016. SMSC/15-12-02 Essence–Kernel and Language for Software Engineering Methods, Specification v.1.1.

[3] Park J.S., McMahon P.E. and Myburgh B. 2016. Scrum powered by Essence. ACM SIGSOFT Software Engineering Notes. 1-8, 41(1).

[4] Park J.S. 2015. Essence- Powered Scrum: A generic approach to describing practices using essence kernel and language, retrieved from http://old.semat.org/wp-content/uploads/ 2015/03/ 1-Essence-Powered-Scrum-June.pdf,

[5] Giray G., Tüzün E., Tekinerdoğan B. and Macit Y. 2016. Systematic approach for mapping software development methods to the Essence Framework. In Proceedings of 5th International Workshop on Theory-Oriented Software Engineering (Austin, TX, USA, May 16 2016).

[6] Sedano T., P´eraire C. and Lohn J. 2015. Towards generating Essence Kernels using Genetic Algorithms. In Proceedings of International Conference on Soft Computing and Software Engineering (Berkeley, California, USA, March 5-6, 2015).

[7] Uysal M.P. and Giray G 2017. An Essence Framework approach to software engineering research. In Proceedings of 11th. National Software Engineering Symposium. (Alanya, Antalya, Turkey, September 18-20, 2017).

[8] Wang Y. 2008. On Concept Algebra: A denotational mathematical structure for knowledge and software modeling. Int. Journal of Cognitive Informatics and Natural Intelligence, 2(2), 1-19.

[9] Rubin K.S. 2013. Essential Scrum: A practical guide to the most popular agile process. Addison Wesley, NY, USA.

[10] Schwaber K. and Sutherland J. 2017. The Scrum Guide: The definitive guide to Scrum: The rules of the game. Creative Commons, USA.

[11] Sutherland J. 2014. The art of doing twice the work in half the time. Crown Business, NY, USA.

AUTHORS

Assoc. Prof. Dr. Murat Pasa Uysal is at the Department of Management Information Systems in Baskent University. He holds a B.S degree in electrical & electronic engineering, a M.S degree in computer engineering, and a Ph.D. degree in educational technology. He earned the Assoc. Prof. Dr. degree on Computer Education and Instructional Technologies taking the Turkish Inter-University Council two-phased qualification exams. He completed his post-doctoral studies at Rochester Institute of Technology in New York, which was on software re-engineering, e-learning and Information Technologies (IT) Governance. He served as an advisor and engineer for different types of IT projects in Turkish Army (TA) for many years, and conducted studies addressing the problems of TA in the research areas of IT. He has been teaching IT, information systems and software engineering related courses. His research interest is also in the areas of software engineering, information systems (IS), instructional methods and tools for computer programming and IS.

INTENTIONAL BLANK

REDCLAN - RELATIVE DENSITY BASED CLUSTERING AND ANOMALY DETECTION

Diptarka Saha[1], Debanjana Banerjee[2], Bodhisattwa Prasad Majumder[3]

[1,2]WalmartLabs, Bengaluru, Karnataka, India
[3]Dept. of Computer Science and Engineering,
University of California, San Diego, USA

ABSTRACT

Cluster analysis and Anomaly Detection are the primary methods for database mining. However, most of the data in today's world, generated from multifarious sources, don't adhere to the assumption of single or even known distribution - hence the problem of finding clusters in the data becomes arduous as clusters are of widely differing sizes, densities and shapes, along with the presence of noise and outliers. Thus, we propose a relative-KNN-kernel density-based clustering algorithm. The un-clustered (noise) points are further classified as anomaly or non-anomaly using a weighted rank-based anomaly detection method. This method works particularly well when the clusters are of varying variability and shape, in these cases our algorithm not only finds the "dense" clusters that other clustering algorithms find, it also finds low-density clusters that these approaches fail to identify. This more accurate clustering in turn helps reduce the noise points and makes the anomaly detection more accurate.

KEYWORDS

Clustering, Relative KNN – kernel density, Varying density clusters, Anomaly Detection, DBSCAN

1. INTRODUCTION

In the industry today, data categorization could be the single most important problem - categorize people according to income, categorize customers according to purchase patterns, categorize items according to price and the list goes on. The underlying data for categorization could have any form whatsoever – structured, unstructured, labelled, unlabelled, adhering to assumptions or, not.

In a scenario where the data points are unlabelled, the purpose of categorization would be simply to study the underlying pattern of the data and problems such as these are typically tackled under the realm of *clustering* [1]. In its core, clustering is an exploratory tool for data analysis and has been used in several fields such as statistics, document retrieval, image segmentation [2], biological sciences [3], psychology [4] etc. for some time now. In this paper, we are going to study a novel clustering technology that outperforms existing methods particularly well when the intrinsic clusters are of varying variability, shape& size. This more accurate clustering in turn helps reduce the noise points and makes the anomaly detection more accurate.

Natarajan Meghanathan et al. (Eds) : SOFE, ADCOM, ITCSS - 2018
pp. 25–39, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81303

2. LITERATURE SURVEY

Cluster analysis is a challenging task and despite being studied by numerous authors for decades it still has some documented issues associated with it. This is primarily due to the vagueness of the success criterion of the problem – clustering attempts at separating unlabelled data into *meaningful* groups with minimal input from user; however, what defines meaningful is dependent on the application at hand. In the most general of meanings the objects are clustered or grouped based on the principle of *maximizing the intraclass similarity and minimizing the interclass similarity*[5]. The diverse application potential along with the heterogeneity of the data associated with these applications have given birth to several clustering techniques over the years. Below, we present a brief high-level review of these techniques.

Traditionally clustering techniques are roughly divided into *hierarchical* and *partitioning*[6] methods.

Hierarchical clustering defines a *dendrogram* representing nested groups of objects which represent a hierarchical relationship of these objects [7]. One can either start with a singleton and recursively merge clusters as deemed appropriate: this is known as *agglomerative* (bottom-up) technique. Reversely, one may start by putting all the points into one cluster and recursively split each cluster into daughter clusters until a stopping criterion is achieved, such methods are known as *divisive* (top-down). Example of hierarchical clusters are BIRCH[8], ROCK[9] etc. These algorithms are especially useful when data might have some inherent hierarchical structure and/or features in the data are of varying type. However, they typically tend to be of higher time complexity, also they suffer from the vagueness of the stopping criterion.

On the other hand, partitioning methods cluster the data directly and not in gradual steps. One kind of partitioning method typically attempt at iteratively finding certain centres of data as representative of a cluster and build cluster around those centres. K-means [10] and K-medoid [11] are renowned cluster techniques that use this method. Although these methods are more computationally efficient, they perform poorly when clusters are non-convex and/or data has outliers. The number of clusters is also user – defined which calls for domain knowledge.

Another kind of partitioning method is popularly known as *density-based clustering,* herehigh-density data spaces are identified and points in those spaces are put in the same cluster. DBSCAN is the most well-known density-based clustering technique[12]. OPTICS[13] is another such algorithm which attempts at alleviating shortcomings of DBSCAN. One major characteristic of these methods is the number of clusters to be found is not user – defined and hence it allows for more flexibility. Also, these algorithms typically don't cluster all the points and may leave some points as noise points which may be treated as potential anomalies. Density based algorithms have proven to be improvement over other partitioning methods when data may have arbitrary shape and/or outliers; however, have historically failed to captured clusters of varying density – this drawback is precisely what we are hoping to eliminate while retaining all the aforementioned advantages.

Coming to Anomaly detection, it deals with the problem of finding points in data that do not conform to *expected* behaviour [14].As in the case of clustering defining what is *expected* or *normal* is the most difficult job and is typically application dependent. Most basic approaches [15][16] use the method of flagging off the most extreme points, points which typically fall beyond a certain threshold, these thresholds are mostly higher quantiles. Even though, the approach might be non-parametric, it fails to look at any more than what the data has to suggest at its surface. Another parametric alternative, Minimum Volume Ellipsoid estimation (MVE)[17] fits the smallest permissible ellipsoid volume around the majority of the data which represents *densely populated normal region*. Among other outlier detection techniques, *density-based anomaly detection* techniques such as Local Outlier Factor (LOF)[18] are also very popular due to their intuitive approache and interpretability. A comprehensive survey on outlier detection methods can be found in [19].

We however, instead of looking for anomalies overall, look for anomalies with respect to every cluster. The motivation behind such approach is the understanding that a point lying further apart in the space may be an anomaly or, it could simply be part of a different lower-density cluster; whereas a point lying close to a densely populated cluster may not present itself as anomalous in an absolute sense; but in a relative sense, may be a potential outlier. Precisely due to this architecture, a relative density-based clustering approach such as REDCLAN outperforms traditional methods by identifying potential outliers more accurately and robustly.

3. Relative Density Based Clustering and Anomaly Detection

3.1. Motivation

An important property of many real-data sets is that their intrinsic cluster structure cannot be characterized by global density parameters [13]. Very different local densities may be needed to reveal clusters in different regions of the data space. For example, in Figure 1,it is not possible to detect the clustersC_1, C_2, C_3simultaneously using one global density parameter.

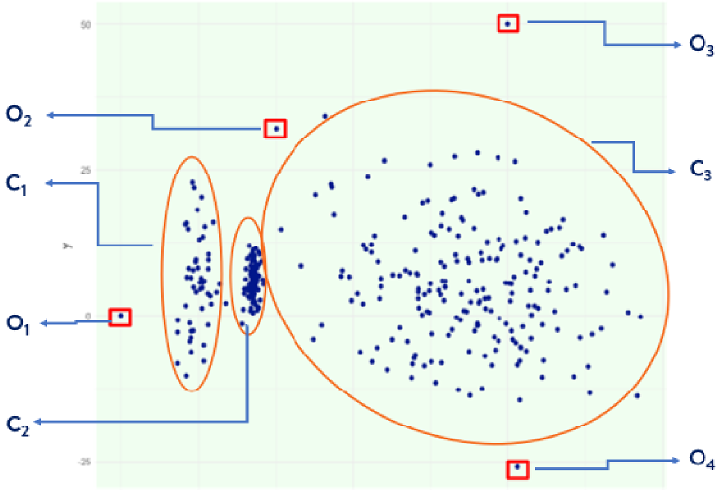

Figure 1. Data with Clusters of Varying Density

Here, we should note that this synthetic 2D dataset will be used several times for illustration purposes; this contains three Gaussian clusters of varying density $\{C_1, C_2, C_3\}$and 4 deliberately introduced outliers $\{O_1, O_2, O_3, O_4\}$

The aforementioned drawback of density-based clustering techniques (such as DBSCAN) can be understood as following: if the density of the high density cluster $C_2$is taken as the global density parameter then many a points in $\{C_1, C_3\}$will be seen as noise points – as they have (relatively) lower density, on the other hand if density of $C_3$is taken as the global density parameter then C_2 will be over-fragmented by algorithms such as DBSCAN [12], as is evident in the following Figure2.

