Collaboration Patterns in Distributed Software Development Projects

Igor Čavrak, Marin Orlić
University of Zagreb
Faculty of Electrical Engineering and Computing
Zagreb, Croatia
Emails: igor.cavrak@fer.hr, marin.orlic@fer.hr

Ivica Crnković
Mälardalen University
School of Innovation, Design and Engineering
Västerås, Sweden
Email: ivica.crnkovic@mdh.se

***Abstract*—The need for educating future software engineers in the field of global software engineering is recognized by many educational institutions. In this paper we outline the characteristics of an existing global software development course run over a period of nine years, and present a flexible project framework for conducting student projects in a distributed environment. Based on data collected from fourteen distributed student projects, a set of collaboration patterns is identified and their causes and implications described. Collaboration patterns are a result of the analysis of collaboration links within distributed student teams, and can assist teachers in better understanding of the dynamics found in distributed projects.**

***Keywords*-Distributed software development; Teaching; Patterns; Collaboration; Teamwork**

I. Introduction

Market demand for distributed software development knowledge and skills is high and increasing, as a large number of IT companies and research teams develop software either with, or for, foreign companies. The reasons for this demand range from economic to the fact that projects are becoming more multi-disciplinary, which makes it hard to gather experts with the necessary skills in one place.

There are many obstacles to successful distributed software development [1]. Some of the obstacles are technical, while others have roots in cultural and language differences. Teams often experience communication difficulties, starting even at very short distances [2], which cause irregular information flow and overhead in exchange of information [3]. Qualities important in teamwork, such as trust or cooperation, can be an additional challenge if various cultures are involved [4]. It is therefore important to prepare young professionals for work in a distributed environment and provide software engineering students with experience while they are still in the education process. Several courses dealing with global software engineering (GSE) exist, such as [4], [5], [6], [7]. Such courses can help students to recognize and analyze challenges through practical work with colleagues from another environment.

Providing such an experience in a sustainable manner to students in a higher education environment has been shown to be very difficult. Special attention must be devoted to the creation of a project framework in which students are exposed to characteristic obstacles in a controlled manner, to provide valuable and positive experience and not overwhelm them and jeopardize final project success. Thus, new, unobtrusive methods of distributed student project supervision and steering are necessary, allowing students freedom to experiment and create, and at the same time providing enough project insight and control from the teacher's point of view. One of the key factors of success or failure of distributed student projects is collaboration. In order to design and employ appropriate observation and steering mechanisms, characteristic collaboration patterns need to be studied, as well as their typical variations, causes and consequences to project success and their impact on student learning.

This paper describes the analysis of cultural differences, the project evaluation process and coordination patterns within distributed student project teams extracted from fourteen successful (but not problem-free) student projects, provided as part of the 2009 and 2010 Distributed Software Development (DSD) course. The analysis is based on qualitative and quantitative information collected from student questionnaires, as well as teachers' insight into project dynamics acquired from direct contact with project participants. This work constitutes a part of our larger effort to design a general framework for running distributed student projects and is based on our extensive experience gathered during nine years of running the DSD course between two European universities.

II. Related Work - Important Elements in GSE Courses

Various GSE courses are mentioned in the literature [4], [5], [6], [7], [8], [9], [10], [11]. In order to compare various approaches, we propose extracting a set of characteristics commonly found in GSE courses. In software engineering courses, which are typically project-based courses, course syllabi deal with common topics such as elicitation of requirements, development processes, and soft skills and technical skills related to both distributed and collocated teamwork. We focus on course elements specific to, or influential, in distributed courses.

Course positioning places the course at undergraduate or graduate level. Undergraduate courses generally have a stronger focus on skills adoption [9], [6], while graduate courses expose students to all project roles [10], [11]. This is similar to mixed-level courses [12], [13], which also included doctoral students. GSE courses require a wide set of student competencies and skills and are often positioned in the final years of corresponding education levels as capstone courses [6], [9], [12].

Diversity is a natural consequence of distribution, and students need to deal with colleagues from different schools, cities or countries. Through additional training [14], [7], diversity within teams can be *normalized* with respect to the balance of subteam sizes [9], [14], [11], nationalities [15], or level of knowledge and experience. Cultural diversity is often cited as one of the key issues in GSE [12], but it does not necessarily stem only from differing nationalities – a realistic setting resembling industrial environments mixes students of various backgrounds (engineering, business, arts) in a single team [11].

Time management adds an additional level of complexity, as students need to synchronize their personal agendas with project schedules due to different timezones [12], [9], differences in university schedules [16], [10], and even differences in timezone regulations [16]. Schedule of a course held across large time differences needs to account for participant fatigue and adapt the length of sessions [11].

The project framework of the course defines projects, roles, team structure and processes to be used. Projects are specified by customers who also give input on requirements. **The role of customers** can be taken by students [13], [12], lecturers [17], [14], [16], partners or clients from the industry [5], [17], [16], software engineering contests [6], [17], or software engineering projects [9]. The students must establish good communication with the customers and discuss project goals and requirements without assumptions. Gotel et.al. [12] have demonstrated that students, working in a different cultural setting than their clients, can form false assumptions and have trouble with ambiguities even in the relatively simple task of creating a library management system for a university.

The development process to be used in the project can be prescribed by teachers in full [5], [13], or in part [12], or can be selected by students [15]. The hierarchy and relationship between the teams is, however, mostly prescribed – e.g. equal peers [14], point-to-point [18], producer-consumer [10], or chains [12].

