‘Using data analytics for collaboration patterns in distributed software team simulations: The role of dashboards in visualizing global software development patterns’ 2016 IEEE 11th International Conference on Global Software Engineering Workshops

Georgios A. Dafoulas (Comput. Sci. Dept., Middlesex Univ., London, UK); Fatma C. Serce (Dept. of Inf. Syst. Eng., Atilim Univ., Ankara, Turkey); Kathleen Swigger (Dept. of Comput. Sci. & Eng., Univ. of North Texas, Denton, TX, USA); Robert Brazile (Dept. of Comput. Sci. & Eng., Univ. of North Texas, Denton, TX, USA); Ferda N. Alpaslan(Dept. of Comput. Eng., Middle East Tech. Univ., Ankara, Turkey); Victor Lopez (Fac. de Ing. de Sist. Computacionales, Univ. Technologica de Panama, Panama City, Panama); Allen Milewski (Dept. of Comput. Sci. & Software Eng., Monmouth Univ., West Long Branch, NJ, USA)

SECTION I.

Introduction

The work presented in this paper is ongoing for more than six years based on an initial research project funded by the US National Science Foundation with the participation of universities from US, Turkey, Panama and the UK. The original project focused on the investigation of software development teams working over different time zones. The project utilized student teams to observe communication, collaboration and coordination patterns. The project used industrial advisory roles to gain an understanding of the logistics involved in real GSD projects, as well as issues relating to the way individuals participate in GSD teams and fulfill different roles. The project funding completion did not disrupt the on going research, which involved further institution allowing the researchers to investigate different team structures, cultural differences, software development tasks and GSD coordination.

The research is predominantly based on the observation of pilots involving student groups organized in GSD projects including time zone differences, multi-cultural teams and the use of computer-supported cooperative work tools for communication, file sharing, data repository and collaborative authoring. In this paper we will discuss two pilots involving six universities, three from Palestine and three from Egypt. These institutions participated in a EU funded project developing a joint postgraduate degree in software engineering and their students were trained in various aspects of distributed teamwork and GSD.

At earlier stage of this work, emphasis was given on issues such as coding mechanisms for communication in GSD, gamification and the role of culture in distributed software teams. During the past couple of years, the research branched out towards the role of data analytics in GSD and how data visualization would assist in understanding patterns of communication and collaboration in GSD projects.

As we will discuss in the following pages, there are two main drivers for using data analytics in this project, (i) obvious benefits for learning teams, as instructors are able to identify potential issues with individual students or certain teams, (ii) evident benefits for facilitating performance management of software development activities.

The work presented in this paper is in line with existing work on tools supporting offshore software development focusing on the use of sociograms and the visualization of dependencies [5] [11]. Our work from several pilot studies focuses on understanding the inter-team interactions in a way similar to other works on awareness of inter-team development [17], human activities in software development and group awareness enhancing collaboration [24].

SECTION II.

Background on GSD

Following several years of research in GSD simulations including pilot studies with the involvement of several institutions, the authors had the opportunity to apply their findings as part of a knowledge transfer project in software engineering education. The common theme was the ability to comprehend the needs of a software engineering programme in terms of its design, deployment and delivery. It was important to identify the necessary mechanisms for quality assurance that could be implemented across all institutions, based on common practice in EU HEIs. In order to design the software engineering curriculum, all partners reviewed the state of the art in similar programmes across the world.

The ACM and IEEE guidelines for software engineering curricula were used as guidelines as well as the relevant literature [1] [2] [3] [6] [13] [14]. As part of the identified curriculum, the importance of GSD was identified early on. This led to the inclusion of virtual team topics (e.g. testing, requirements elicitation, user centred design).

It was necessary for the consortium to appreciate the special characteristics of GSD instructions for learning teams and in particular the student perspective [15]. This would allow educating students in key discipline topics with an understanding how GSD issues may affect the way core discipline practices may be applied. It was important to demonstrate the way information exchange and virtual team knowledge sharing takes places in such team structures [18]. It was also critical for the project to demonstrate how collaborative work should fit the needs of a GSD project [8]. This would involve a transformation in communication between team members as GSD project communication, seems to follow certain patterns [20] and assessing how task designs and the GSD scenario impact team behavior and communication [22].

Previous work on investigations of communication and collaboration patterns in software engineering learning teams [9] helped in identifying ways for collecting, filtering, analyzing and representing data from GSD simulations. The GSD simulations used in the pilot studies of the programme adapted the follow the sun development process model consisting of six sub-processes as suggested by Kroll et al [19]. The workflow patterns provided for any identified tasks were also designed in line with the proposed criteria for GSD workflow from the relevant literature [9].

SECTION III.

Investigating GSD Simulations

As mentioned earlier, this study is based on GSD simulations involving student teams, meaning that at least two universities would be involved in each pilot. The partner institutions were dispersed in such a way that time zone difference was a minimum of two hours and a maximum of eight. The international nature of the project meant that participants had to resort on virtual learning environments, wikis and video conferencing for all collaboration. Mainly communication was in the form of asynchronous exchange of messages, but at times synchronous communication was deemed necessary.