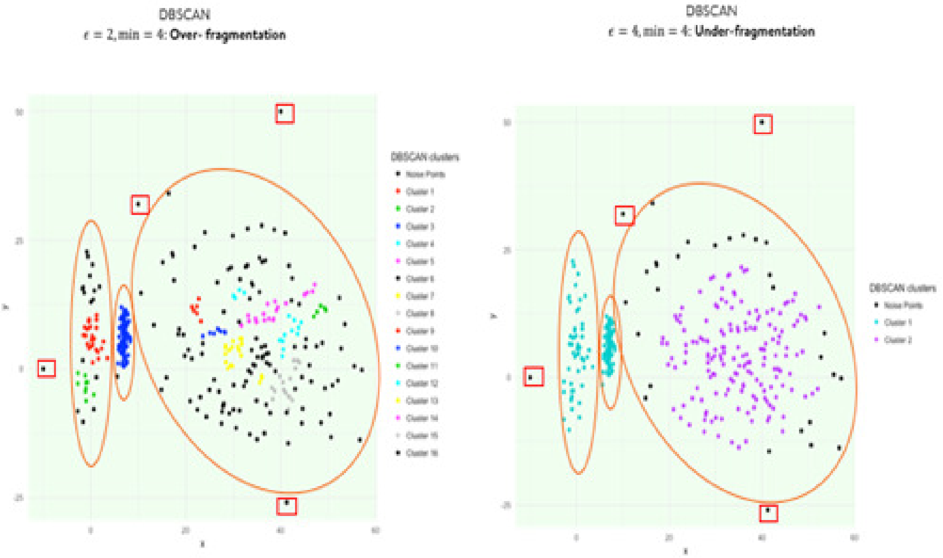

Figure 2. Data with Clusters of Varying Density: Use of DBSCAN

To overcome this problem, one needs to consider density relative to its neighbour, something called *relative density*, which will be formally defined later – this essentially means the density parameter for considering a set of points to be included in a cluster or to be left as noise points, will vary from point to point.

As we understand, after the clustering step we will have clusters and noise points; The natural next step for a data categorization algorithm such as ours would be to find outliers in the data; which we will accomplish by using a *weighted rank-based anomaly detection* technique. Since, the performance of outlier detection algorithms depends on how good the clustering algorithm captures the structure of cluster [20]– this algorithm provides significant improvement in datasets comprised of clusters of varying density.

3.2. Definitions

The following definitions will be used while describing the algorithm. First, consider the set of all d – dimensional points in the given data to be denoted by $\mathcal{D} = \{X_1, \dots, X_n\}$

For what follows, whenever we mention for a point p, it is understood$p \in \mathcal{D}$.Whenever distance of two points$\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$is discussed, we assume the natural Euclidean distance: $d(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$

- **kNN – neighbourhood:** If $d_k(p)$is the distance between p and its k^{th} nearest neighbour, then denote the set of k nearest neighbors of p by
$$\mathcal{N}_k(p) = \{q \in \mathcal{D} \setminus \{p\}: d(p,q) \leq d_k(p)\}$$

- **Adaptive Bandwidth:** Suppose, for a point p, we have its kNN neighborhood $\mathcal{N}_k(p)$, and given fixed $\epsilon > 0$
$$\Delta_k(p) = max\,(d(p,q): q \in \mathcal{N}_k(p))$$
$$\delta_k(p) = min\,(d(p,q): q \in \mathcal{N}_k(p))$$
$$\bar{\delta}_k(p) = \frac{1}{k} \sum_{q \in \mathcal{N}_k(p)} d(p,q)$$

  we form an *adaptive bandwidth* around the point p as following

$$h(p) \;=\; \Delta_k(p) \;+\delta_k(p) \;+\epsilon \;-\backslash\bar{\delta}_k(p)$$

- **kNN-based Relative Density:** Following definition 2, a balloon estimator [21] might be defined as

$$\rho(x) = \frac{1}{n\big(h(x)\big)^d}\sum_{1}^{n} K\left(\frac{(x - X_i)}{h(x)}\right)$$

  on top of this rather dynamic definition of density, we add another layer of local scaling and define our *relative density* as

$$\tilde{\rho}(x) = \frac{\rho(x)}{\frac{1}{k}\sum_{X_i \in \mathcal{N}_k(p)} \rho(x)}$$

- **Core Points:** A point will be denoted as *core point* if it has high relative density, i.e. for some threshold θ_1a point p will be denoted a core point iff

$$\tilde{\rho}(x) \geq \theta_1$$

  authors typically determine $\theta_1$using bootstrap on the entire set of relative densities
  The set of core points will be denoted by $\mathfrak{C}$

- **Directly Reachable:** A point p is said to be *directly reachable* from another point q iff

$$-\, q \in \mathfrak{C}$$
$$-\, p \in \mathcal{N}_k(q)$$

- **Reachable:** A point p is said to be *reachable* from another point q iff

$$\exists p_1, p_2, \dots\ , p_n with\, p_1 \;=\; q, p_n = p\ such\ that$$
$$p_{i+1}\ is\ directly\ reachable\ from\ p_i\ \forall i = 1,2, \dots, n-1$$

- **Connected:**A point p is said to be *connected* with another point q iff

$$\exists\ o \in \mathcal{D}\ such\ that\ bothp\ andq\ are\ reachable\ fromo$$

- **Rank:**The *rank* [22] of p w.r.t. q is defined as

$$rank_q(p) \;=\; |\{X_i \in \mathcal{D}\colon d(q, X_i) \leq\; d(p,q)\}|$$

  in informal terms, this is the order rank of p w.r.t. $q$quantified by the number of points between q and p plus 1

- **Outlierness:**Outlierness of a point is a function of the weighted sum of its rank w.r.t. its neighbour. Suppose $q \in \mathcal{N}_k(p)$. Define weights by

$$w(q) = \begin{cases} \dfrac{1}{|\mathcal{C}|}, & q \in \mathcal{C} \\ 1, & q\ is\ noise\ point \end{cases}$$

Essentially, if q is part of a cluster $\mathcal{C}$ define, $w(q) = \frac{1}{|\mathcal{C}|}$ if q is not part of any cluster (noise point) then $w(q) = 1$. This is to say every cluster has weight 1 which is equally divided among its components.

With this weightage scheme in hand *Outlierness* will be defined as

$$O(p) = \frac{\sum_{q \in \mathcal{N}_k(p)} w(q) rank_q(p)}{k}$$

3.3. Methodology

Next, we will discuss the algorithm in detail and why/how it works. It might be helpful to demonstrate the anatomy of the algorithm by using it on the synthetic dataset shown in Fig 1.

I. **Core Point Detection:** The very first step would be to find the set of core points $\mathfrak{C}$, this is done with the help of definition 4. Since, the core points are defined based on relative density and not absolute density – we can note (as in Fig 3), the core points will be spread across all the clusters – both dense and sparse ones, this forms the backbone of our algorithm.

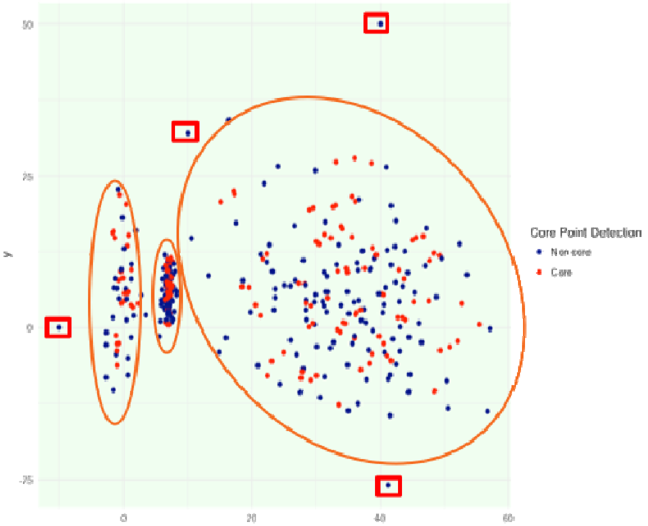

Figure 3. Core point detection

II. **Clustering:**Given the set of core points$\mathfrak{C}$, we will cluster the points into separate clusters. The clustering logic will be the following logic.

Define, $\Psi : \mathcal{D} \rightarrow \mathbb{N}$to be the clustering function which assigns a cluster number to every point in $\mathcal{D}$

Also denote,

$$n = |\mathcal{D}|$$

$\mathcal{A}(p)$: union of p and the set of points directly reachable from p

Initialize:

$$\Psi(p) \leftarrow 0 \; \forall p \in \mathcal{D}$$

$$C \leftarrow \max(\Psi)$$

```
while$p \in \mathcal{D}$do:
    if $\Psi(p) = 0$then
        if $p \in \mathfrak{C}$then
            $\Psi(\mathcal{A}(p)) \leftarrow C + 1 \;\forall p \in \mathcal{D}$
            $C \leftarrow \mathrm{C} + 1$
        end if
    end if
end while
```

At the end of this step(Fig 4) points are either put into a cluster or left as noise points. $\Psi(p)$ is the cluster number a point is assigned. If $\Psi(p) = 0$, this means the point is left as a noise point.

This is a one-time breadth-first process and depends on two input parameters, k_1, k_2.
$\boldsymbol{k_1}$**:** the k used in determining adaptive bandwidth in step I (core point detection), the higher the value of $k_1$the lower number of core points will be found, and more and more core points will be concentrated towards the denser cluster.

$\boldsymbol{k_2}$: the k used in determining the reachability of the points in step II, the higher the value of $k_2$fewer clusters will be found.

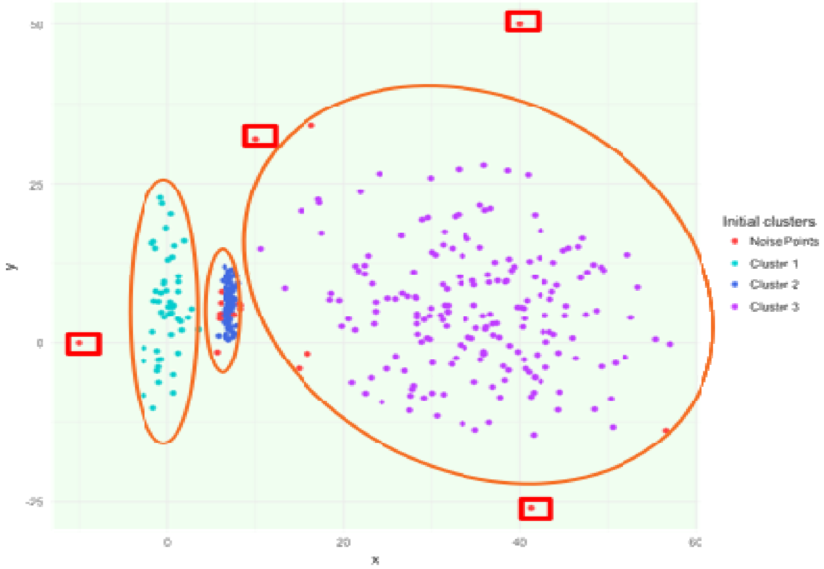

Figure 4. Clustering

III. **Anomaly Detection:** At the end of step II, we have clusters and noise points – these noise points maybe either anomalies or just boundary of given clusters, so we will call them potential outliers. Our algorithm goes the extra mile by finding outliers from the set of potential outliers using the weighted rank-based anomaly detection method.

Now, for a suitable threshold, θ_2 we do the following:

```
while$p \in \mathcal{D}$do:
        if $\Psi(p) = 0$then
                $O(p) = \frac{\sum_{q \in \mathcal{N}_k(p)} w(q) rank_q(p)}{k}$
                if$O(p) \geq \theta_2$then
                        $\Psi(p) \leftarrow -1$
                end if
        end if
end while
```

Note, authors have used $\boldsymbol{k} = \boldsymbol{k_2}$in determining neighbourhood while calculating Outlierness.
At the end of this step (Fig 5), If $\Psi(p) = -1$, this means the point is assigned an **outlier**status.

