Supporting communication and cooperation in learning
The case of Data Science Minor

Pavel Okopnyi
Abstract

Supporting communication and cooperation in learning: The case of Data Science Minor

Pavel Okopnyi

Both teaching and learning Data Science are hard and require instrumental support. In the present thesis report, I describe a design intervention which was introduced to students and teachers of the Data Science Minor at Higher School of Economics (St. Petersburg, Russia). The intervention is a code sharing tool which is integrated with the Virtual Learning Environment - a set of software that teachers and students use as a primary instrument in the educational process. The code sharing tool, titled Gist, enabled teachers and students to share code and other digital artefacts, e.g., reports, which are used in the education. More importantly, it became a basis of a new emerging platform for educational applications.
Acknowledgements

I would like to express my gratitude towards all people who helped this project to happen:

• my friend and colleague Ilya Musabirov who provided enormous amounts of input during the process;

• my former supervisor and HSE’s deputy director, dean, and professor Daniel Alexandrov;

• my former students Alina Bakhitova, Denis Bulygin, Viktor Karepin, and Stanislav Pozdniakov who helped me with caring out the evaluation of software systems and solutions;

• DSM teachers, TAs, and students who are kind enough to tolerate systems which I develop for them;

• and, of course, my reviewer Iordanis Kavathatzopoulos who provided me with scientific guidance in the course of this project.

This report has been produced during my scholarship period at Uppsala University, thanks to a Swedish Institute scholarship.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>v</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Objective</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Report structure</td>
<td>1</td>
</tr>
<tr>
<td>2 Theoretical framework</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Communication in Education</td>
<td>3</td>
</tr>
<tr>
<td>2.2 ICT in Education</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Social Media in Education</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Personal Learning Environment</td>
<td>5</td>
</tr>
<tr>
<td>2.5 Social Affordances</td>
<td>6</td>
</tr>
<tr>
<td>3 Background</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Data Science Minor</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Virtual Learning Environment</td>
<td>9</td>
</tr>
<tr>
<td>3.2.1 Rstudio Server</td>
<td>9</td>
</tr>
<tr>
<td>3.2.2 Biostars Q&amp;A platform</td>
<td>11</td>
</tr>
<tr>
<td>3.2.3 VLE custom modules</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Private Learning Environment</td>
<td>12</td>
</tr>
<tr>
<td>4 Methods</td>
<td>15</td>
</tr>
<tr>
<td>4.1 Delimitations</td>
<td>15</td>
</tr>
<tr>
<td>4.2 User-centered design</td>
<td>15</td>
</tr>
<tr>
<td>5 Design and Evaluation</td>
<td>17</td>
</tr>
<tr>
<td>5.1 Project Goal</td>
<td>17</td>
</tr>
<tr>
<td>5.2 Initial Data Gathering</td>
<td>17</td>
</tr>
<tr>
<td>5.3 Design Idea</td>
<td>21</td>
</tr>
<tr>
<td>5.4 Development</td>
<td>23</td>
</tr>
<tr>
<td>5.5 First Implementation</td>
<td>24</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6</td>
<td>Adding Sharing Options</td>
<td>28</td>
</tr>
<tr>
<td>5.7</td>
<td>Interaction and Intelligence Workshop</td>
<td>29</td>
</tr>
<tr>
<td>5.8</td>
<td>Group Gists</td>
<td>30</td>
</tr>
<tr>
<td>5.9</td>
<td>Specialized Applications</td>
<td>31</td>
</tr>
<tr>
<td>5.9.1</td>
<td>Reports and assignment submissions</td>
<td>31</td>
</tr>
<tr>
<td>5.9.2</td>
<td>Presentations</td>
<td>32</td>
</tr>
<tr>
<td>5.10</td>
<td>Evaluation</td>
<td>33</td>
</tr>
<tr>
<td>6</td>
<td>Discussion and future work</td>
<td>35</td>
</tr>
<tr>
<td>6.1</td>
<td>Discussion</td>
<td>35</td>
</tr>
<tr>
<td>6.2</td>
<td>Future work</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td><strong>Bibliography</strong></td>
<td>39</td>
</tr>
</tbody>
</table>
List of Figures

3.1 Rstudio user interface .............................................. 10
3.2 Biostars Q&A platform user interface ......................... 12
3.3 Recently accepted answers and unanswered questions .... 13
3.4 PLE and VLE ......................................................... 14