This research involved setting up more than twenty pilot studies over the past few years, at times involving four to six institutions and quite often more than ten GSD teams working in parallel. On occasions pilots reached or even exceeded 100 participants. As part of knowledge transfer initiative, it was decided to offer GSD pilot studies to developing countries, during a EU funded project. The scope of the project was to create a joint postgraduate degree delivered in parallel from six institutions in two countries, with the support of Higher Education Institutions from four EU countries. One of the primary objectives in the partner countries (i.e. Egypt and Palestine) was the creation of sufficient volumes of highly skilled software developers who could be eligible for future outsourcing and offshoring software development projects.

Therefore, the scope of the GSD pilot studies was to provide a suitable framework for setting up virtual teams and train participants in the necessary skills. This meant that both participating instructors and learners should gain an understanding of realistic virtual team projects. The framework used for GSD simulations focused on the following: (i) training instructors in the facilitation and coordination tasks required for GSD, (ii) providing realistic opportunities for virtual teamwork to participating learners, (iii) specifying the necessary elements for the required supporting infrastructure and (iv) designing a method for organizing virtual teamwork projects for GSD scenarios.

The first concern for preparing a GSD simulation involved the selection of suitable activities that would sufficiently cover certain aspects of the software engineering curriculum. For example the collaborative creation of conceptual design models, coauthoring of code and testing of code listings are such tasks. The next concern relates to the way participants would collaborate while undertaking certain roles. Clear roles should be identified and a specific protocol for collaboration would be required to control exchanges between individual members, but also the way teams would interact and solve specific problems. Next, the way communication would take place should be agreed, in terms of frequency, etiquette, timing, volume, nature of messages and tools used. The selection of tools that would be provided in the virtual learning environment available to the members of the GSD scenario should take under consideration the nature of the project, the collaboration needs, the communication exchanges that should be supported and any constraints. Finally the way in which work should be collated, presented and used for project reporting was an important aspect of each simulation. This work focused on gamification and its results are published in previous papers.

SECTION IV.

Setting Up the GSD Pilot Studies

As mentioned earlier, this research has conducted several pilot studies in the past, leading to sufficient experience in establishing a framework of good practice for setting up GSD simulations. As part of this work, a number of key areas have been identified, as follows:

Structure – focusing on team formation, it is essential to avoid too hierarchical structures and ensure that all members are treated equally according to the roles they fulfill and their classification.

Roles – focusing on the way tasks are assigned to individuals with emphasis on their prior knowledge, experience and skillset.

Instruction – focusing on the role of instructors and tutors which would range from facilitating communication between those members who are collocated, monitoring progress, checking issues relating to effective communication across sites, assessing performance indicators, ensuring interim deliverables are exchanged and assisting with technical problems.

Assessment – focusing on a number of issues relating to the method used for grading student progress such as mapping project performance to specific assessment criteria.

Mix – focusing on the way participants would be grouped together, addressing the need for heterogeneous or homogeneous teams, aiming for certain time zone differences, blending skillsets, and experience levels.

Scenario – focusing on carefully putting together a range of tasks that could be achieved by all participating institutions keeping in mind the curriculum taught, the timing of the pilot, any institutional constraints, the level of study of each participant and the core subject area.

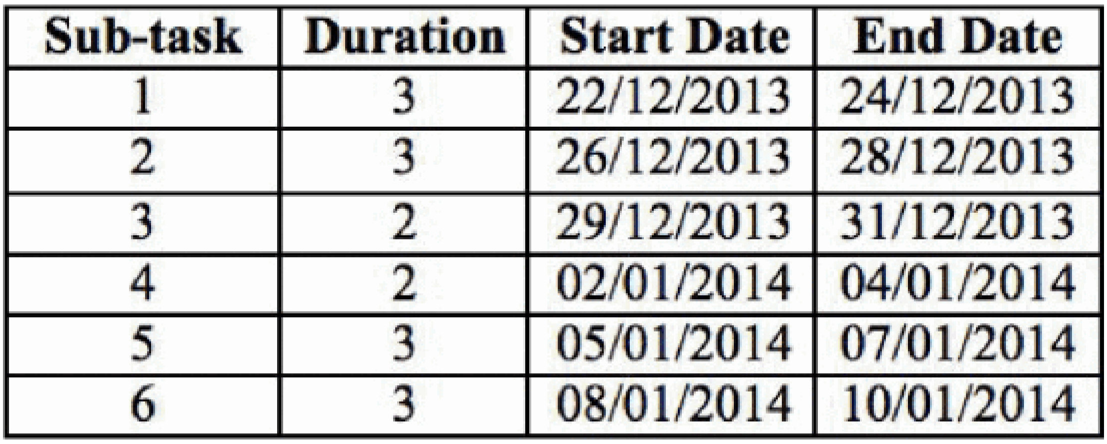
Data – focusing on the information that can be generated for the GSD tasks and ways for collecting, analyzing and displaying useful findings.