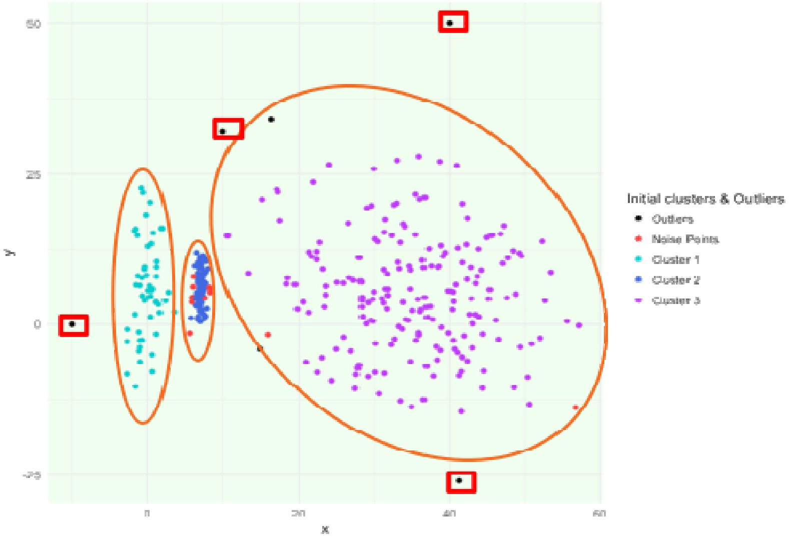

Figure 5. Outlier Detection

IV. **Cluster Proposal:**In this *optional* final step, we wrap up the process by suggesting a cluster for the set of points $\{p: \Psi\ (p) = 0\}$, i.e. points which are noise points but not anomalies. User may choose to run this step, or may ignore it and keep non-anomalous noise points as is based on application requirement

This is a rather easy task as we already have somewhat labelled scenario. Let, $\mathcal{C}_j$ denote the j^{th}cluster – We do the following:

```
whilep $\in \mathcal{D}$do:

        if $\Psi(p) = 0$then
                for $j\ in\ 1\ to\ max(\Psi)$do:
                        $\delta_p(j) = \frac{1}{|\mathcal{C}_j|} \sum_{q \in \mathcal{C}_j} d(p,q)$
                end for
                if$l = argmin\ (\delta_p)$then
                $\Psi(p) \leftarrow l$
            end if
        end if
end while
```

As we see (Fig 6.) this clears up the remaining points by assigning them a cluster closest to them. This concludes our algorithm.

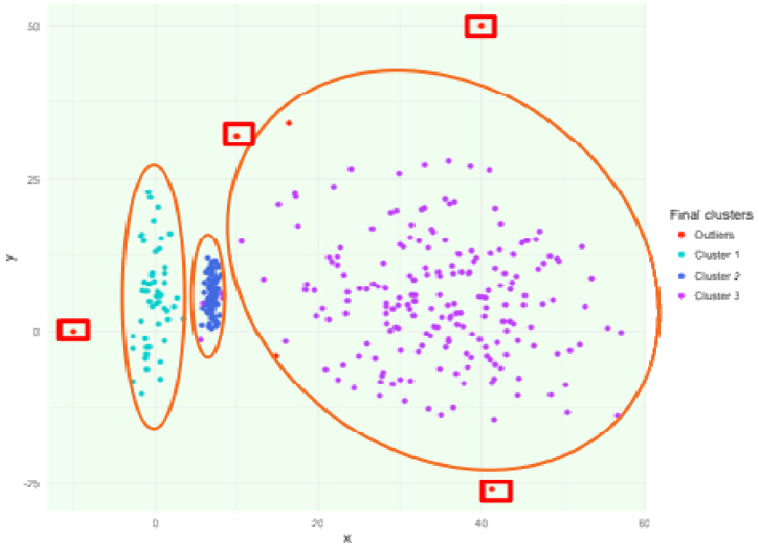

Figure 5. Final Cluster Proposal

4. EXPERIMENTS

In this section we show results of the experiments we have performed over two 2D synthetic datasets, only 2D datasets is demonstrated here partly because similar data sets have been used most extensively by other authors [23] and partly because it is easy to evaluate the quality of clustering on 2D data sets by naked eye – hence these are better suited for space constrained scenarios such as these.

4.1. Benchmarks

For comparison purposes, we will be using two other algorithms. First is DBSCAN, which is probably the most renowned and most used density-based clustering algorithm. Second is SNN based clustering proposed in [24], which has shown empirical superiority over similar methods such as k-means, DBSCAN, CURE [25] etc. We have chosen these two algorithms as the anatomy of these two matches with our algorithm – as all three revolve around the idea of identifying core points and building clusters around them. All of these do not require number of clusters to be defined by user and works better than other methods when applied on spatial data. However, as mentioned earlier: While DBSCAN can find clusters of arbitrary shapes, it cannot handle data containing clusters of differing densities, since its density-based definition of core points cannot identify the core points of varying density clusters; something that SNN seems to alleviate, proving to be superior in terms of identifying clusters of widely different shapes, sizes, and densities.

Authors would like to emphasize, since the parametrization of all three algorithms used here are very fluid and different values of parameters provide vastly different results, we have experimented with a wide array of inputs for all of the algorithms, and will only be sharing the best outcomes for individual algorithms and their corresponding input values.

4.1. Synthetic Dataset 1

Coming first is the dataset we have used for illustration purposes throughout the paper; Figures 6, 7show how DBSCAN and SNN perform on this dataset respectively

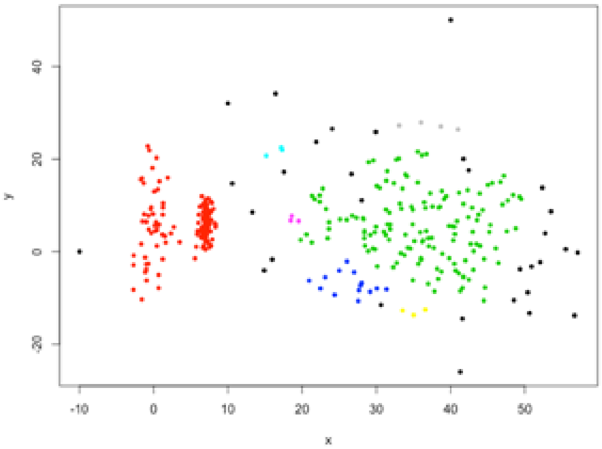

Figure 6. DBSCAN on Dataset1: $\epsilon = 3; Minpt = 3$

We can see even at its best, DBSCAN fails miserably – over-fragmenting C_3 which is the low-density cluster and mixing the other two higher-density cluster together all the while creating plenty of noise point for the user to deal with.

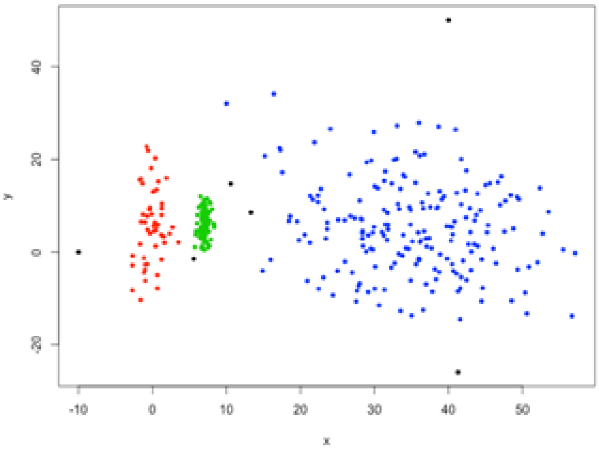

Figure 7. SNN on Dataset1: $\epsilon = 3; Minpt = 3; k = 12$

SNN on the other hand is quite adept at handling clusters with varying density identifying the clusters near perfectly, however, it fails to observe the anomalies, labelling some points as anomalies when they are actually part of clusters and failing to identify 1 out of the 4 outliers. However, one should acknowledge SNN as a huge improvement over more traditional DBSCAN.

Fig 8 shows REDCLAN almost perfectly identifies every cluster and also recognises 4 (and only 4) outliers in the dataset, only mis-classifying one point which as a boundary point. This dataset provides a great case study - on one hand without doubt our algorithm surpasses DBSCAN, it also enjoys a unique edge over algorithms such as SNN which do correct for varying density but don't have any way of differentiating between noise points created and actual outliers. This makes REDCLAN somewhat Swiss army knife for data mining tasks - which is reflected not only in quality of result but also user-friendliness and satisfaction.

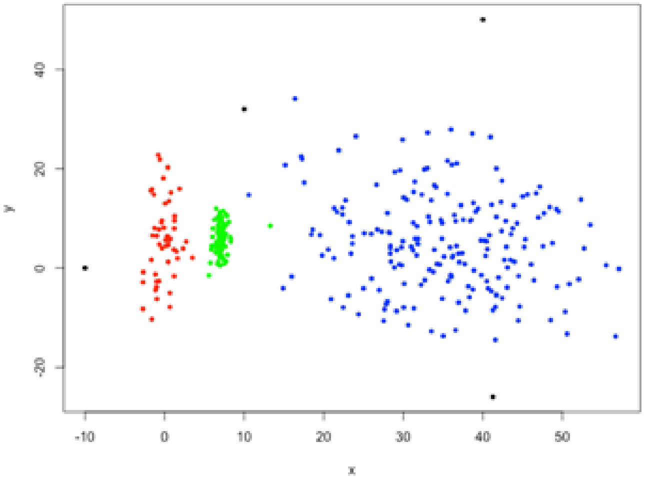

Figure 8. REDCLAN on Dataset1: $k_1 = 4; k_2 = 11$

4.3. Synthetic Dataset 2

This dataset (Fig 9) was originally part of CHAMELEON [23] study and is publicly available as part of the R package 'seriation' [26]. We can see 8 different clusters of vastly different shape, size and density floating in a pool of noise points. This appears to be a comprehensive test of competence for algorithms working on low-dimensional spatial data.

The results of the algorithms on this data can be viewed in Figures10,11,12.

Again, one can note similar results and a clear hierarchy of proficiency among the three algorithms, DBSCAN when faced with various degrees of densities gives unsatisfactory results; unnecessarily creating smaller clusters in a low-density cluster and merging two higher density cluster just as earlier. SNN,as expected, performs better than DBSCAN at least in terms of identifying lower – density clusters correctly. However, it falls short of the accuracy it gained in the previous dataset. In fact, we can see merging of higher – density cluster here too: possibly due to inability to adapt to such changes in density in the data. Moreover, the pool of noise points creates problems for SNN; it ends up creating small inconsequential clusters among these points.
The ability of REDCLAN in dealing with all these issues can be demonstrated here. It again outperforms the other two methods by pinpointing the 8 clusters and the surrounding noise points. One can notice however, few noise points are assigned a cluster number - this is due to the fact that they are so close to the cluster spatially, they almost act as boundary points.