Motivation is a key factor in GSE courses, as the lack of personal contact can easily lead to alienation and demotivation of distant partners. Motivation problems, as well as the language barrier, are noted as the main source of difficulty for the students and a key reason for subsequent project failure [11], [14]. Initial student motivation for the course is higher for elective courses [14], [9]. Individual competition and personal curiosity can be a positive internal motivational force [18], [9], but care must be taken to create an atmosphere of a shared venture with equal, but different, contribution from all sides [12]. Participating in a real project that will be deployed to the benefit of its end users is a great motivating factor for students [9]. Assignments with deadlines early in the course encourage students to start collaborating early on and bond as a team [11], [14], [18].

Reflection and introspection give students valuable insight into their own performance, as well as into the effect their actions have on team progress. Students can be required to review documents or code [6], [14], implement the blueprints created by their remote partners [8], [10], or switch tasks in other parts of the process [7]. Soft elements of the course can also be reviewed – the importance of communication and management competence can be demonstrated in simple exercises [14], and students can assess their own presentation skills from video recordings.

The visibility of project progress, team dynamics and personal stance is maintained with weekly reports, minutes of meetings, blogs, message tracking (e-mail and instant), questionnaires, polls, and through informal talks [18]. Code and documentation review can reveal assumptions and misunderstandings between the teams [14].

Personal contact between remote students is removed, which hinders social bonding and non-mandatory (mostly informal) communication. To overcome the distance barrier, the course instructor can visit all sites to establish a hub of trust [11], [12], some or all students can travel to a central location to establish teams and start projects [9] or to transfer knowledge [16], or the project customer can be available on-site to assist with the project [16].

All elements mentioned here work together to support or hinder **collaboration** in student teams. In the following sections, we will introduce our DSD course in detail, and discuss the effect these elements have on collaboration.

III. DSD Course

A. Course Introduction

The Distributed Software Development (DSD) course is an elective course developed as a result of cooperation between the University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia (FER) and the University of Mälardalen (MDH), School of Innovation, Design and Engineering in Västerås, Sweden. Conducted successfully since academic year 2003/2004, DSD is a graduate level project-based course designed to offer students an experience of working on software development projects in distributed student teams, in geographically dispersed locations, and with several cultures, throughout all phases of a large, real-world software project. The course is conducted remotely, using communication technologies. In such an environment, students are faced with several issues common to global software development, and rarely observed in a homogeneous educational environment:

- *Differences in knowledge background* – the universities involved emphasize different elements of software engineering and computing education, which leads to unequal levels of knowledge;
- *Communication problems* – for most students, it is the first time they have had to work closely with their team members at a distance, using only e-mail, instant messengers and audio/video conferencing. Good project definition, specification and clearly defined interfaces between subsystems are crucial for project success;
- *Language difficulties* – most students are not native English speakers. Communication problems can arise, from misunderstandings delays in decision making, and even mild exclusion of team members (or the whole remote team) who can not communicate well in English;
- *Social and inter-cultural issues* – due to the diversity of nationalities of students at the Swedish university, it is common that the team comprises members from around five to seven different nations, mainly from Europe and Asia. Differences in educational background, views on work, life and communication habits can affect the team; team members may have a hard time accepting behavior patterns and customs not similar to their own, which can lead to lack of team spirit and trust.

B. Course Structure

The course consists of introductory lectures and a set of projects. In the introductory lectures, emphasis is put on the possible issues teams will encounter in the project work, and some of the ways to handle these problems. Topics range from an introduction to distributed software development, software configuration management, distance communication, and project management, through to lectures about soft skills such as cultural differences or presentation techniques. Guest lecturers from the industry are invited whenever possible, in order to convey first-hand experiences on project work in an distributed environment. The lectures take place only in the first five weeks, and in the remaining three months the course is completely focused on project work. During the practical part of the course, students need to present their progress on several occasions. The presentations are also conducted remotely, and include all sites.

C. Project Framework

Projects typically include 6–8 students, 3–4 per location, but minor deviations are possible in the case of significant disparity in the number of students enrolled at the two universities.

The DSD course project framework (Figure 1) defines four mandatory roles (customer, supervisor, project leader, team leader) and relations between those roles, but leaves enough freedom for student teams to define additional roles

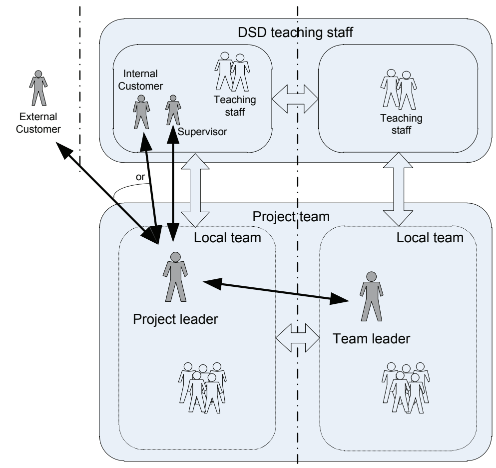

Figure 1. DSD Project Framework

and adjust their particular framework to the selected development process, supporting infrastructure and other project-specific requirements.

The Customer role provides a central authority in the project. The customer is responsible for defining general project requirements and accepting final project results. The customer can be internal, played by one of the senior teaching staff, or external, when the role is assigned to a collaborator from the industry or to a project proponent (when participating in a student competition, such as the SCORE competition at ICSE 2009 and 2011 [17]).