This paper is concerned with two pilot studies involving students from three Egyptian and three Palestinian Higher Education institutions (HEIs). The first pilot involved 2 teams, each consisting of 12 members. The pilot scenario focused on the implementation of a database for computer software installation and management. The tasks involved ERD modelling, database implementation and testing such as identifying stakeholders, specifying requirements, identifying conceptual entities, drawing the ER diagram, performing data normalization and developing the database schema. Emphasis on the first pilot was to monitor how team formation would affect performance; therefore the two teams were structured in different ways. Both teams consisted of six sub-teams, each assigned a specific task that once fulfilled it generated output that was necessary for another sub-team. The project was based on a sequential workflow, meaning that one team had to produce interim deliverables for the next till all six sub-teams completed their assigned workload. Both teams worked on the exact same task, while equivalent workload was assigned to all sub-teams. The key difference between the two teams was in the way their sub-teams were constructed, as in the first team each sub-team included two members from the same university, while in the second team sub-teams were composed of students from different universities. This meant that the second team involved sub-teams that required additional communication as students resided in different countries (one in Egypt and one in Palestine).

The data collected for both pilots was primarily classified according to GSD collaboration (activity), GSD communication (messages) and GSD interaction (patterns). The main patterns monitored included (i) generic team interaction pattern, (ii) interaction spread of team members across project timeline, (iii) user to team interaction by task, (iv) user to team interaction clustered by team and (v) user to team interaction clustered by task. As shown in table 1, each task would last 2–3 days, and required continuous communication from the members of each sub-team. The mode of communication followed was similar to the pair programming technique, with each pair working towards a set of clear tasks that should be completed by the interim deadline provided.

Table I. Task allocation and interim deliverable timings

Table I.



Sub-task	Duration	Start Date	End Date
1	3	22/12/2013	24/12/2013
2	3	26/12/2013	28/12/2013
3	2	29/12/2013	31/12/2013
4	2	02/01/2014	04/01/2014
5	3	05/01/2014	07/01/2014
6	3	08/01/2014	10/01/2014

The first key finding related to the constraints provided from the timing of the first pilot, as it was affected by the Christmas break, and weekend periods. This meant that the EU partners and the US team that coordinated the Redmine VLE servers had to provide additional support. The language barrier of some members meant that there were additional communication overheads for the sub-teams with distributed members. It also meant that there was the need for a communication etiquette that was introduced in the second pilot. The sequential nature of the scenario meant that only two sub-teams communicated at any time. Although this is expected in realistic scenarios, it did not maximize the exposure to virtual team communication for participants, leading to a different task structure in the next pilot. The pilot was based on a sequence of single hand-overs between sub-teams, significantly affecting the pattern of communication and collaboration nearer the interim deadlines.

The classification of contributions according to project milestones, clearly demonstrates how certain peak times gather the vast majority of entries to the platform. The main hand-over period dominated log-ins and use of the chat and forum as well as the file sharing facility. The second busier period was the kick-of date for the project.

The second pilot involved 4 teams, each consisting of 6 sub-teams. Each sub-team will be representing a different PCI and all sub-teams would have to work together throughout the scenario. In other words, all teams would have to communicate throughout the pilot's duration and there would be no idle periods for any teams as in the first pilot. The same number of members was assigned to each sub-team, meaning that there were identical human resources for each project. The distributed nature of the development project shifted to the communication between sub-teams. In order to address the communication barriers and coordination mix-ups of the first pilot, a series of rules were used to help students directing their messages to the appropriate recipient(s).

It was decided that three milestones would be identified for the project (i) design of the ERD, (ii) logical design and (iii) database population. The students were required to, compile a list of tables, specify a list of attributes per table, identify a list of relationships, create a list of constraints and finalize the database. The different structure of the second pilot allowed local teams to introduce a number of mechanisms for improving performance including (i) virtual groups, (ii) walk-talk pairs, (iii) graded assessment and (iv) mentors.

The tools used for collaboration, included wiki, chat, and file sharing, in support of various software development tasks. Furthermore, incentives included a detailed assessment schema based on the following grading: 50% for completing the task: 50%, 30% for communication (only when the Redmine platform was used and communication followed the specified protocol) and 20% for submitting a final report and completing all questionnaires. All three of the above assessment components were compulsory to achieve a pass grade.

SECTION V.