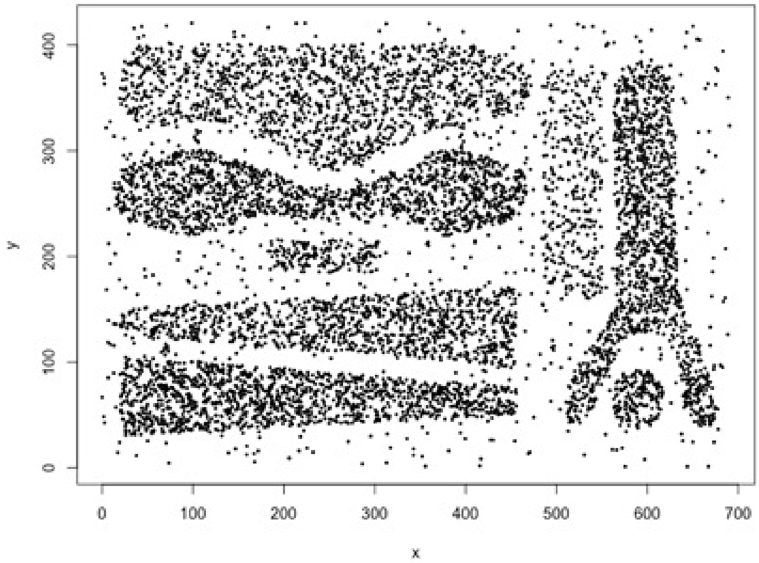

Figure 9. CHAMELEON dataset

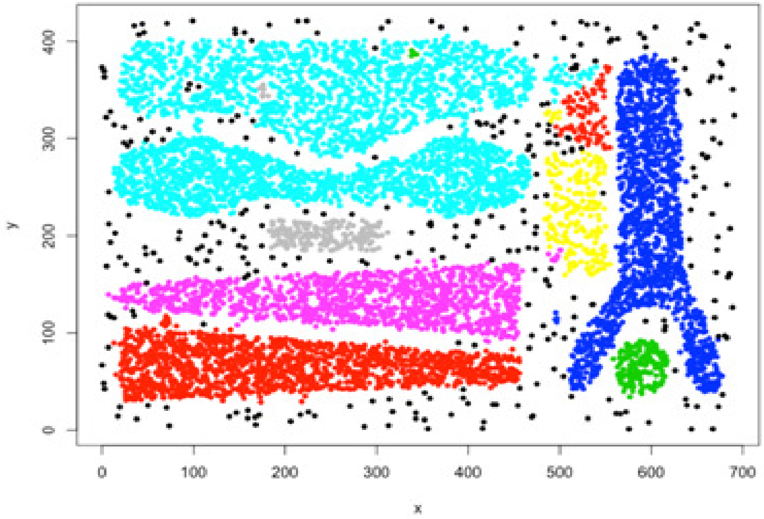

Figure 10. DBSCAN on Dataset1: $\epsilon = 8; Minpt = 4$

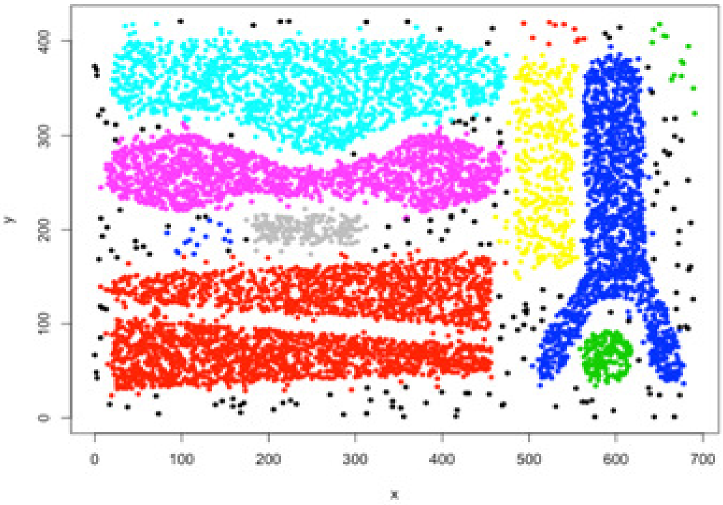

Figure 11. SNN on Dataset2: $\epsilon = 5; Minpt = 10; k = 15$

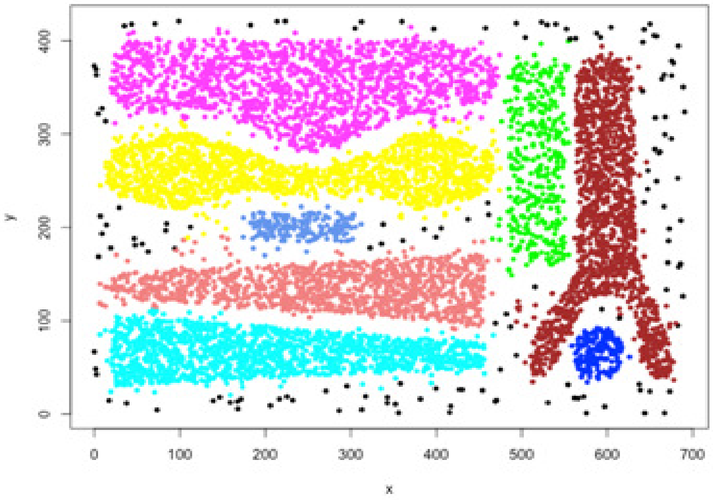

Figure 12. REDCLAN on Dataset2: $k_1 = 35; k_2 = 14$

5. CONCLUSIONS

In this paper, we present a novel technique of clustering and anomaly detection where both work in a complementary fashion. We have established the case for identification of varying density clusters which is the most practical case owing to the multifarious nature of the data. Our methodology shows notable improvements over previous density-based clustering methods like DBSCAN and SNN which are popularly used. Even though we have demonstrated the performance on synthetic datasets for the sake of comparison with previous methods, our technique particularly becomes effective while dealing with various problems in e-commerce and finance. Identifying various minute classes of substitutes or finding database anomalies from a large streaming data or identifying anomalous behaviour in the buyer-seller network are some of the prominent use-cases where our method has seen success.

REFERENCES

[1] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.

[2] D. c. a. review, "A.K. Jain; M.N. Murty; P.J. Flynn," ACM COMPUTING SURVEYS, vol. 31, no. 3, 1999.

[3] Y. Zhao and G. Karypis, "Data clustering in life sciences," Molecular Biotechnology, vol. 31, no. 1, pp. 55-80, 2005.

[4] F. H. Bprgen and D. C. Barnett, "Applying Cluster Analysis in Counseling Psychology Research," Journal of Counseling Psychology, vol. 34, no. 4, pp. 456-468, 1987.

[5] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, San Francisco: Morgan Kaufmann Publishers Inc., 2011.

[6] P. Berkhin, "A Survey of Clustering Data Mining Techniques," in Grouping Multidimensional Data, Springer, 2006, pp. 25-71.

[7] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," Annals of Data Science, vol. 2, no. 2, pp. 165-193, 2015.

[8] R. R. ,. L. T Zhang, "BIRCH: an efficient data clustering method for very large databasesss," in ACM Sigmod Record, 1996.

[9] S. Guha, R. Rastogi and K. Shim, "ROCK: A robust clustering algorithm for categorical attributes," Information systems, vol. 25, no. 5, pp. 345-366, 2000.

[10] MacQueen and James, "Some methods for classification and analysis of multivariate observations," in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, 1967.

[11] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," Expert systems with applications, vol. 36, no. 2, pp. 3336--3341, 2009.

[12] M. E. Xu, H.-P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in KDD'96 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining Pages, Portland, 1996.

[13] M. A. Sander, M. M. Breunig, H.-P. Kriegel and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," in International Conference on Management of Data, 1999.

[14] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, vol. 41, no. 3, 2009.

[15] J. Laurikkala, "Informal Identification of Outliers in Medical Data," in Proceedings of the Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology, Berlin, 2002.

[16] F. E. Grubbs, "Procedures for Detecting Outlying Observations in Samples," Technometrics, 1969.

[17] P. J. Rousseeuw and A. M. Leroy, Robust Regression and Outlier Detection, New York: John Wiley & Sons, Inc., 1996.

[18] H.-P. K. R. T. N. J. S. Markus M. Breunig, "LOF: Identifying Density-Based Local Outliers," in ACM SIGMOD, Dallas, 2000.

[19] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies.," Artificial Intelligence Review, vol. 22, no. 2, p. 85–126, 2004.

[20] I. Syarif, A. Prugel-Bennett and G. Wills, “Unsupervised clustering approach for network anomaly detection,” Communications in Computer and Information Science, vol. 293, 2012.

[21] G. R.Terrell and D. W. Scott, “Variable Kernel Density Estimation,” The Annals of Statistics, pp. 1236-1265, 1992.

[22] H. Huang, “Rank Based Anomaly Detection Algorithms,” Electrical Engineering and Computer Science - Dissertations.331, 2013.

[23] G. K. Kumar, E.-H. (. Han and V. Kumar, “CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling,” Computer, vol. 32, no. 8, pp. 68-75, 1999.

[24] L. E. a. M. S. a. V. Kumar, “Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data,” in SDM, 2003.

[25] S. Guha, R. Rastogi and K. Shim, “Cure: an efficient clustering algorithm for large databases,” Information Systems, vol. 26, no. 1, pp. 35-58, 2001.

[26] M. Hahsler, K. Hornik and C. Buchta, “Getting things in order: An introduction to the R package seriation,” Journal of Statistical Software, vol. 25, no. 3, pp. 1-34, 2008.

[27] A. K. a. D. R. C. Jain, Algorithms for clustering data, Prentice-Hall, Inc., 1988.

[28] X. Xu, M. Ester, H.-P. Kriegel and J. Sander, “A distribution-based clustering algorithm for mining in large spatial databases,” in Data Engineering, 1998. Proceedings., 14th International Conference on, 1998.

[29] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A k-means clustering algorithm,” Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pp. 100-108, 1979.

INTENTIONAL BLANK

WHY WE NEED A NOVEL FRAMEWORK TO INTEGRATE AND TRANSFORM HETEROGENEOUS MULTI-SOURCE GEO-REFERENCED INFORMATION: THE J-CO PROPOSAL

Gloria Bordogna[1] and Giuseppe Psaila[2]

[1]CNR IREA - Via Bassini 15 - 20133 Milano - Italy
[2]University of Bergamo - Viale Marconi 5 – 24044 Dalmine (BG) - Italy

ABSTRACT

The large number of geo referenced data sets provided by Open Data portals, social media networks and created by volunteers within citizen science projects (Volunteered Geographical Information) is pushing analysts to define and develop novel frameworks for analysing these multisource heterogeneous data sets in order to derive new data sets that generate social value. For analysts, such an activity is becoming a common practice for studying, predicting and planning social dynamics. The convergence of various technologies related with data representation formats, database management and GIS (Geographical Information Systems) can enable analysts to perform such complex integration and transformation processes. JSON has become the de-facto standard for representing (possibly geo-referenced) data sets to share; NoSQL databases (and MongoDB in particular) are able to natively deal with collections of JSON objects; the GIS community has defined the GeoJSON standard, a JSON format for representing georeferenced information layers, and has extended GIS software to support it.

However, all these technologies have been separately developed, consequently, there is actually a gap that shall be filled to easily manipulate GeoJSON objects by performing spatial operations. In this paper, we pursue the objective of defining both a unifying view of several NoSQL databases and a query language that is independent of specific database platforms to easily integrate and transform collections of GeoJSON objects. In the paper, we motivate the need for such a framework, named J-CO, able to execute novel high-level queries, written in the J-CO-QL language, for JSON objects and will show its possible use for generating open data sets by integrating various collections of geo-referenced JSON objects stored in different databases.

KEYWORDS

Collections of JSON objects, Geo-tagged data sets, Query Language for geographical analysis, Powerful spatial operators

1. INTRODUCTION

Geo-referenced information from Open Data portals, Volunteered Geographic Information (VGI), and crowdsourced information from social networks are recognized as a potential driver for social changes: companies are relying on such information to enhance their existing services or to derive knowledge from its analysis to create social value [3]. As described in the European Commission report on the reuse of open data, the European Data Portal has more than doubled the amount of

Natarajan Meghanathan et al. (Eds) : SOFE, ADCOM, ITCSS - 2018
pp. 41–60, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81304

data it references. In general, a large number of sources are now available to get information about territories, including corpora managed by private companies like Google and Facebook, that collect and integrate official information, VGI and crowd-sourced information about any kind of place.

In order to turn such geo-referenced information into social value, the so-called data-value chain process must be carried out: once geo-referenced data (geo-data for short) are created, they have to be validated, for example through filtering, normalization and quality assessment, and shared by means of a Web geo-portal, after which they can be analysed. From integrating different geo-data sets, new data can be created, which can lead to new data services or products. It can be seen that, in order for a geo-data analyst to perform such tasks in an easy way, a framework is needed so as he/she can perform several manipulation operations on geo-data, which are heterogeneous, as far as their source, structure, format and semantics are concerned.

In effect, performing integration and transformation processes asks for the convergence of various technologies, originally developed separately, that now all together contribute to this ambitious goal. In particular, we consider data representation formats, database management and GIS (Geographical Information System) technology.