The project supervisor role is a central role in the framework. The supervisor monitors project progress with a focus on the development process rather than product, formally and informally interacts with both local and remote team members, and provides the technical support the students need. Support in solving organizational, social and inter-cultural issues is also very important. The supervisor role is typically assigned to a junior member of the teaching staff.

Each project team must be assigned a project leader and a team leader. Overall project coordination and communication with the customer and supervisor are the main tasks of the project leader. The project leader role is always assigned to a student who resides at the same location as the project supervisor (if the supervisor is one of the teaching staff at MDH, the project leader is a student at MDH). The team leader is always a student located at the other site from the project leader. The team leader's task is to coordinate local teamwork and help the project leader in overall coordination of project work.

The importance of cross-site communication and pivotal roles of project and team leaders in keeping communication channels open are repeatedly emphasized to students. Risks of both horizontal communication breakdown (lack of communication between distributed teams) and vertical communication breakdown (lack of communication between the supervisors and the teams) are analyzed and mitigation mechanisms explained. However, constant monitoring of project activities by project supervisors is always necessary.

Recognition, definition and assignment of other necessary project roles, as well as work organization (task assignment, policy definition, coordination and decision processes, conflict resolution mechanisms, etc.) are left to project teams to define. Some guidelines are provided by mechanisms ranging from mandatory publication of certain project policy documents to public discussions on project milestone presentations held by students.

The teaching staff selects project teams based on:

- Technical competencies and motivation for proposed projects as stated in the initial student questionnaire;
- A short test of students' knowledge and skills (optional);
- Past experience with a particular student (if available);
- Heterogeneity of resulting team (avoiding potential isolation of individual students based on cultural background);
- Technical competence of the resulting team.

Key student roles (project and team leader) are assigned by students, and the teaching staff does not intervene unless the team is unable to make a decision or there is clear evidence that the selected student is not suitable for the role. The teaching staff does not monitor assignment of other project roles.

The DSD project framework does not (in general) prescribe communication mechanisms or tools for communication among team members. However, to facilitate communication between remote teams it uses the following mechanisms:

- Mandates the use of the Subversion version control system according to the repository usage policy that must be published by the project team;
- Encourages production of "Minutes of Meeting" documents;
- Encourages the use of publicly available mailing lists or collaboration groups, and other means of traceable communication;
- Encourages the use of other computer supported cooperative work (CSCW) tools;
- Requires granting access to project supervisors to collaboration systems used.

D. Project Evaluation Methods

Project evaluation criteria are grouped into four sections, namely documentation quality, presentations, product quality and process quality.

Documentation quality criteria evaluate the existence, evolution and quality of a large set of mandatory documents students produce and maintain during the project. Of those documents, some are related to definitions of project policies students must adhere to during their work (code version control, communication, testing, etc.), some are of a general nature (weekly reports, etc.), and some are strongly dependent on the selected development process.

Presentation criteria address the quality (i.e. content and presentation) of a number of mandatory presentations and the involvement of all project members, regardless of their location or project role.

Product quality criteria evaluate the final product both externally (implementation of functional and nonfunctional requirements) and internally (design, coding quality, etc.).

The process quality section measures the quality of the development process used by the team to create the final product. Defined criteria address project planning, team organization, communication, tool usage, risk assessment, etc.

While most of the effort in evaluating project work lies with the project supervisor, the project customer is also required to evaluate the final product from her/his point of view.

The project supervisor or customer evaluates each criterion by assigning a number from 0 (non-existent) to 5 (excellent). The final project score is calculated as the weighted sum of all criteria, where the weighting denotes the significance of each criterion in the overall project effort. At the moment, around 50 criteria are defined, with weighting values ranging from 1 to 5. The second evaluation round is performed at the course level, where project performances are compared and evaluations equalized among supervisors. The project points awarded are scaled with respect to team size and presented to project leaders. Her/his duty is to convert the points into individual team member grade suggestions, where the sum of the suggested grades (1–5) cannot exceed the project points received.

The criteria and corresponding weighting differ between projects, predominantly to adjust for different development processes used and the set of optional project documents negotiated with the project customer. We have found that the variation in the maximum number of points projects can be awarded varies within a 5% margin, thus making comparisons between project evaluations sufficiently valid.

The evaluation method described provides a fairly objective quantitative measure of process and product quality. However, it does not capture all the properties of project teamwork we are interested in as teachers. Primarily, it does not reveal the internal project dynamics – work distribution, communication and collaboration patterns, information flow, individual effort, and tensions among team members. Such information is important in:

- Determining individual course grades based on both team and individual performance;
- Gaining insight into students' experience of the simulated distributed work environment; and

- Improving the simulated distributed work environment in future courses.

The qualities are assessed based on insight the project supervisor and customer gain during project work. The intensity and nature of supervisor interaction with the student team is strongly related to the amount of problems experienced by the team. In problem-free projects, the supervisor is predominantly concerned with development process issues and product quality. Problematic projects require supervisors to devote most of their effort to adjusting team organization or resolving personal issues between team members.