Using Data Analytics To Understand Collaboration Patterns

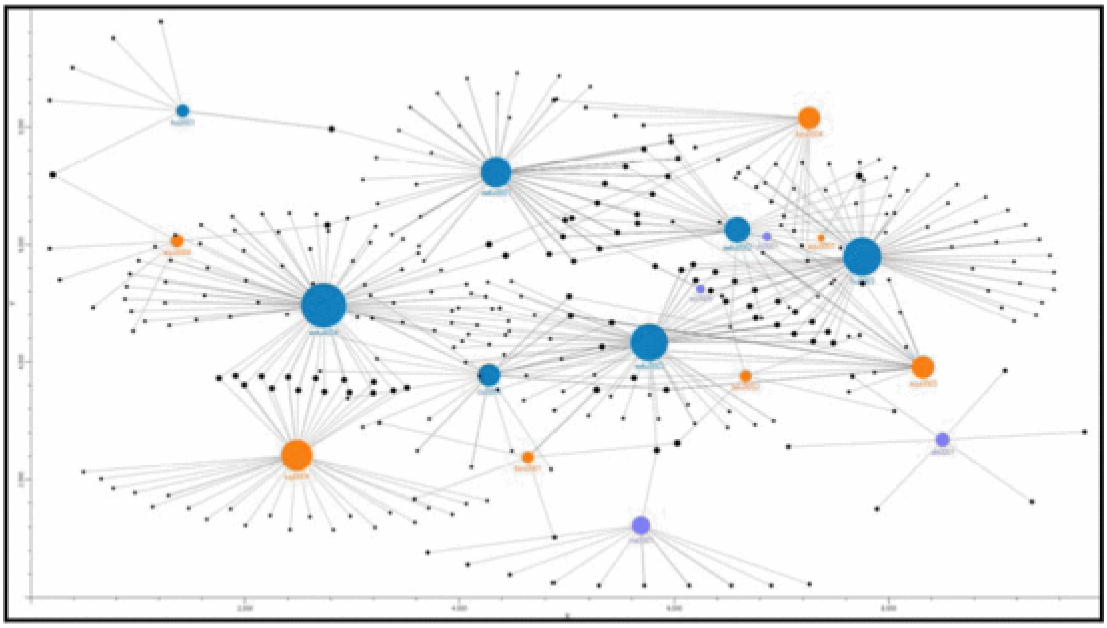
As mentioned earlier, our research focus has shifted over the past few years towards the investigation of data analytics for understanding collaboration patterns in GSD simulations. In particular, it was very interesting to observe the differences in communication and collaboration when changing certain aspects of GSD simulations, as we did in the two pilots discussed in the previous section. The main differences between the two pilots, included (i) the composition of sub-teams, (ii) the duration of sub-team involvement, (iii) the number of hand-overs and (iv) the introduction of a communication protocol.

We have used NodeXL for providing visual representations of GSD communication, collaboration and coordination. The scope of this paper is to share of the authors' views on the importance of such visualizations and the need for introducing a framework for using certain data analytics methods as the means for forecasting and also supporting decision making in the management of remote software development resources. It is important to state that such visual assistance was helpful for our instructors during the assessment of individual and team effort but could not provide a definite answer in relation to the quality of contributions. This is an issue that we were concerned in earlier phases of our project and we have published our views on message codification in GSD [20].

Figure 1 shows a generic team interaction pattern representing students as nodes of different size based on their interaction with other members. Each black node is a message, showing the number of messages that have been exchanged between team members as well as those that have remained unanswered. More work is needed to identify the reasons why certain messages do not have an answer. From the graphs there is no pattern emerging with regards to the messages without answers belonging to participants with high or low contribution. Parsing through the discussion logs, it appears that most messages can be classified either as ending messages to a thread (useful) or messages with confusing, or irrelevant content (useless).

Figure 1

Figure 1.



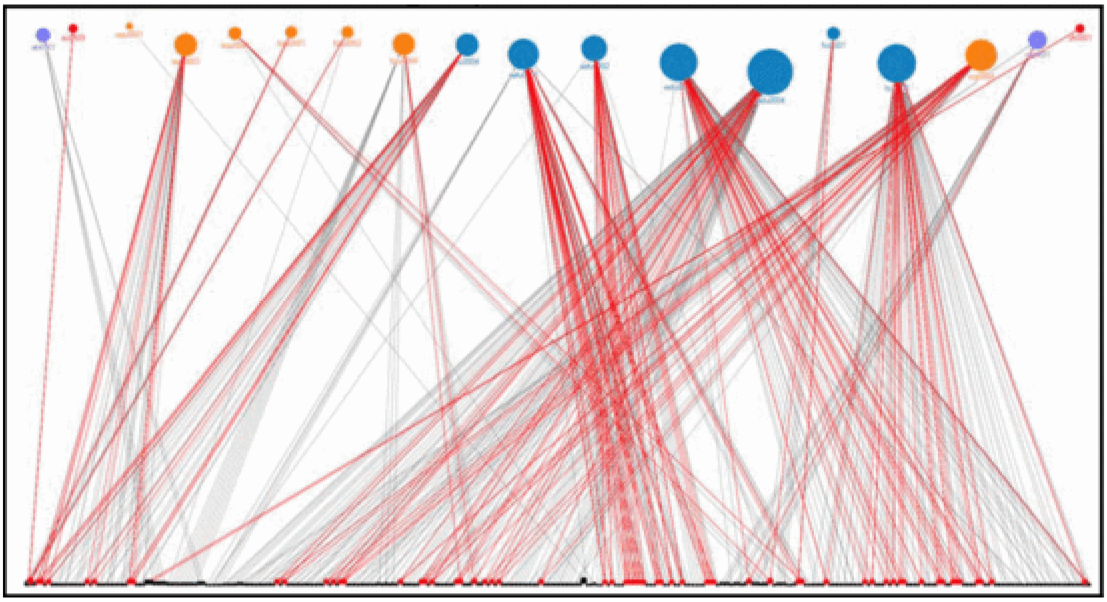
Generic team interaction pattern.