Let us start with the area of *data representation formats*. After the introduction of XML (eXtensible Mark-up Language) at the end of the 1990s, that had to become “the language” for information interchange on the Internet, currently we are observing the rapid diffusion of JSON (JavaScript Object Notation) as a de-facto standard for data interchange, in particular through API interfaces and Web Services. JSON is a flexible format to encode semi-structured compact information. Often, data sets provided by Open Data portals as well as by Web Service APIs contain geo-referenced information, i.e., data are tagged with positions on the Earth Globe (in terms of longitude and latitude), since they describe data concerning territories.

As far as the area of *Database Management* is concerned, the last decade is characterized by the development of so-called *NoSQL databases*, i.e., DBMSs (Data Base Management Systems) which are not based on the relational data model and, consequently, abandon SQL as query language. In particular, value-store, column-store and document-store are different models of NoSQL DBMS, where document stores are able to manage collections of JSON objects in a native way. The most famous representative of this category is *MongoDB* designed to manage large amounts of (relatively small) JSON objects, even though it provides spatial indexes that enable to efficiently perform some types of spatial queries.

As far as *GIS Technology* is concerned, GIS tools are now even more important than in the recent past: in fact, the availability of geo-data sets asks for visualizing such data on maps in the form of information layers, possibly integrating data from distinct data sets. In this respect, it was essential to introduce a standard format for describing information layers: the GIS community has defined the *GeoJSON* format, i.e., a standard format for describing geographical information layers that relies on JSON as syntactic framework.

Apparently, the above-mentioned convergence should be mature, but this is not true. In fact, the different perspectives that have driven the diverse developments make actually difficult to easily and effectively transform and integrate JSON data sets and/or GeoJSON layers, in particular when they are collected within databases managed by MongoDB (or, worse, in simple files). A unifying framework that provides analysists with the capability of effectively integrating and transforming JSON data sets and GeoJSON layers is essential.

These are the main reasons that motivated us to conceive a new framework, named J-CO. The goal of the framework is twofold: it has to provide the capability of working on different

MongoDB databases at the same time, allowing analysts to easily integrate data sets stored in different databases; it has to provide a query language, named J-CO-QL, for manipulating collections of JSON and GeoJSON objects, natively supporting spatial operations and representations. The paper will describe, through a study case example, how the defined query language can be used to integrate and transform JSON data sets to create GeoJSON layers: specifically, we consider quartiers and pharmacies of Milan (city in the northern Italy) and we will generate two GeoJSON layers describing quartiers with less than two pharmacies and quartiers with at least two pharmacies. We hope this way the paper will clarify how the J-CO framework can implement the convergence of the above-mentioned technology.

The paper is organized as follows. Section 2 presents the background of our project and related work. Section 3 introduces the J-CO framework, detailing the main features of the J-CO-QL language. Section 4 shows the example and how the J-CO-QL language can be used to perform complex transformations on collections of JSON data. Finally, Section 5 draws the conclusions.

2. RELATED WORKS

2.1. Motivation of the Proposal

Our seminal idea of developing a framework for integrating and transforming collections of JSON objects originated in the *Urban Nexus Project* [12]. The goal of this project is to gather information from several distinct open data repositories on the Web, authoritative and statistical sources, social media and so on, to study how city users live their city and territory. The idea is that geographical studies should take advantage of Big Data, in the sense of large variety of data sets coming from diverse data sources. In such a context, analysts are not programmers and need an integrated framework for performing their analyses. Nevertheless, since data come from many sources as JSON or GeoJSON data sets, it was necessary to develop a novel query language for this purpose. These considerations motivated the development of the J-CO framework.

In [5, 18, 19], we tackled the objective of exploiting social media to trace movements of social media users. We named this project *FollowMe*, because we traced (and we are still tracing) travellers that post geo-located messages on *Twitter*, detecting them in a pool of 30 airports potentially connected with the airport of Bergamo (northern Italy). Next [7], we integrated the *FollowMe* project within a framework for analysing trips of *Twitter* users, furthermore [8], we experimented a clustering technique for identifying common paths followed by users during their trips. That was a preliminary work of the *Urban Nexus* project, in which huge numbers of trips of *Twitter* users represented as JSON objects have to be analysed on the basis of multi-paradigmatic approach (see [12]). While facing this analysis task, we experienced the limitations of current query languages for heterogeneous geo-data in the form of JSON objects.

2.2. Manipulating Heterogeneous Big Geo-Data

The first attempt to abandon the relational data model in favour of more flexibility on the structure of data dates back to the late 1990s with the advent of XML (eXtendible Mark-up Language) as the universal data format for exchanging data over the Internet that stimulated the idea of developing database technology for storing and querying XML documents. Many proposals for XML databases and related query languages were proposed. The reader can refer to [22, 24,27] for some surveys. Obviously, speaking about convergence of technologies, the research area of data mining met the research area of XML-native databases, in order to perform data mining and knowledge discovery directly on XML documents stored within XML databases [30,33]. The idea is that the ability of XML to represent semi-structured and complex data enables to store, within the same XML database, both the mined data and the mined patterns.

However, the potentiality of XML to become “the representation format” met several practical obstacles that have limited its diffusion and favoured the emergence of JSON, namely its extreme verbosity and difficulty of importing data described by XML documents within programs and information systems.

The adoption of JSON and NoSQL databases are motivated by the need for both flexibility and compactness as far as data structures are concerned; an interesting survey on NoSQL databases can be found in [12], where several systems are catalogued and classified. In particular, a DBMS like MongoDB falls into the category of document databases, because collections of JSON objects are generically considered as documents [26]. The query language provided by such systems does not allow complex and multi-collection transformations. Readers interested in NoSQL DBMSs evaluation can refer to [37] and to [15].

As far as query language for JSON objects are concerned, several proposal were made. However, none of them is explicitly designed to provide geographical data analysis capabilities, natively integrated in a high level query language, as for J-CO-QL [5]. Anyway, it is worth mentioning them.

Jaql (see [34]) was designed to help Hadoop (see [40]) programmers writing complex transformations, avoiding low-level programming, to perform in a cloud and parallel environment. Flexibility and physical independence are the main goals of Jaql: in particular, its execution model is similar to our execution model, since it explicitly relies on the concept of pipe; in fact, the pipe operator is explicitly used in Jaql queries. However, it is still oriented to programmers; its constructs are difficult to understand for non programmer users.

An interesting proposal is SQL++,defined to query both JSON native stores and SQL databases. The SQL++ semi-structured data model is a superset of both JSON and the SQL data model [35]. Yet, SQL ++ is SQL backwards compatible and is generalized towards JSON by introducing only a small number of query language extensions to [35]. In SQL++ the classical SELECT statement of SQL is adapted and extended to perform queries on collections of JSON objects. In our opinion, this is a clean proposal, if compared with others, that tries to work at a higher abstraction level. However, it does not deal explicitly with heterogeneity of objects, i.e., it does not provide constructs similar to the WHERE branches provided by J-CO-QL. Furthermore, complex transformations that require several queries sequentially would executed need to explicitly save intermediate results into the persistent database (although in [35] nothing is said about data manipulation operators such as INSERT). In contrast, the execution model on which J-COQL relies clearly separate persistent databases and temporary databases, by means of the temporary collection and the intermediate result database IR.

The industry is looking at the extension of SQL to query JSON objects. An example is N1QLyy that is a declarative language extending SQL for JSON objects stored in NoSQL databases, specifically implemented for *Couchbase 4.0*, in order to handle semi-structured, nested data. It enables querying JSON documents without any limitations sort, filter, transform, group, and combine data with a single query from multiple documents with a JOIN. Nevertheless it does not provide operators to manipulate GeoJSON objects. Finally, other declarative languages for JSON objects have been defined as extensions of structured languages for semi-structured documents, such as JSONiq, that borrowed a large numbers of ideas from XQuery, like the functional aspect of the language, the semantics of comparisons in the face of data heterogeneity, the declarative snapshot-based updates. However, unlike XQuery, JSONiq is not concerned with the peculiarities of XML, like mixed content, ordered children, or the complexities of XML Schema, and so on. Nevertheless, like XQuery it can be hardly used by unexperienced users.

Although these languages are declarative, they are still oriented to a programmer vision.

Other approaches to manipulate heterogeneous big data recognize the importance of a declarative query language to guarantee the data independence principle [20, 26]. For example, SparkSQL [1], was developed with an SQL interface to query heterogeneous big data sets managed within the Spark distributed processing infrastructure. It introduces a new data abstraction called SchemaRDD, which provides support for structured and semi-structured data. Nevertheless, it does not support spatial operators.

GeoSPARQL [2] is a Geographic Query Language for RDF Data proposed as standard by the OGC consortium for querying geospatial data on the Semantic Web. GeoSPARQL is designed to accommodate systems based on qualitative spatial reasoning and systems based on quantitative spatial computations to ease data analysis.

Also our proposal is oriented to data analysts, which need to manage heterogeneous collections of real world entities, namely collection in both JSON and GeoJSON.

It is somehow related to the world of *PolyStore DBMS*, i.e., database management systems that deal with several DBMS at the same time, each of them possibly providing a different logical model, such as relational, graph, JSON, pure-text, images, videos. An interesting work on this topic is *BigDAWG*.

Our proposal moves from our previous work on the problem of querying heterogeneous collections of complex spatial data (see [11, 36]). In that works, we proposed a database model capable to deal with heterogeneous collections of possibly nested spatial objects, based on the composition of primitive spatial objects; at the same time, an algebra to query complex spatial data is provided, inspired by classical relational algebra. W.r.t. those previous works, J-CO-QL relies on the JSON standard, thus we do not define an ad-hoc data model; furthermore, J-CO-QL abandons the typical relational algebra syntax, because it relies on a more flexible and intuitive execution model. Nevertheless, the experience made in [28] helped us, where we defined a language for manipulating clusters of web searches performed through a mobile device.

3. A Framework For Georeferenced Data Transformations

The framework we conceived for integrating and transforming multisource geo-referenced JSON data is named J-CO (*J*SON *CO*llections) and comprehends several components depicted in Figure 1.

- One or more NoSQL databases managed by MongoDB (and, in the future, by other systems like *ElastichSearch*). This feature resembles a distributed federated database architecture [28].

- The *J-CO-QL Engine* that executes queries directly operating on data stored in MongoDB databases. It receives queries through a Web Service interface. This feature is typical of distributed databases querying [31].

- A GIS application, like, e.g., QGIS [23], the open source GIS software, that can be used as environment to query and to show GeoJSON layers.

- A (future) User Interface. Through this interface, the analysis will be able to carry on the transformation process dynamically. This interface can be developed as a plug-in of a GIS application.

- Any kind of publishing tool for geo-data stored within MongoDB-managed databases, such as Open Data geo portals, Web Map services and Web Feature Services [32], etc.

Note that the ability of the J-O-QL Engine of connecting with several databases during the same transformation process is a key feature: in fact, this feature allows analysts to easily integrate data sets, by taking them from the servers that store them. This way, it is possible to avoid a large amount of efforts for transferring data from one server to another, in accordance with optimization techniques in loosely coupled federated databases [13]. In them, a unique schema for queries does not exists but a uniform query language is made available ,which abstracts from the query languages of the components, and hides technical and language heterogeneity. Thus, every user is responsible for handling logical heterogeneity in the components.

Nevertheless, the possibility of visualising data and results of the analysis through a GIS software can greatly help analysts formulate queries and perform a visual analysis of results. This can be done by generating GeoJSON layers during the transformation process performed by means of the J-CO-QL Engine.

On the same line, it is important to be able to publish results anywhere, for example in Open Data Portals. These portals often provide geo-referenced data as GeoJSON layers, but this is not mandatory. Often, when data are not geo-referenced, simple JSON collections are published. In this scenario, the J-CO framework is designed to play a central role, towards the simplification of tasks that, without it, could be very tedious and much more time consuming than necessary.