For the purpose of describing the overall characteristics of student projects, we have introduced qualitative measures which provide a general valuation of project work, including characteristics not captured by the quantitative method. A qualitative judgment is a combination of the quantitative values and the experience of the supervisors and teaching staff resulting from communication with students, and from listening to their presentations and demonstrations. According to the qualitative method, projects are classified as:

- *Green* – projects without problems or with minor problems;
- *Yellow* – projects with problems but still producing acceptable results; and
- *Red* – projects with serious unsolved problems and substandard results.

IV. Project Evaluation

The research conducted is based on a subset of data collected from 2009 and 2010 DSD course instances. Of a total of 19 projects, 14 were selected for analysis (of the remaining five projects, three projects were not distributed and two were distributed among three sites), having in total 90 students (40 at FER and 50 at MDH). While the environment at FER was mostly uniform (35 students from Croatia, two from Bosnia and Herzegovina and three from India), the MDH environment was multicultural consisting of students from Pakistan (14), India (13), Iran (4), Nepal (2), China (2), The Netherlands (2), Italy (2), France (2), Germany, Ukraine, Lithuania, Croatia, Eritrea, Bangladesh, Uzbekistan, Jordan and Kenya.

Most of the analyzed projects were successful: 8 were classified as green (51 students) and six as yellow (39 students). Table I presents the list of projects along with the number of students at each location, their quantitative (percentage of maximum points awarded), qualitative (color) evaluation results and the customer type. The number of students in boldface indicates the location of the project leader.

Figure 2 depicts the correlation between the quantitative and qualitative project evaluation results. Horizontal bars represent the percentage of yellow- or green-classified projects (qualitative evaluation) within the respective project score range, where the project score is determined by the quantitative project evaluation.

Table I
Project Evaluation

Name	MdH	FER	(% Points)	(Color)	Type
A	**4**	3	97	Yellow	Internal
E	4	**3**	96	Green	Internal
F	**4**	3	96	Green	Internal
C	3	**2**	95	Green	Industry
D	4	**3**	90	Yellow	Internal
B	**4**	3	89	Green	Industry
L	**3**	3	88	Green	Competition
N	3	**3**	88	Green	Internal
M	**4**	3	86	Yellow	Competition
I	4	**3**	84	Green	Industry
H	**3**	3	83	Yellow	Competition
J	**3**	3	76	Green	Competition
G	4	**2**	69	Yellow	Internal
K	**4**	3	63	Yellow	Competition

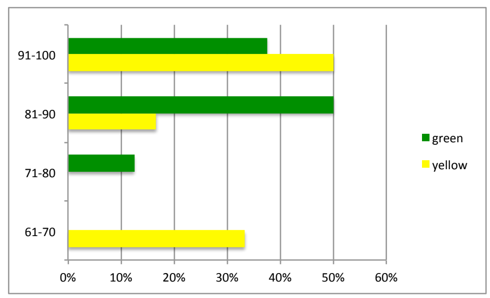

Figure 2. Qualitative (color) and Quantitative (range) Results

The projects with relatively low quantitative evaluation results (61–70%) were expected to score low on qualitative evaluation. However, there is a somewhat unexpected result in higher quantitative ranges, where green and yellow ranked projects are almost equally represented. The explanation for the observed results is as follows:

- Lower ranked yellow projects that struggled with internal team issues and could not compensate/adapt did not yield a quality process or end product. Thus, internal team issues were reflected in the end product and process.
- Higher ranked green projects had no serious internal team issues and could produce quality process and end-products.
- Higher ranked yellow projects managed to compensate internal team issues and managed to follow a process and deliver quality end products. Typically in these projects some of the students would take the largest burden, while other students would contribute to the project only marginally.

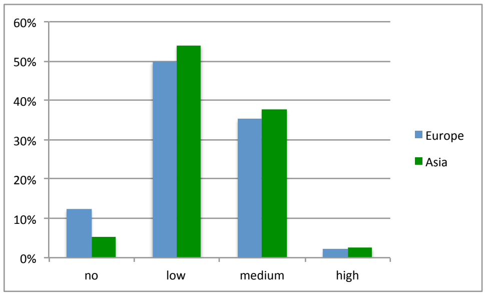

Figure 3. Cultural Difference Perception per Continent

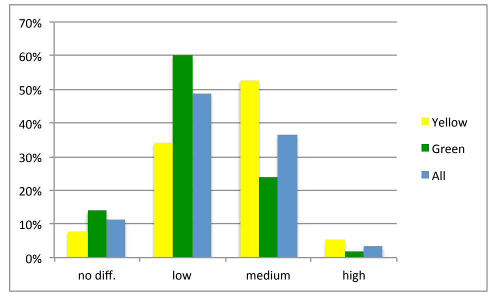

Figure 4. Cultural Differences and Qualitative Evaluation

No significant correlation has been found between project customer type and quantitative project evaluations, as well as between customer type and qualitative evaluations.

V. Cultural Differences

In order to investigate students' perceptions of cultural differences (CD) on the DSD course, we collect related information as part of the final questionnaire. Students express overall perception of cultural differences observed on their projects by classifying differences as *no*, *low*, *medium* and *high*. In addition to the general observations, students are requested to list all observed differences and rate their impact on project work (selecting an impact factor in the range 1–*low impact* to 5–*high impact*).

The distribution of overall CD perception has been analyzed with respect to participating students and project evaluation results. For every data set, a distribution of cultural difference observation (*no, low, medium, high*) is displayed with respect to the percentage of its occurrence in the overall observed population and disjunctive population subsets.