View All

Figure 1 shows how team members interact and can easily identify members who are more active than others. After manually eliminating noise (irrelevant messages) with a codification scheme, we can then use such visualizations during monitoring stages in order to assess leading individuals in GSD simulations. The size of each node demonstrates the volume of messages sent by each member, as well as communication between members of different teams during the hand-over tasks. What would be even more important is the ability to appreciate individual contribution across a project's duration. This can be seen in figure 3, showing the spread of individual interaction across a project's timeline (i.e. how team members contribute to Project tasks)

Figure 2

Figure 2.



Interaction spread across project timeline.

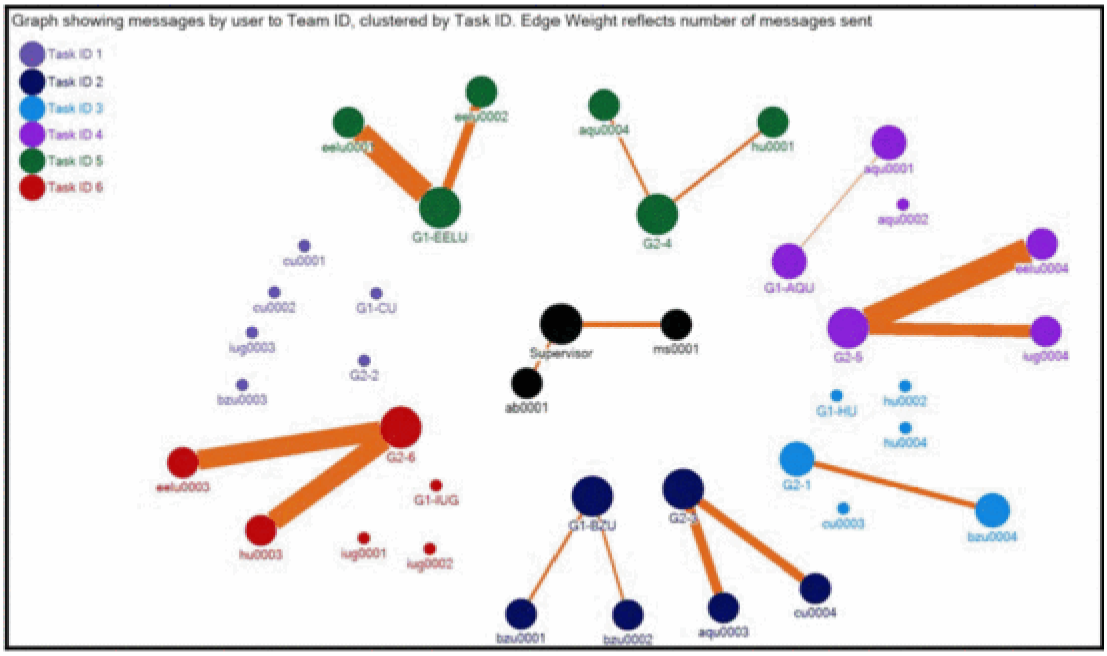
View All

Figure 2 shows the interaction spread of team members across project timeline, a particularly useful view of how each team member contributes at certain periods of the project. The angle of contribution becomes bigger when members are involved in several tasks. The team formation and role allocation of the two teams differed hence the wider spread of the blue team members.

Another interesting finding would be the contribution of each team member for specific tasks. This can be seen in figure 3 that shows the user to team interaction clustered by task, focusing on the demonstration of how certain tasks may affect the sub-teams responsible with respect to communication volume. Furthermore, the codification of messages could facilitate assessing whether the tasks affect the number of messages, or if this is a result of sub-team issues such as conflicts.

Figure 3

Figure 3.



User to team interaction clustered by task.

View All

There is a plethora of such visualizations for our collection of pilot studies. Depending on the focus of each GSD simulation, instructors could focus on assessing whether certain tasks are more complex than others, investigating communication patterns between certain individuals or even evaluating the collaboration in groups of certain structure or membership homogeneity. The next section discusses the way a dashboard can assist in visualizing GSD patterns.