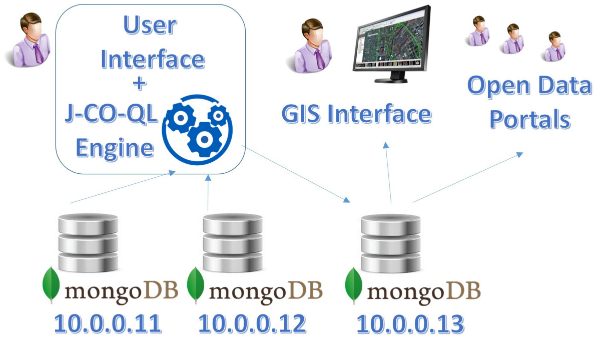

Figure 1. Application Scenario for the J-CO Framework

3.1. J-CO-QL Main Features

The query language named J-COQL is defined to work with *collections of JSON objects*. JSON is a serialized representation for objects. Fields (object properties) can be simple (numbers or strings), complex (i.e., nested objects) and vectors (of numbers, strings and objects).

JSON does not consider geo-references. An official proposal in this sense is the GeoJSON standard [14, 17]. Defined by the GIS community, it provides an excellent format for defining geometries of geo-referenced data. Fields describing geometries are named, in the GeoJSON standard, `geometry`. In J-CO-QL, we rely on the same standard for representing geometries, but the name of the field considered by the J-CO-QL language to handle geometries is `~geometry`: this way, J-CO-QL is able to handle GeoJSON layers in a seamless way.

When the `~geometry` field is absent in an object processed by J-CO-QL, this means that no geo-reference is present in the object and no spatial operations can be performed on it. Specifically, the `~geometry` field (when present) is based on the `GeometryCollection` type for the GeoJSON standard.

Figure 2, reports sample objects of JSON objects with `~geometry` field. The reader can see objects describing quartiers of Milan, as well as objects describing pharmacies in Milan. Coordinates (longitude and latitude) are expressed based on the (World Geodetic System) 84, our default CRS (Coordinate Reference System).

In a NoSQL environment such as MongoDB, a *Database* is viewed as a *set of collections*, while a *Collection* has a *name* and its instance is viewed as a vector of JSON objects. To manipulate JSON collections and to store their results into new collections, in a transparent way w.r.t. to the databases from which to get collections and to which to store collections, we need operators that meet the *closure property*, that is, they get collections and generate collections. This is a first design requirement for the J-CO-QL language [38]. Other key features of the language are reported hereafter.

- J-CO-QL provides operators specifically designed to deal with objects with different structure within the same operation.

- Operators provided by J-CO-QL are high-level operators, which allow analysts to think directly to objects structure; they do not have to write low-level procedures.

- Finally, but not less important, J-CO-QL directly deals with geo-reference possibly contained in JSON objects, because the data model explicitly deals with them through the `~geometry` field.

Queries are sequences of operators applied to collections [16, 29]. The execution process of queries is based on the concept of *state of the query process*, that is a pair *s* = (*tc*; *IR*), where *tc* is a collection named *Temporary Collection*, while I*R* is a database named *Intermediate Results database*.

Each operator starts from a given query process state and generates a new query process state. During the process, the J-CO-QL Engine, can be asked (by a suitable operator) to store *tc* (the *Temporary Collection*) into IR (the *Intermediate Results database*), that could be taken as input by a subsequent operation. Obviously, J-CO-QL provides an operator to store the temporary collection into a persistent database, (a database managed by MongoDB) as well as an operator to get a collection from IR or from a persistent database as new temporary collection. In fact, the idea is that an operator takes the temporary collection as input and generates a new instance of the temporary collection as output. The reader can see an execution trace in Figure 6, depicting the execution process of the query presented in Section 4.1.

The J-CO-QL Engine executes each query process in isolation: several users can use the engine at the same time. Thus, the goal of the IR database, one for each query process, is twofold. First of all, it permits to temporarily store intermediate results of the process, that do not have to be stored in persistent database (since they are intermediate). Second, it ensures isolation of query process execution as far as intermediate results are concerned.

```
Collection Quarters: [
{"ID ": 74, "Name": "SACCO",
 "~geometry": {"type": "MultiPolygon",
               "coordinates":  [  [  [  [  9.121949242919204,
45.516020899111012     ],     ,…,     [     9.121949242919204,
45.516020899111012 ] ] ] ] } },
{"ID ": 82, "Name": "COMASINA",
 "~geometry": {"type": "MultiPolygon",
               "coordinates":  [  [  [  [  9.168870308198338,
45.523965029425476     ],     …,     [     9.168870308198338,
45.523965029425476 ] ] ] ] } },
…]


Collection Pharmacies: [
{"Address": "Via ANGELONI LUIGI",
 "Name": "COMUNALE N.33",
 "~geometry": {"type": "Point",
               "coordinates": [ 9.17529749573575,
                                45.527028643302899 ] } },
{"Address": "Via CASARSA 130",
  "Name": "CASARSA",
   "~geometry": {"type": "Point",
                 "coordinates": [ 9.174392445342329,
                                  45.524802296955897 ]} },
…]
```

Figure 2. Excerpt of collection Quartiers and collection Pharmacies

4. Generating Open Data Concerning Geo-referenced Contents

In this section, we practically show the effectiveness of our framework. We consider real data sets coming from the Open Data portal (url: `https://dati.comune.milano.it/`) of Milan (Italy) City Council.

Suppose we own an information system having three MongoDB servers, whose toy IP addresses are 10.0.0.11, 10.0.0.12 and 10.0.0.13. The first one, with address 10.0.0.11, has a database named `Boundaries`, that contains collections concerning some cities; we say that collection `Milan_Quartiers` contains JSON objects describing quartiers in Milan. The upper part of Figure 2 shows a few objects describing quartiers.

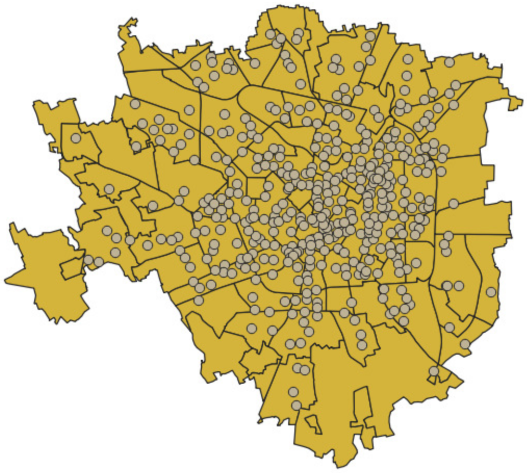

Figure 3. Quartiers and Pharmacies in Milan (Italy)

Server with address 10.0.0.12 manages a database named `MilanInfo`: it contains collections related with territory. As an example, consider collection `Pharmacies`, that describes pharmacies in Milan. We report a few objects in collection `Pharmacies` in the lower part of Figure 2.

Finally, server 10.0.0.13 manages the database named `dbToPublish` that has to store collections containing GeoJSON layers to publish as open data.

The sample process we show in the rest of this section has the following goals:

- Count, for each quartier, the number of pharmacies in the quartier.
- Create two GeoJSON layers to save into a collection named `NilanQuartiersAndPharmacies` and store it in database `dbToPublish`: one layer is named `"Few Pharmacies"` and describes quartiers having less than 2 pharmacies; the other layer is named `"Many Pharmacies"` and describes all other quartiers.

Figures 3 and 7 graphically illustrate the process. We start from the descriptions of quartiers (brown-filled polygons) and pharmacies (grey points). We want to obtain the two layers jointly depicted in Figure 7: red-filled polygons are quartiers with less than 2 pharmacies; green-filled polygons are quartiers with at least 2 pharmacies. The two main steps of the process are described in Sections 4.1 and 4.2, where J-CO-QL queries able to perform the process are presented.

4.1. Counting Pharmacies

The J-CO-QL query for performing the first task, i.e., counting pharmacies in each quartier, is reported hereafter. Afterwards, we will describe single operators and the full process.

```
USE DB Boundaries
ON SERVER MONGODB "http://10.0.0.11:2707";
USE DB MilanInfo
ON SERVER MONGODB "http://10.0.0.12:2707";
USE DB dbToPublish
On SERVER MONGODB "http://10.0.0.13:2707";

SPATIAL JOIN OF COLLECTIONS
   Milan_Quartiers@Boundaries AS Q, Pharmacies@MilanInfo AS P
ON INCLUDED(RIGHT)
SET GEOMETRY LEFT
CASE
WHERE WITH Q.QID, P.Name
GENERATE {.QID: .Q.QID, .Name: .Q.Name, .PharmacyName: .P.Name}
KEEPING GEOMETRY
DROP OTHERS;

GROUP
PARTITION WITH .QID, .PharmacyName
BY .QID, .Name, .~geometry INTO .Pharmacies
DROP OTHERS;

FILTER
CASE WHEN WITH .QID, .Pharmacies
GENERATE {.QID, .Name, .NumOfPharmacies: COUNT(.Pharmacies)}
   KEEPING GEOMETRY

SET INTERMEDIATE AS QuartiersWithPharmacies;

FILTER
CASE WHERE WITH .QID, .Name
 GENERATE {.ID,.Name} KEEPING GEOMETRY
DROP OTHERS;

SUBTRACT COLLECTIONS Milan_Quartiers, TEMPORARY;

FILTER
CASE WHERE WITH .QID, .Name
 GENERATE {.ID,.Name, .NumOfPharmacies: 0}
   KEEPING GEOMETRY
```

```
DROP OTHERS;
MERGE COLLECTIONS TEMPORARY, QuartiersWithPharmacies;
SET INTERMEDIATE AS QuartiersWithPharmaciesCount;
```

We now describe the query. Notice that the execution trace is depicted in Figure 6.

First of all, it is necessary to specify to which databases to connect. The first three USE DB operators do this work. Notice that the ON clause specifies the connection string necessary to connect to the desired MongoDB server.

The real procedure starts with the SPATIAL JOIN operator. This is the key operator provided by J-CO-QL, in order to perform complex transformations concerned with geo-referenced data. Recall that collection Milan_Quartiers describes quartiers in Milan: each object in the collection contains a field named ~geometry, that describes the boundary of the quartier as a polygon. This collection is aliased as Q in the operator. On the other side, collection Pharmacies contains objects whose field ~geometry denotes the point where the pharmacy is located. This collection is aliased as P in the operator.

The SPATIAL JOIN operator computes pairs of objects in the two collections, such that the spatial join condition specified in the ON clause is satisfied. Specifically, a pair of objects is built if the geometry of the right object (in this case, coming from collection Pharmacies) is included in the geometry of the left objects (in this case, coming from collections Milan_Quartiers). The SET GEOMTERY clause specifies the geometry to assign to the object obtained by pairing the two original ones: we specify that we want to maintain the geometry of the left object, i.e., the boundary of the quartier. The upper part of Figure 4 reports an excerpt of the objects resulting from the generation of pairs satisfying the spatial join condition. Notice field Q that contains the original object coming from collection aliased ass Q, field P that contains the original object coming from the collection aliased as P and the ~geometry field resulting from the join (in this case, it coincides with the left geometry, as specified in the operator).

The subsequent CASE WHERE clause is necessary to restructure the objects, removing nesting. The WHERE selection condition uses the WITH predicate, that selects objects having the desired fields; then, the GENERATE sub-clause specify how to restructure each object that satisfies the condition; note that we want to maintain the geometry (KEEPING GEOMTRY option). The lower part of Figure 4 reports an excerpt of the temporary collection t_1 (as reported in Figure 6) resulting from the SPATIAL JOIN; notice how the CASE WHERE block restructured the output objects.