An analysis of overall CD perception with respect to student origin (Europe or Asia) is depicted on Figure 3. The distribution of observed CD shows negligible differences between groups, meaning there is no relevant bias among students from different cultures (the same result was obtained in analysis of potential bias between MDH and FER student groups). The figure does show some consistent minor differences in distribution where European students tend to have lower CD observations than Asian students. Part of the explanation for this could be that Asian students are relocated to the western environment and are aware of more differences than their western counterparts who mostly remain in their familiar cultural surroundings.

Figure 4 depicts the perception distribution with respect to qualitative project evaluations; the distribution for yellow and green project classes significantly differs for low and medium perceptions; perception ratings for green projects tend to display lower cultural differences, and ratings for yellow projects tend to display higher differences.

Observation of CD perception distribution with respect to the quantitative project evaluation results does not yield as clear a distinction as it does for qualitative evaluation. The general observation is that projects with higher score (81–91% and 91–100% point ranges) tend to have more right-skewed distributions, and lower score projects more left-skewed distributions.

The analysis conducted leads to the conclusion that the observation of cultural differences by DSD students strongly depends on the level of internal project tensions (qualitative evaluation) and is influenced by the ability of the team to compensate for those tensions and yield acceptable project results (qualitative evaluation). Student location or origin does not create a significant bias in the observation of cultural differences.

VI. Collaboration Patterns

The project framework used in the DSD course requires only two dedicated roles within a team (project leader and team leader), while allowing the teams to define other roles and to assign them to project members as needed. The main goal of prescribing the two roles above is to provide support for establishing and maintaining collaboration between distributed teams. The flexibility this approach provides is necessary for incorporating various development processes, customer and project types. However, the main drawback proves to be the teaching staff's inability to control emerged roles and collaboration channels due to rather low project work visibility, which is emphasized by the distributed nature of the projects. Such lack of control leads to the emergence of undesired collaboration patterns within project teams, influencing not only process and product quality, but more importantly educational goals the teaching staff is trying to accomplish.

Considering the relatively small size of project teams, there are two expected and, from an educational point of view, desirable general collaboration patterns: *virtual team* and *distributed team*. In a virtual team [19] the geographical distance between collaborators is neglected (all members collaborate independently of the collaborators' location), while in distributed teams the local sites are explicitly managed. Both patterns imply strong collaboration between the project leader and the team leader, but differ significantly in collaboration between other team members. Virtual teams should exhibit strong cross-border collaboration among smaller groups (or individuals) while strong local and weaker cross-border collaboration should be characteristic of distributed teams. The presence of significant collaboration within the local team is also very important at both locations.

A. Collaboration Analysis

To understand the consequences of different collaboration patterns emerging in student projects, we have collected and analyzed collaboration-related data found in the students' final questionnaires. In the final questionnaire the students are required to quantitatively rate (0–*no collaboration at all*; 5–*very intensive collaboration*) their involvement with other project members, both local and remote. Resulting from the collected data is a square collaboration matrix representing students' subjective perception of collaboration links within a project team.

In order to identify significant collaboration links within a team, collaboration matrix C is decomposed into a symmetric matrix CS and anti-symmetric matrix CA. Matrix CS presents an average collaboration weight between pairs of students within a project team and matrix CA contains a measure of "collaboration perception asymmetry". Matrix CS provides a basis for more advanced analysis of student collaboration, namely average ratings of local and distant collaboration, identifying isolated individuals or groups, etc.

B. Collaboration Density

The collaboration density analysis focuses on determining the proportion of significant collaboration links within the project team as a whole, within local teams, and significant links crossing the barrier between local teams. Collaboration density is defined here as a percentage of significant collaboration links with respect to the total number of possible collaboration links within the studied group. Additional analyses include average valuations of all collaboration links with further refinements taking into consideration student locations, influence of project and team leaders on collaboration valuations, etc.

Figure 5 contains the results of collaboration density analysis for all 14 DSD projects studied. The horizontal bars represent the density of collaboration links between distributed locations, among team members at MDH and team members at FER respectively. Significantly lower densities of distributed collaboration links are expected, compared to local densities. Interestingly, several projects also reveal lower local densities.

If compared with qualitative project evaluation results, it can be noticed that yellow projects (A, D, G, H, K and M) have mostly lower-than-average distributed collaboration densities and at least one lower local collaboration density. Unfortunately, this is not the general case. For example, project K, the lowest scoring project, has maximum local densities and average distributed density, characteristic of virtual team collaboration projects. Also, project M has all the characteristics of a distributed team project with strong local collaborations and rather weak distributed ones.

C. Pattern Visualization

Collaboration density analysis, as well as other numerical analysis based on collaboration matrix data, can provide an indicator of a general collaboration pattern or potential problems within a project. However, an additional tool is necessary to fully understand the details of collaboration within a student DSD project. For this purpose, a collaboration graph is used to visualize collaboration links within a project team by representing students as vertices and collaboration channels as edges. It is constructed from matrix CS by filtering out all collaboration weights less than three and representing collaboration intensity by different line thickness. Perception asymmetry data from matrix CA is added to edges as *less than* ($<$) symbols, pointing from a team member with higher collaboration valuation to a member with lower valuation. The number of symbols corresponds to a detected degree of asymmetry and also serves as an indicator of a member's valuation credibility. Figure 6 contains the collaboration graphs for projects A–F.