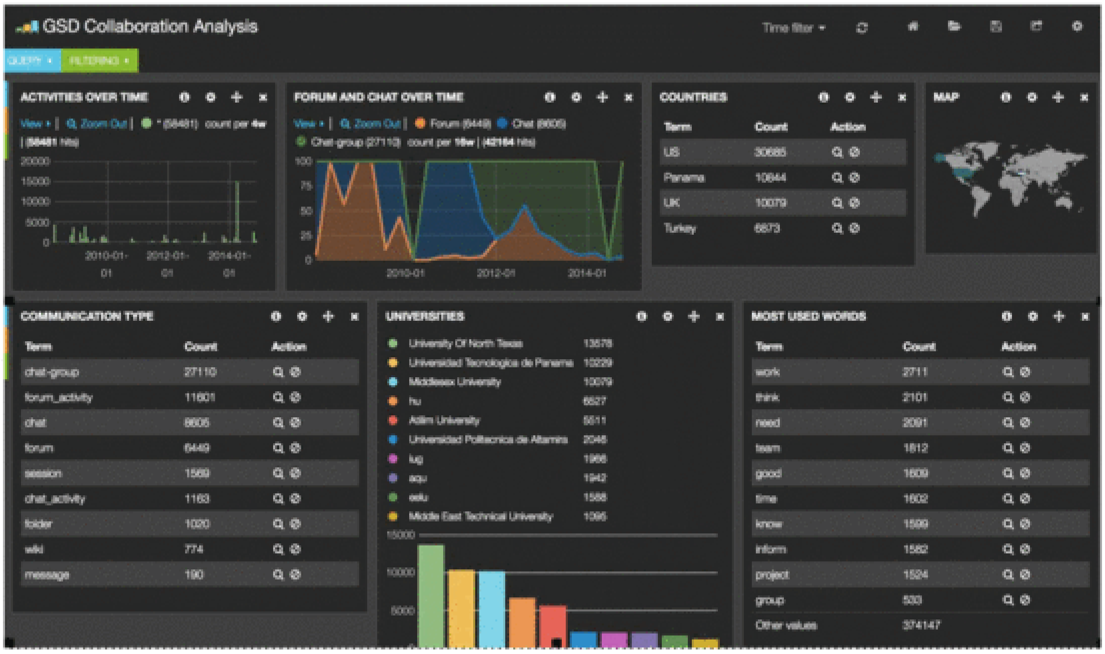
SECTION VI.

Introducing a Dashboard for GSD Visualisation

Using the Elasticsearch real-time search and analytics engine with the Kibana plugins, a proof of concept dashboard is created, connected to the data of the GSD pilot studies. The custom dashboard illustrated in figure 4 supports filtering communication activities that are stored according to time, country, university, group, gender, project task, etc.

Figure 4

Figure 4.


GSD Collaboration Analysis
Time filter
QUERY
FILTERING
ACTIVITIES OVER TIME
View | Zoom Out | * (58481) count per 4w | (58481 hits)
20000
15000
10000
5000
0
2010-01-01
2012-01-01
2014-01-01
FORUM AND CHAT OVER TIME
View | Zoom Out | Forum (6449) Chat (8605)
Chat-group (27110) count per 16w | (42164 hits)
100
75
50
25
0
2010-01
2012-01
2014-01
MAP
15000
10000
5000

COUNTRIES

Term	Count	Action
US	30685	🔍 ⊘
Panama	10844	🔍 ⊘
UK	10079	🔍 ⊘
Turkey	6873	🔍 ⊘

COMMUNICATION TYPE

Term	Count	Action
chat-group	27110	🔍 ⊘
forum_activity	11601	🔍 ⊘
chat	8605	🔍 ⊘
forum	6449	🔍 ⊘
session	1569	🔍 ⊘
chat_activity	1163	🔍 ⊘
folder	1020	🔍 ⊘
wiki	774	🔍 ⊘
message	190	🔍 ⊘

UNIVERSITIES

● University Of North Texas	13578
● Universidad Tecnologica de Panama	10229
● Middlesex University	10079
● hu	6627
● Atilim University	5511
● Universidad Politecnica de Altamira	2046
● lug	1966
● aqu	1942
● eelu	1588
● Middle East Technical University	1095

MOST USED WORDS

Term	Count	Action
work	2711	🔍 ⊘
think	2101	🔍 ⊘
need	2091	🔍 ⊘
team	1812	🔍 ⊘
good	1609	🔍 ⊘
time	1602	🔍 ⊘
know	1599	🔍 ⊘
inform	1582	🔍 ⊘
project	1524	🔍 ⊘
group	533	🔍 ⊘
Other values	374147	

Sample of a custom GSD dashboard

View All

The dashboard's panels are interactive, meaning that the visualization panels are dependent on each other. Therefore, once a country is selected on the map, the remaining panels show data from the specific location. An ‘Activities Over Time’ panel is used to show activities for certain time periods. This is particular useful for filtering activities according to different pilot periods, but may be also used for filtering according to pilot phases when we wish to focus on specific timestamps such as hand-over periods, and interim deadlines.

The ‘Communication Activity’ panel can be used for displaying highly ranked projects, students and universities for a range of activities (e.g. posts). Complex queries can be executed in order to display the contributions from certain team members over a selected period and for a particular task.

A particularly useful feature is the filtering of ‘most used words’, which is based on using synonyms analysis (i.e. WordNet engine) and can help assessing the most frequently used words from members in different countries during certain tasks. This feature can help investigating the use of selected keywords during key software development phases and at specific tasks such as requirements elicitation, conceptual modeling and brainstorming over design concepts.