At this point, it is necessary to group together objects resulting from the SPATIAL JOIN in order to count the number of pharmacies in each quartier.

The GROUP operator is, intuitively, similar to the GROUP BY clause of SQL. However, it is specifically designed to work with collections of heterogeneous objects. Thus, the goal of the PARTITION clause is to select objects (from the t*emporary* collection produced by the previous operator) that have some common fields or characteristics. In the query, we select objects having fields QID (quartier identifier) and PharmacyName (in other words, we define a partition of the full set of objects); objects in the partition are then grouped on the basis of fields QID, Name and ~geometry field, as specified by the BY clause. For each identified group of objects, a new object is put into the output collection, such that all common fields are reported and a new field, an array of grouped objects named `Pharmacies` (as specified by the `INTO` clause) is added.

Within `SPATIAL JOIN`: and Before `CASE WHERE [`

```
{"Q": {"QID": 83,
       "Name": "BRUZZANO",
       "~geometry": {"type": "MultiPolygon",
                     "coordinates": […] } },
 "P": {"Address": "Via ANGELONI LUIGI",
       "Name": "COMUNALE N.33",
       "~geometry": {"type": "Point",
                     "coordinates": [ … ] } },
 "~geometry": {"type": "MultiPolygon",
               "coordinates": […]}
},
{"Q": { "QID": 83,
        "Name": "BRUZZANO",
        "~geometry": {"type": "MultiPolygon",
                      "coordinates": […] } },
"P": {"Address": "Via CASARSA 130",
      "Name": "CASARSA",
      "~geometry": {"type": "Point",
                "coordinates": […]} },
"~geometry": {"type": "MultiPolygon",
                          "coordinates": […] }
…]
```

At the end of `SPATIAL JOIN`, t_1 : `[`

```
{"QID": 83, "Name": "BRUZZANO",
 "pharmacyName": "COMUNALE N.33",
"~geometry": {"type": "MultiPolygon",
              "coordinates": […] }},
{"QID": 83, "Name": "BRUZZANO",
 "PharmacyName": "BRUZZANO",
 "~geometry": {"type": "MultiPolygon",
               "coordinates": […] }},
…]
```

Figure 4. Excerpt of objects generated by SPATIAL JOIN

At the end of GROUP, t_2: [

```
{"QID": 83, "Name": "BRUZZANO",
 "Pharmacies": [
{"QID": 83, "pharmacyName": "COMUNALE N.33",
 "~geometry": {"type": "MultiPolygon",
                         "coordinates": […] }},
{"QID": 83, "PharmacyName": "BRUZZANO",
 "~geometry": {"type": "MultiPolygon",
                         "coordinates": […] }}],
"~geometry": {"type": "MultiPolygon",
                         "coordinates": […] }}, },
…]
```

Temporary collection t_3 and Intermediate Collection QuartiersWithPharmacies: [

```
{"QID": 83, "Name": "BRUZZANO",
 "NumOfPharmacies": 2},
…]
```

Figure 5. Excerpt of objects generated by the objectys generated by the first GROUP operator and in collection saved into the QuartiersWithPharmacies IR database.

Note the presence of the ~geometry field in the BY clause: this is necessary to avoid the loss of geometry of quartiers during grouping (the geometry is implied by the quartier identifier). An excerpt of the temporary collection t_2, as numbered in Figure 6, is reported in the upper part of Figure 5.

At this point, it is necessary to add a field witch counts how many elements are present in array Pharmacies. The FILTER operator selects the desired objects and restructures them by adding the field named NumOfPharmacies. The lower part of Figure 5 reports the temporary collection t_3, as numbered in Figure 6, resulting from the FILTER operator.

The temporary collection is saved with name QuartiersWithPharmacies into the Intermediate Results Database. It will be used a few operators later. Notice in Figure 6 that the temporary collection in the state produced by SET INTERMEDIATE operator does not change (it is still labelled as t_3). In contrast, the IR database, depicted with empty braces in previous states, now contains the new saved collection.

Some quartiers may have not been produced by the spatial join, i.e., those quartiers without pharmacies. To restore them, we subtract quartiers with pharmacies from the full set of quartiers. Preliminarily, it is necessary to make the structure of objects in the temporary collection homogeneous with that of objects in collection Milan_Quartiers, with a FILTER operator that removes field NumOfPharmacies.

Then, the SUBTRACT operator performs the set-oriented difference between objects in collection Milan_Quartiers and the temporary collection (that actually contains quartiers with at least one pharmacy). Since only quartiers without pharmacies survive the difference, the next FILTER operator adds the missing NumOfPharmarcies field (set to 0). Finally, the MERGE operator unites the objects in the temporary collection and in collection QuartiersWithPharmacies, previously saved into the *Intermediate Results database*.

The first part of the process end ssaving the temporary collection into the Intermediate Results database with name `QuartiersWithPharmaciesCount`.

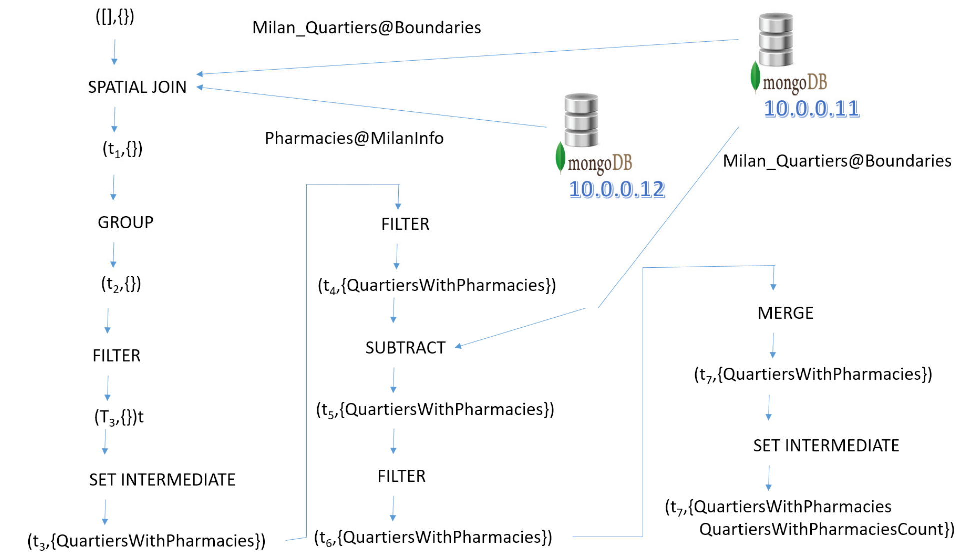
([],{})
Milan_Quartiers@Boundaries
SPATIAL JOIN
mongoDB
10.0.0.11
Pharmacies@MilanInfo
mongoDB
10.0.0.12
(t_1,{})
Milan_Quartiers@Boundaries
GROUP
FILTER
(t_2,{})
(t_4,{QuartiersWithPharmacies})
MERGE
FILTER
SUBTRACT
(t_7,{QuartiersWithPharmacies})
(T_3,{})t
(t_5,{QuartiersWithPharmacies})
SET INTERMEDIATE
SET INTERMEDIATE
FILTER
(t_7,{QuartiersWithPharmacies
QuartiersWithPharmaciesCount})
(t_3,{QuartiersWithPharmacies})
(t_6,{QuartiersWithPharmacies})

Figure 6. Execution trace for the query performing the first task

4.2. Generating GeoJSON Layers

The J-CO-QL query for performing the second task, i.e., generating the desired GeoJSON layers, is reported hereafter.

```
GET COLLECTION QuartiersWithPharmaciesCount;

FILTER
CASE WHERE WITH ..QID, .Name, .NumOfPharmacies AND
           .NumOfPharmacies < 2
  GENERATE {.ID, .Name, .LayerName: "Few Pharmacies"}
            KEEPING GEOMETRY
 WHERE WITH ..QID, .Name, .NumOfPharmacies AND
      .NumOfPharmacies >= 2
  GENERATE {.ID, .Name, .LayerName: "Many Pharmacies"}
            KEEPING GEOMETRY
DROP OTHERS;

FILTER
CASE WHEN WITH .ID, .Name, ..NumOfPharmacies, LayerName,
  GENERATE {.type: "Feature",
            .properties:{.QID, .Name,
                         .NumOfPharmacies,
                         .LayerName},
                         .geometry: .~geometry }
    DROPPING GEOMETRY
DROP OTHERS;

GROUP
PARTITION WITH .type, .properties, .geometry,
               .Properties.LayerName
BY .Properties.LayerName INTO .features
DROP OTHERS;

FILTER
CASE WHERE WITH .LayerName, .features
  GENERATE {.tye:"FeatureCollection",
            .name: .LayerName,
            .features}
DROP OTHERS;

SAVE AS MilanQuartiersAndPharmaciesdbBToPublish;
```

In order to generate two GeoJSON layers from objects stored in the intermediate collection `QuartiersWithPharmaciesCount`, we have to perform a sequence of transformations.

First of all, the operator `GET COLLECTION` retrieves the desired collection from the *Intermediate Results* database and makes it the temporary collection.

The subsequent `FILTER` operator adds a new field to objects, named `LayerName`. Notice the two `WHERE` conditions: if an object satisfies the first one, the new field has value `"Few Pharmacies"`; if an object satisfies the second condition, the new field has value `"Many Pharmacies"`. These are the names of the two layers we are going to generate, The second `FILTER` operator is necessary to restructure objects, in order to comply with the structure of GeoJSON features. In particular, notice the field specification `.geometry: .~geometry`, that is necessary to rename the `~geometry` field (required specified by the J-CO-Ql data model) into `geometry`, as required by the GeoJSON format.

At this point, it is possible to generate a layer by aggregating all objects having the same value for field `LayerName`. This is easily performed by the `GROUP` operator, which groups the objects based on the value of field `LayerName` nested within field `properties`.

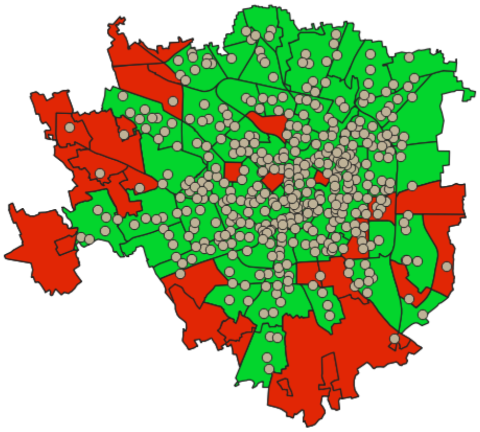

Figure 7. Layers "Few Pharmacies" (in red) and "Many Pharmacies" (in green)

The last FILTER operator is necessary to add the missing field type at the external level and rename field LayerName as name.

The obtained collection is saved into the dbToPublish database, that is managed by server with IP address 10.0.0.13. Hereafter, we report an excerpt of layer "Few Pharmacies", that is depicted in red in Figure 7. Layer "Many Pharmacies" is identical, apart from the described quartiers and the name of the layer.

```
{"type": "FeatureCollection",
 "name": "Few Pharmacies",
 "features": [
   {"type": "Feature",
    "properties": {"QID ": 74, "Name": "SACCO",
                   "LayerName": "Few Pharmacies"},
    "geometry": {"type": "MultiPolygon", "coordinates": [ … ] }
   },
   {"type": "Feature",
    "properties": {"QID": 75, "Name": "STEPHENSON",
                   "LayerName": "Few Pharmacies"},
    "geometry": {"type": "MultiPolygon", "coordinates":  [… ]}
   },
…] }
```

At the end of this section, we want to point out the major results we obtained with J-CO-QL.

- A J-CO-QL query is certainly a procedural specification, but it is not a procedural program, in the sense of classical procedural programming languages.