D. Identified Collaboration Patterns

Identified collaboration patterns are classified according to two criteria:

- Their focus on local coordination within teams or distributed collaboration between teams (local or distributed collaboration patterns);
- Their scope: individual or small number of team members, or whole teams (micro or principal patterns).

Star Pattern

A local principal pattern, this describes the organization of collaboration links within a local project team where there is, in general, only one central team member (usually the team or project leader) connecting all other local members into a functioning team. In Figure 6, project A uses the Star pattern in the local FER team. This pattern is neither positive nor negative with regard to the success of the project, although it clearly does not promote high collaboration density between local team members.

Mesh Pattern

A local principal pattern, this describes a highly interconnected local team with high local collaboration density. Most of the local teams in the DSD course projects exhibit such a local collaboration pattern due to mostly small local team sizes. In the project B, MDH local team exhibits such a collaboration pattern, excluding member MdH3.

Team Split Pattern

A local principal pattern, this indicates a split between local team members, where there is a clear collaboration gap. A local gap does not imply a disruption to project level communication, as split sub-teams can exhibit strong

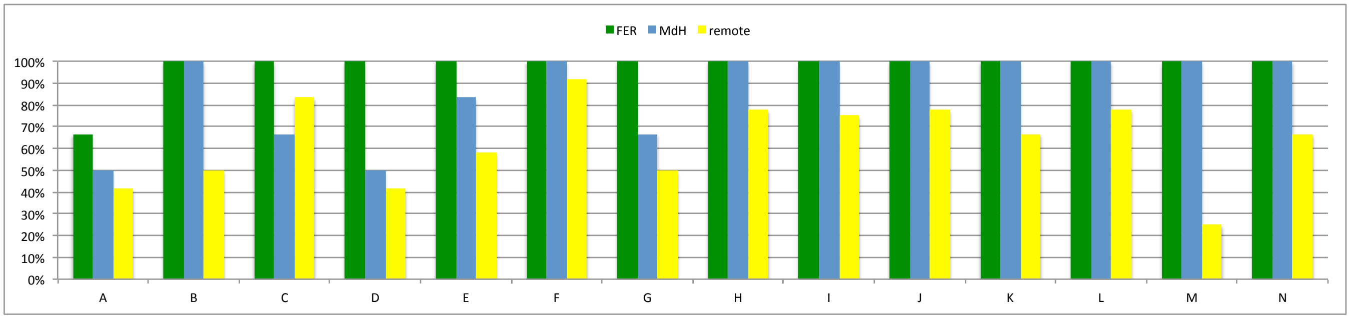

Figure 5. Collaboration Densities

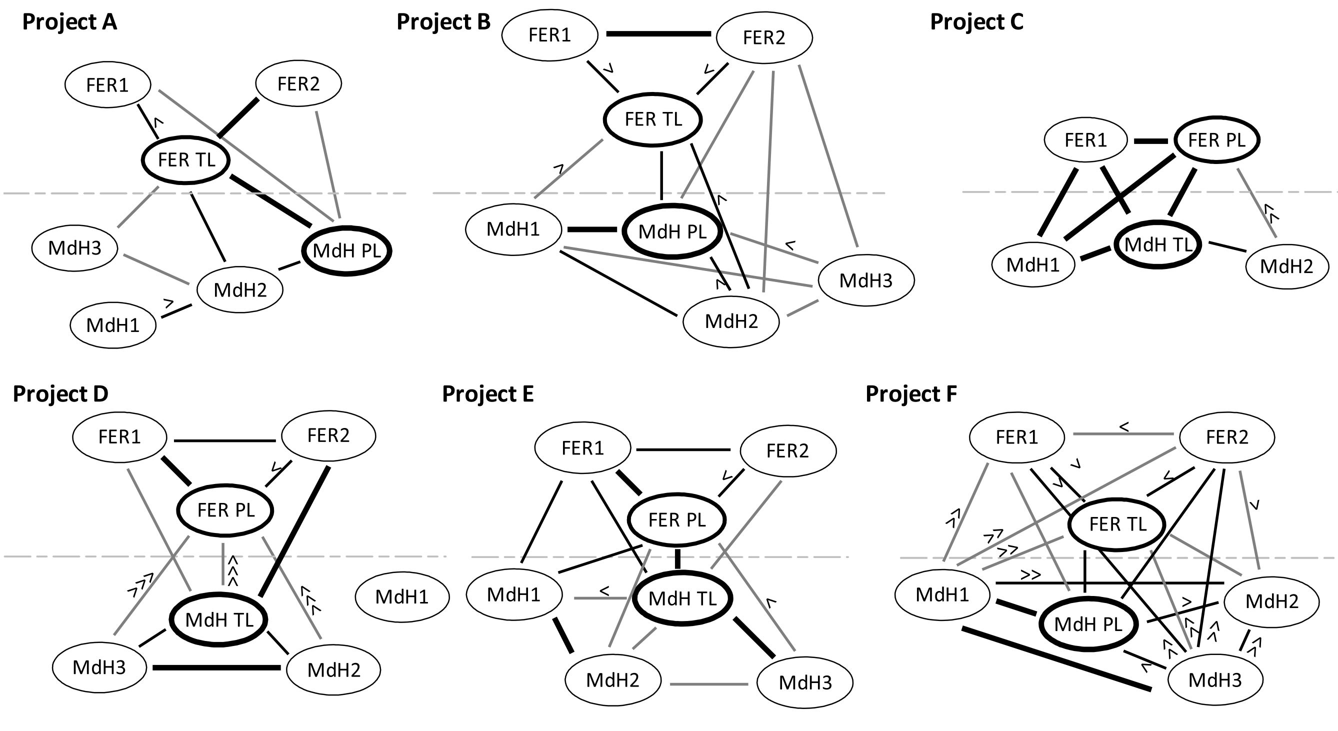

Figure 6. Collaboration graphs for projects A-F

collaboration with the remote project team. Reasons for team split pattern occurrence can be both positive and negative: project related (dictated by the product architecture) or socially related (cultural differences, personal issues, etc.). The project E, MDH team exhibits a Team Split pattern, where two sub-teams were formed, and one effectively joined the FER local team.