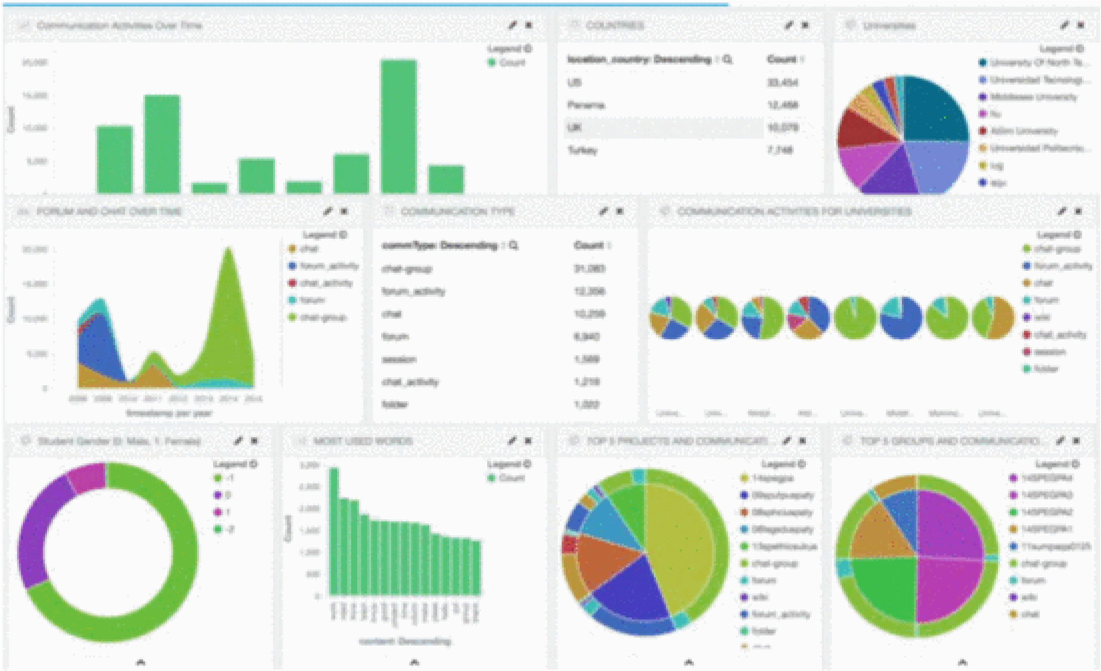
As the draft dashboard can be easily customized to fit the needs of different pilot studies, we can select different panels to display the necessary features of a specific GSD simulation. This means that we can provide a visual representation of each simulation and help participants in assessing specific aspects of collaboration, communication and coordination (see figure 5).

The interaction patterns visualized through the dashboard (e.g. message types, degree of contribution, frequency of keyword use) can help decision-making in GSD within an educational context. An empirical validation would help assessing the usefulness of the proposed approach. Therefore, further work involves another pilot involving the use of sensors to monitor stress levels of individual members involved in GSD scenarios. Emphasis will be also on evaluating the coordination of GSD resources during key GSD activities.

The dashboard can be used in a number of ways. Based on interaction data it is possible to provide accurate rankings of universities, teams and individuals for one or several pilot studies. Projects can be also ranked in terms of communication volume and frequency as well as the most active individual members. It is also possible to assess the popularity of different communication tools, types of interaction and communication activities over time.

Figure 5

Figure 5.


Communication Activities Over Time
Legend
Count
COUNTRIES
Universities
FORUM AND CHAT OVER TIME
chat
forum_activity
chat_activity
forum
chat-group
COMMUNICATION TYPE
COMMUNICATION ACTIVITIES FOR UNIVERSITIES
Student Gender (0: Male, 1: Female)
MOST USED WORDS
TOP 5 PROJECTS AND COMMUNICATI...
TOP 5 GROUPS AND COMMUNICATIO...

location_country: Descending	Count
US	33,454
Panama	12,466
UK	10,079
Turkey	7,748

commType: Descending	Count
chat-group	31,283
forum_activity	12,356
chat	10,259
forum	6,940
session	1,569
chat_activity	1,218
folder	1,022

Customizing dashboard panels.

View All

The dashboard is focused on assessing team interactions across different projects, while other dashboards focus on visualizing roles and emotions across tasks [23]. Dashboards are also used for viewing the evolution of participant interaction over time and annotate key events that occur along this timeline [4]. Previously, email graphs have been used for assessing the evolution of group cohesion [16]. Web applications provide the means to demonstrate individual and group learning analytics “showing conceptual and social network patterns, which we propose as indicators of meaningful learning” [10]. Further work in the field is focused on using social network analysis to understand patterns of collaboration and coordination in global software teams [12].

SECTION VII.

Conclusion

This paper emphasized the importance of using data analytics for collaboration patterns in distributed software team simulations and focused on the role of dashboards in visualizing global software development patterns, by using real time data from pilot studies. The work contributes on the provision of mechanisms and tools for learning teams, as instructors are able to identify potential issues with individual students or certain teams. The provided method for using data analytics in measuring key performance indicators and success criteria in distributed software teams can provide the means for facilitating performance management of software development activities.