- The syntax of operators is English-like, inspired by the same approach adopted for SQL. This way, a certain degree of semantics of operators implicitly is expressed by the syntax.

- We are aware that operators need training to be properly used, but it is more intuitive for non programmers that other languages for JSON objects. For example, the same operations performed with the native query language of MongoDB could result a little bit hard to perform (probably exploiting JavaScript in many case, thus again a programming language).

- JSON collections are heterogeneous, i.e., they can contain objects with different structure. J-CO-QL natively deals with such a situation.

- The language natively deals with spatial operations on geometries. This is an essential feature of the language, very useful for managing data with associated geometries, a more and more frequent situation in the Open Data world.

As far as the J-CO framework is concerned, we think that this section has shown some of the reasons why we decided to devise it.

- The J-CO framework is able to provide a unique environment to integrate data coming from different databases managed by different servers. This feature is essential to integrate data in a seamless way. Given this characteristics it could be a suitable framework towards querying distributed NoSQL databases in a distributed processing infrastructure such as Hadoop or Spark.

- Consequently, the J-CO framework can be considered a milestone toward a flexible framework for building polystore database systems for data science applications.

5. CONCLUSIONS

In this paper, we proposed an innovative framework, named J-CO, for integrating and transforming heterogeneous data sets in the form of collections of possibly geo-tagged JSON objects. The idea is to provide both analysts and geographers with a powerful tool that makes possible to perform complex analysis processes without writing procedural programs, but specifying transformation processes in a high-level way. The framework is founded on a high-level query language, named J-CO-QL, specifically devised to query heterogeneous collections of (possibly) geo-tagged JSON objects. Furthermore, the framework and the query language have been designed to retrieve input collections and to store output collections to several NoSQL databases in a seamless way, allowing analysts to easily integrate collections stored in different databases.

An example is illustrated. We show how it is possible to integrate geo-referenced information concerning quartiers and pharmacies of a city (we considered Milan, Italy) to create two new GeoJSON layers, one reporting quartiers with less than 2 pharmacies and one reporting quartiers with at least two pharmacies. Through the example, we introduced many J-CO-QL operators, briefly describing them. For a more detailed presentation, the reader can refer to [5].

The development of the J-CO framework is ongoing.

An important issue is to develop a suitable user interface that allows users to write query processes step by step, possibly inspecting the temporary collection and the intermediate results database and, if necessary, backtracking the query. Such a user interface is currently under development.

More ambitiously, we want J-CO-QL to meet the *data independence principle* by defining a shell layer framework of operators for easing the transformation of JSON objects transparently to the user. This is inspired by the concept of mediators, lightweight integration components, deemed to access sources on demand [39]. As defined in [39] "A mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications": sources are encapsulated by wrappers (which access their data sources in a transparent way to mediators) to present data in the form needed by a mediator.

A weakness of the current proposed operators is that users need to be aware of the structure of JSON objects, thus violating the independence of the language from data and forcing to user to be aware of the structure of the data. In our next evolution of the language, we aim at defining two layers of operators: the user-layer consisting of operators directly invoked by users; the hidden layer, consisting of operators automatically invoked by the user–layer operators, whenever it is necessary through the mediators to transform a JSON object to allow its comparison/join with another JSON objects having a different structure.

The final goal of the project is to define a powerful language suitable for integrating and transforming big data concerning territorial and geographical data sets, coming from heterogeneous sources with the less effort as possible to the final user, possibly supporting novel applications such as location-based queries [9, 10].

REFERENCES

[1] M. Armbrust, R.S. Xin, C. Lian, Y. Huai, D. Liu, J.K. Bradley and M. Zaharia, "Spark sql: Relational data processing in spark". In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 1383-1394). ACM. 2015.

[2] R. Battle and D. Kolas, "Geosparql: enabling a geospatial semantic web". Semantic Web Journal, 3(4), 355-370. 2011.

[3] F. Benitez-Paez, A. Degbelo, S. Trilles and J. Huerta, "Roadblocks hindering the reuse of open geodata in Colombia and Spain: a data user's perspective. ISPRS International Journal of Geo-Information, 7(1), 6. 2017.

[4] G. Bordogna, A. Campi, G. Psaila and S. Ronchi, "An interaction framework for mobile web search". In Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia (pp. 183-191). ACM.2008.

[5] G. Bordogna, S. Capelli, D.E. Ciriello, and G. Psaila, , "A cross-analysis framework for multi-source volunteered, crowdsourced, and authoritative geographic information: the case study of volunteered personal traces analysis against transport network data". Geo-spatial Information Science, 2017.

[6] G. Bordogna, S. Capelli, and G. Psaila, "A big geo-data query framework to correlate open data with social network geotagged posts," Proceedings in AGILE 2017 international conference, 2017.

[7] G. Bordogna, A. Cuzzocrea, L. Frigerio, G. Psaila and M. Toccu, "An interoperable open data framework for discovering popular tours based on geo-tagged tweets. International", Journal of Intelligent Information and Database Systems, 10(3-4), 246-268, 2017.

[8] G. Bordogna, L. Frigerio, A. Cuzzocrea and G. Psaila, "Clustering geo-tagged tweets for advanced big data analytics". In Big Data (BigData Congress), 2016 IEEE International Congress on (pp. 42-51). IEEE. 2016.

[9] G. Bordogna, M. Pagani, G. Pasi and G. Psaila, "Evaluating uncertain location-based spatial queries". In Proceedings of the 2008 ACM symposium on Applied computing (pp. 1095-1100). ACM. 2008.

[10] G. Bordogna, M. Pagani, G. Pasi and G. Psaila, "Managing uncertainty in location-based queries". Fuzzy sets and systems, 160(15), 2241-2252. 2009.

[11] G. Bordogna, M. Pagani, and G. Psaila, "Database model and algebra for complex and heterogeneous spatial entities," in Progress in Spatial Data Handling, pp.79–97, Springer, 2006.

[12] F. Burini, N. Cortesi, K. Gotti, and G. Psaila, "The Urban Nexus Approach for Analyzing Mobility in the Smart City: Towards the Identification of City Users Networking". Mobile Information Systems, 2018.

[13] S. Busse, R.D. Kutsche, U. Leser and H. Weber, "Federated Information Systems: Concepts, Terminology and Architectures", Forschungsberichte des Fachbereichs Informatik Bericht Nr. 99-9, Technische Universität Berlin, seen at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.71.8618&rep=rep1&type=pdf the 01-09-2018. 2018.

[14] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub, "The GeoJSON format," tech. rep., 2016.

[15] R. Cattell, "Scalable SQL and NoSQL data stores," SIGMOD Record, vol.39 (4), pp.12–27, 2011.

[16] D. Chamberlin, J. Robie, D. Florescu, "Quilt: an XML query language for heterogeneous data sources", e Int. Workshop on the Web and Data Bases (WebDB), 53-62, 2000.

[17] T.E. Chow, "Geography 2.0: A mashup perspective, "Advances in web-based GIS, mapping services and applications", pp.15–36, 2011.

[18] A. Cuzzocrea, G. Psaila and M. Toccu, "Knowledge discovery from geo-located tweets for supporting advanced big data analytics: A real-life experience". In Proceedings of MEDI-2015 Int. Conf. on Model and Data Engineering (pp. 285-294). Springer, 2015.

[19] A. Cuzzocrea, G. Psaila, and M. Toccu, "An innovative framework for effectively and efficiently supporting big data analytics over geo-located mobile social media," Proceedings of the 20th International Database Engineering & Applications Symposium, pp.62–69, ACM, 2016.

[20] C. Doulkeridis and K. Nørvåg, "A survey of large -scale analytical query processing in MapReduce", The VLDB Journal, 1-26, 2013 .

[21] J. Duggan, A.J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner and S. Zdonik, "The BIGDAWG polystore system". ACM Sigmod Record, 44(2), 11-16. 2015.

[22] J. H. Feng, Q. Qian, Y.G. Liao, G.L. Li, N. Ta and L.Z. Zhou, "Survey of Research on Native XML Databases", Application Research of Computers, 6, 1-7. 2006.

[23] U. Gandhi, "QGIS tutorial and Tips", Last updated on Apr 30, 2018. seen the 01-09-2018 at https://www.qgistutorials.com/en/index.html#. 2018.

[24] G. Gou and R. Chirkova, "Efficiently querying large XML data repositories: A survey". IEEE Transactions on Knowledge and Data Engineering, 19(10).2007.

[25] V. Goyal and D. Soni, "Survey paper on Big Data Analytics using Hadoop Technologies", Int. J. of Current Engineering and Scientific Research (IJCESR) ISSN (PRINT): 2393-8374, (ONLINE): 3(7), 2394-0697. 2016.

[26] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database", Pervasive computing and applications (ICPCA), 2011 6th international conference on, pp.363–366, IEEE, 2011.

[27] S.C. Haw and C.S. Lee, "Data storage practices and query processing in XML databases: A survey". Knowledge-Based Systems, 24(8), 1317-1340. 2011.

[28] D. Heimbigner and D. McLeod, "A federated architecture for information management". ACM Transactions on Information Systems (TOIS), 3(3), 253-278. (1985).

[29] G. Hubert, G. Cabanac, C. Sallaberry, and D. Palacio, Query Operators Shown Beneficial for Improving Search Results, in TPDL'11: Proceedings of the 1st International Conference on Theory and Practice of Digital Libraries . Sous la dir. de S.Gradmann, F. Borri, C. Meghini and H.Schuldt .T. 6966. LNCS., p. 118-129. Springer, 2011.

[30] L.A. Kurgan and P. Musilek, "A survey of Knowledge Discovery and Data Mining process models". The Knowledge Engineering Review, 21(1), 1-24. 2006.

[31] W. Litwin, L Mark and N. Roussopoulos, "Interoperability of multiple autonomous databases". ACM Computing Surveys (CSUR), 22(3): 267 –293. 1990.

[32] M. Lupp, OGC Web Services, Encyclopedia of GIS 2008 Edition, Editors: Shashi Shekhar, Hui Xiong . 2008.

[33] R. Meo and G. Psaila, "An XML-based database for knowledge discovery". In International Conference on Extending Database Technology (pp. 814-828). Springer, Berlin, Heidelberg. 2006.

[34] A. Nayak, A. Poriya, and D. Poojary, "Type of nosql databases and its comparison with relational databases," International Journal of Applied Information Systems, vol.5, no.4, pp.16–19, 2013.

[35] K.W. Ong, Y. Papakonstantinou, and R. Vernoux, "The sql++ unifying semi-structured query language, and an expressiveness benchmark of sql-on-hadoop, nosql and newsql databases," CoRR, abs/1405.3631, 2014.

[36] G. Psaila, "A database model for heterogeneous spatial collections: Definition and algebra," Data and Knowledge Engineering (ICDKE), 2011 International Conference on, pp.30–35, IEEE, 2011.

[37] H. Robin and S. Jablonski, "Nosql evaluation: A us case oriented survey," CSC-2011 International Conference on Cloud and Service Computing, Hong Kong, China, pp.336–341, December 2011.

[38] M.Y Vardi, M. Y., “A theory of regular queries”. In Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (pp. 1-9). ACM. 2016.

[39] G. Wiederhold, “Mediators in the architecture of future information systems”. Computer, 25(3), 38-49. 1992.

[40] T. White, “Hadoop: The definitive guide”, O’Reilly Media, Inc.", 2012.

AUTHORS

Gloria Bordogna is researcher at CNR-IREA (Italy). Her research activity mainly concerns the representation and management of imprecision and uncertainty within information retrieval systems (IRSs) database management systems (DBMSs) and Geographic Information Systems (GIS), soft computing.

Giuseppe Psaila is researcher and professor at University of Bergamo (Italy). He works on many topics concerning data management, such as data mining, XML processing, query languages and soft computing, information retrieval, Big Data and Open Data.

AUTHOR INDEX