Sub-team Pattern

A local micro pattern, this identifies a local group whose internal collaboration links are stronger than links with their team or project leader, and with collaboration links crossing the team border almost non-existent. Such patterns are not negative by definition, but present a potential risk as they tend to isolate themselves from other project participants and ignore the need to closely collaborate with the remote location. The project D, MDH location exhibits a Sub-team collaboration pattern.

Loose Member Pattern

A local micro pattern, this represents an individual team member with weak collaboration links with her/his local team members and with almost no remote collaboration links. Such students usually do not possess enough technical skills to participate in the project equally with their local colleagues, and are marginalized and assigned trivial tasks. The rule of thumb is that they always maintain some kind of connection with their local team or project leader. The project B, MdH3 team member is an example of a Loose Member pattern.

Proxy Pattern

A local micro pattern, similar to the *Loose Member* pattern, although more radical. Students conforming to this pattern are attached to their team only by maintaining one significant collaboration link with a non-team leader, and typically sharing the same culture (cultural proxy variant of the pattern). Such students are completely detached from the project work and are effectively not part of the course. Project A, students MdH1 and MdH2 are an example of the Proxy collaboration pattern.

Island Pattern
A local micro pattern, this represents a complete lack of significant collaboration links of an individual member with any of the other project members. Lack of collaboration does not imply lack of contribution, just complete absence of proactivity and will to participate in common project activities. Project D, student MdH1 is an example of an Island pattern.

Virtual Team Pattern
A distributed principal pattern, this represents a project with strong collaboration links among a majority of project members. This pattern is typical of small projects (for example project C), but can be found in larger projects as well (project F, to some extent).

Project Core Pattern
A distributed principal pattern, this represents the existence of a collaboration *nucleus* within the project, where nucleus members are from both locations and are mutually connected by significant collaboration links. Project A contains a Project Core pattern encompassing the FER team leader, the MDH project leader and the team member MdH2.

Project Backbone Pattern
A distributed principal pattern, this denotes a single significant collaboration link between distributed teams. Such a pattern implies insufficient communication between distributed teams as all information is exchanged using only one communication channel and mediators, and is mostly administrative (the team and project leaders are usually connected by the backbone). Such a pattern can be a sign of a communication breakdown and can raise the alarm for potential problems during product integration. Another cause of the Backbone pattern can be a complete takeover of product development at one location, while the other location is assigned minor tasks, thus rendering strong collaboration unnecessary. An example of the Project Backbone pattern can be found in project D, between FER2 and MDH team leader projects members.

Triangle Pattern
A distributed principal pattern, this represents a constellation of significant collaboration links where a majority of the team members from one location have collaboration links established with only a team or project leader from the other location. Such a pattern indicates that at one location the majority of remote communication is led by a location *proxy*. The occurrence of this pattern can indicate that other team members are not confident enough in communication, as a result of cultural differences or insufficient technical knowledge.

Sand Glass Pattern
A distributed principal pattern, this represents a Triangle pattern existing at both project locations.

Remote Absorption Pattern
A distributed micro pattern, where team members from one location are *absorbed* into another location's collaboration structures, effectively leaving their original project location and decreasing collaboration with local team members. Project E, team members MdH1 and MdH2 are an example of a Remote Absorption pattern, where they effectively join FER location team management structures. The example presented is neutral with respect to educational standpoint and possibly beneficial for the project and process quality. An extremely negative effect of the Remote Absorption pattern could be observed in one instance (project M), where a project leader was absorbed into a remote team, leaving other local team members virtually out of the project.

Leadership Takeover Pattern
A distributed micro pattern, this represents a case where another team member from the same location effectively relieves a team or project leader of collaboration duties. This change in roles is not formal and is never communicated to the teaching staff. An example of a Leadership Takeover pattern can be seen in project D, where the FER2 team member relieved the FER project leader from coordinating the project. Such situations occur when the team or project leader is not technically competent as one of the team members, or there are strong tensions between the project leader and the remote team, where the best solution is to change the person in charge of the overall project coordination.

E. Pattern Incidence

Further analysis of the occurrence of patterns in student projects does not produce conclusive results for the existence of a single pattern or group of patterns responsible for poor project results. The presence of the Backbone pattern could indicate problems, as it is not found in green projects. It seems that the combination of patterns causing lower collaboration density and low student motivation is responsible for problems detected. As for the beneficial patterns, the Virtual Team pattern has only been found in green projects. The most frequently occurring distributed pattern is the Core pattern, while the most frequent local pattern is the Mesh pattern.