References

[1] ACM Council, ACM Code of Ethics and Professional Conduct, October 1992, http://www.acm.org/constitution/code.html

[2] ACM/IEEE-CS Joint Task Force on Computing Curricula, Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, August 2004, http://www.acm.org/education/curricula-recommendations

[3] ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices, Software Engineering Code of Ethics and Professional Practice, Version 5.2, 1999, http://www.acm.org/about/se-code/

[4] Bakharia, A. and Dawson, S. 2011. SNAPP: A Bird'S-eye View of Temporal Participant Interaction, Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 168-173.

[5] Borici, A., Blincoe, K., et al. ProxiScientia, 2012. Toward real-time visualization of task and developer dependencies in collaborating software development teams’. Fifth Int. Workshop on Cooperative and Human Aspects of Software Engineering.

[6] British Computer Society, Code of Conduct & Code of Good Practice, 2004 and 2006, http://www.bcs.org/server.php?show=nav.10967

[7] Carmel, E., Espinosa, J.A.,Dubinsky, Y. 2010. “Follow the Sun” workflow in a global software development. Journal of Management Information Systems, 27, 2010, pp. 17-37.

[8] Dafoulas, G., Swigger, K., Brazile, R., Alpaslan, F.N., Lopez, V., Serce, F.C. Futuristic models of collaborative work for today's software development industry, IEEE, Proceedings of the 42nd Hawaii International Conference on Systems Sciences, January 5-8, Hawaii, USA, s. 1-10, 2009.

[9] Dafoulas, G. 2014. Investigating virtual teams: patterns of communication and collaboration in software engineering learning teams. In: ICERI 2014 : 7th International Conference of Education, Research and Innovation. Seville, November 17-19.

[10] De Liddo, A. Shum, S.B., Quinto, I., Bachler, M. and Cannavacciuolo, L. Discourse-centric Learning Analytics, Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 23-33

[11] De Souza, C.R.B., Hildenbrand, T., and Redmiles, D. 2007. Toward visualization and analysis of traceability relationships in distributed and offshore software development projects. 1st International Conference on Software Engineering Approaches for Offshore and Outsourced Development.

[12] Ehrlich, K., Valetto, G. and Helander, M. 2007. Seeing inside: Using social network analysis to understand patterns of collaboration and coordination in global software teams. International Conference on Global Software Engineering (ICGSE 2007), Munich, 2007, pp. 297-298.

[13] IEEE STD 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE Computer Society, 1990.

[14] Integrated Software & Systems Engineering Curriculum (iSSEc) project, Graduate Software Engineering 2009 (GSwE2009), Curriculum Guidelines for Graduate Degree Programs in Software Engineering, Verion 1.0, Stevens Institute of Technology.

[15] Filipovikj, P., Feljan, J. and Crnkovic, I. "Ten tips to succeed in global software engineering education: What do the students say?" in Collaborative Teaching of Globally Distributed Software Development (CTGDSD), 2013 3rd International Workshop on, 2013, pp. 20–24.

[16] Reffay, C. and Chanier, T. 2002. Social Network Analysis Used for Modelling Collaboration in Distance Learning Groups, Proceedings of the 6th International Conference on Intelligent Tutoring Systems, pp. 31-40.

[17] Sarma, A., Van Der Hoek, A. 2006. Towards awareness in the large. Int. Conf. on Global Software Engineering.

[18] Sole, D.L., and Applegate, L.M. Beyond Knowledge Transfer: A Typology of Knowledge Sharing Behavior in Virtual Teams. Proceedings of the European Conference on Organizational Knowledge, Learning and Capabilities. 2010.

[19] Kroll, J., Richardson, I., Audy, J.L.N. 2014. A Software Process Model for Follow the Sun Development: Preliminary Results. Global Software Engineeering Workshops (ICGSEW), 2014 IEEE International Conference on. Shangai. August 18-20.

[20] Serçe, F.C., Swigger, K.M., Alpaslan, F.N., Brazile, R.P., Dafoulas, G.A., Lopez Cabrera, V. Exploring the communication behaviour among global software development learners. IJCAT 40(3): 203-215 (2011).

[21] Storey, M.-A.D., Cubranic, D., Germán, D.M. 2005. On the use of visualization to support awareness of human activities in software development: a survey and a framework. SOFTVIS 2005, pp. 193–
202

[22] Swigger, K., Serçe, F.C., K., Alpaslan, F.N., Brazile, R., Dafoulas, G., Lopez, V. 2010. The Effects of Task Type on the Patterns of Communication Behaviors among Global Software Student Teams, International Engineering Education Conference, November 4-6, Antalya, Turkey (MEUK2010).

[23] Vivian, R., Tarmazdi, H., Falkner, K., Falkner, N. and Szabo, C. 2015. The Development of a Dashboard Tool for Visualising Online Teamwork Discussions, 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, 2015, pp. 380-388.

[24] Ye, E., Lev, A., D. Hiep, Q., Chang, L. 2009. SecondWATCH: a workspace awareness tool based on a 3-d virtual world. 31st Int. Conf. on Software Engineering, Vancouver, Canada, May 16–24, 2009, pp. 291–294