VII. Conclusion

In this paper we have presented a flexible project framework for conducting and evaluating distributed student projects on a DSD university course. Despite the advantages the framework provides, a careful study of its properties from the students' perspective is necessary in order to avoid the risks its flexibility can bring. Analysis of quantitative and qualitative project evaluations has revealed three types of student projects, differing on the existence of internal project issues and the team's ability to deal with them. The perception of cultural differences has also been proven to depend on the project issues. The main part of the paper has been devoted to the analysis of collaboration links within distributed student teams, resulting in the identification of a set of collaboration patterns. Patterns were analyzed with respect to their influence on project work and desired teaching outcomes.

ACKNOWLEDGMENT

This work is supported by the Croatian Ministry of Science, Education and Sport, under the research project ZP0361965 "Software Engineering in Ubiquitous Computing" and the project Hi5 funded by Swedish Vinnova agency.

REFERENCES

[1] B. Sengupta, S. Chandra, and V. Sinha, "A research agenda for distributed software development," in *Proceedings of the 28th international conference on Software engineering*. New York, New York, USA: ACM, 2006, pp. 731–740.

[2] J. Herbsleb and a. Mockus, "An empirical study of speed and communication in globally distributed software development," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 481–494, Jun. 2003.

[3] J. Herbsleb and D. Moitra, "Global software development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, 2001.

[4] S. L. Jarvenpaa and D. E. Leidner, "Communication and Trust in Global Virtual Teams," *Organization Science*, vol. 10, no. 6, pp. 791–815, 1999.

[5] N. Mullick, M. Bass, Z. Houda, P. Paulish, and M. Cataldo, "Siemens Global Studio Project: Experiences Adopting an Integrated GSD Infrastructure," *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, pp. 203–212, Oct. 2006.

[6] J. Favela and F. Pena-Mora, "An experience in collaborative software engineering education," *IEEE Software*, vol. 18, no. 2, pp. 47–53, 2001.

[7] B. Meyer and M. Piccioni, "The Allure and Risks of a Deployable Software Engineering Project: Experiences with Both Local and Distributed Development," *2008 21st Conference on Software Engineering Education and Training*, pp. 3–16, Apr. 2008.

[8] D. Damian, A. Hadwin, and B. Al-Ani, "Instructional design and assessment strategies for teaching global software development: A Framework," *Proceeding of the 28th international conference on Software engineering - ICSE '06*, p. 685, 2006.

[9] E. Stroulia, K. Bauer, M. Craig, K. Reid, and G. Wilson, "Teaching distributed software engineering with ucosp: the undergraduate capstone open-source project," in *Proceeding of the 2011 community building workshop on Collaborative teaching of globally distributed software development*. ACM, 2011, pp. 20–25.

[10] E. Keenan and A. Steele, "Developing a Pedagogical Infrastructure for Teaching Globally Distributed Software Development," in *Proceedings of Collaborative Teaching of Globally Distributed Software Development: Community Building Workshop (CTGDSD 2011)*, 2011, pp. 6–10.

[11] P. Gloor, M. Paasivaara, C. Lassenius, D. Schoder, K. Fischbach, and C. Miller, "Teaching a global project course: experiences and lessons learned," in *Proceedings of Collaborative Teaching of Globally Distributed Software Development: Community Building Workshop (CTGDSD 2011)*, 2011, pp. 1–5.

[12] O. Gotel, V. Kulkarni, L. Neak, C. Scharff, and S. Seng, "Introducing global supply chains into software engineering education," *Software Engineering Approaches for Offshore and Outsourced Development*, no. September 2006, pp. 44–58, 2007.

[13] D. Damian, F. Lanubile, and T. Mallardo, "Investigating ibis in a distributed educational environment: the design of a case study," in *Workshop on Distributed Software Engineering*, vol. 1. Citeseer, 2005.

[14] M. Nordio, C. Ghezzi, B. Meyer, E. Di Nitto, G. Tamburrelli, J. Tschannen, N. Aguirre, and V. Kulkarni, "Teaching software engineering using globally distributed projects: the DOSE course," in *Proceeding of the 2011 community building workshop on Collaborative teaching of globally distributed software development*. ACM, 2011, pp. 36–40.

[15] I. Bosnić, I. Čavrak, M. Orlić, M. Žagar, and I. Crnković, "Avoiding scylla and charybdis in distributed software development course," in *Proceeding of the 2011 community building workshop on Collaborative teaching of globally distributed software development*. ACM, 2011, pp. 26–30.

[16] B. Bruegge, A. Dutoit, R. Kobylinski, and G. Teubner, "Transatlantic project courses in a university environment," *Proceedings Seventh Asia-Pacific Software Engeering Conference. APSEC 2000*, pp. 30–37, 2000.

[17] I. Bosnić, I. Čavrak, M. Žagar, R. Land, and I. Crnković, "Customers' Role in Teaching Distributed Software Development," *2010 23rd IEEE Conference on Software Engineering Education and Training*, pp. 73–80, Mar. 2010.

[18] I. Bosnić, I. Čavrak, M. Orlić, M. Žagar, and I. Crnković, "Student Motivation in Distributed Software Development Projects," in *Proceedings of Collaborative Teaching of Globally Distributed Software Development: Community Building Workshop (CTGDSD 2011)*, 2011, pp. 31–35.

[19] R. E. Miles and C. C. Snow, "Organizations: New concepts for new forms," *California Management Review*, vol. 28, no. 3, pp. 62–73, 1986.