5.1 Connecting components of the PLE ............................. 18
5.2 Code sharing via Social Media .................................... 19
5.3 DSM classroom ....................................................... 21
5.4 Students organize their space ................................... 22
5.5 Students organize their space (2) ............................... 23
5.6 First UI version ....................................................... 24
5.7 Social Media commentaries block ............................... 25
5.8 Addins button in Rstudio UI ..................................... 25
5.9 Share code option in Addins menu .............................. 26
5.10 New gist page ......................................................... 26
5.11 Named gist page ..................................................... 27
5.12 List of gists .......................................................... 27
5.13 Gist sharing options ................................................. 28
5.14 R script saved in users’ home directory ...................... 29
5.15 Gist UI: Share with group button ............................. 30
5.16 Gist report addin .................................................... 31
5.17 Gist report example ................................................ 32
5.18 Gist slides: fullscreen toggle button ......................... 33
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSCL</td>
<td>Computer-Supported Collaborative Learning</td>
</tr>
<tr>
<td>DS</td>
<td>Data Science</td>
</tr>
<tr>
<td>DSM</td>
<td>Data Science Minor</td>
</tr>
<tr>
<td>HSE</td>
<td>Higher School of Economics</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technology</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated and Development Environment</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System</td>
</tr>
<tr>
<td>PLE</td>
<td>Private Learning Environment</td>
</tr>
<tr>
<td>SM</td>
<td>Social Media</td>
</tr>
<tr>
<td>SNS</td>
<td>Social Networking Site</td>
</tr>
<tr>
<td>STEM</td>
<td>Science, Technology, Engineering and Mathematics</td>
</tr>
<tr>
<td>TA</td>
<td>Teaching Assistant</td>
</tr>
<tr>
<td>UCD</td>
<td>User Centered Design</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>VLE</td>
<td>Virtual Learning Environment</td>
</tr>
<tr>
<td>ZPD</td>
<td>Zone of Proximal Development</td>
</tr>
</tbody>
</table>
1 Introduction

The present work is aimed to provide design ideas to support, provoke and enhance educational cooperation and communication among students of the Data Science Minor in Higher School of Economics, St. Petersburg, Russia.

1.1 Objective

The purpose of this project is to propose and evaluate design hypothesis which aims to enhance, facilitate, and promote communication and cooperation among students.

In particular, the objective of the present project is to investigate and enhance practices which are associated with knowledge transfer between teachers and students and among students. These include, for example:

- general code sharing among peers,
- scaffolding of tasks and assignments,
- communication of instructions and usage examples.

The goal of the project is to support these practices by providing a technology or a system which would enable this kind of actions.

1.2 Report structure

This report continues as follows.

In chapter Theoretical framework I provide a theoretical background for the present work. Referring to literature, I demonstrate the importance of communication between students in education and the role of different ICT (including Social Media) in the learning process.
In chapter **Background** I describe the current state of the Data Science Minor (DSM) and technologies which support the learning process of DSM students. I also denote the problem which I am trying to solve in the current project.

Chapter **Methods** describes the methods that were used for design and evaluation of the solution.

In chapter **Design and Evaluation** I describe the design process and the evolution of the design hypothesis as it goes through the iterative process of design, development, and evaluation.

Chapter **Discussion and future work** finishes the report and provides discussion on the design, results of the intervention, and its impact on the educational process. It also provides information on ideas and plans for the future work.
2 Theoretical framework

As the addressed issue is revolving around communication, there is a need to understand what communicational tools and practices are common in the educational context, and what is their role in the whole learning process.

2.1 Communication in Education

Communication is one of the key elements of the learning process. Chickering and Gamson in [9] described seven principals which aim to enhance undergraduate education. Out of seven, three are directly connected to communication: “encourage contact between faculty and students,” “develop reciprocity and cooperation among students,” and “provide feedback.”

According to Vygotsky [36], social interaction plays a crucial role in education as it allows peer learning to happen. To describe the mechanics of the peer learning process, Vygotsky introduced the concept of Zone of Proximal Development (ZPD) - a “distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance, or in collaboration with more capable peers” [36, p. 86]. In other words, the ZPD denotes skills and knowledge that could be acquired by a student with (and only with) someone else’s help. This concept has a lot in common with the term scaffolding: “elements of the task that are initially beyond the learner’s capacity, thus permitting him to concentrate upon and complete only those elements that are within his range of competence” [39, p. 90]. As Dixon-Krauss summarises: “the teacher’s role is mediating the child’s learning activity as they share knowledge through social interaction” [12, p. 18]. The ZPD is an “area” in which effective teacher or instructor needs to operate in order to communicate knowledge to students. A failure to enter the ZPD
Chapter 2. Theoretical framework

means that the instructor does not convey knowledge in a way it can be comprehended by a student. Scaffolding can be then referred to as a technique which is used by teacher to get in the ZPD.

The quality and quantity of communication among students and between students and faculty are, not surprisingly, crucial for student’s retention and academic achievement. In one of the most common models of student retention, Tinto’s, suggests that significant role in student’s dropout plays one’s integration: academic and social. Academic integration refers to an enjoyment of studies and the very presence at the institution, while social integration means making friends and personal connections to peers and faculty [32, 13].

2.2 ICT in Education

Contemporary technologies allow for better and effective communication without requiring for all participants to be present in the same location. However, computer-mediated communication might be limited comparing to offline communication. For instance, Bradley et al. in [8] mention that some project teams have “better communication and performance only when interacting face-to-face.”

At the same time, a computer-mediated team or project group will take more time to establish trustful relationships between its members [38]. As it is reported in [35], this effect can lead to an establishment of negative attitudes towards the teamwork.

To facilitate the learning process, many institutions employ Learning Managements Systems (LMS) which provide instrumental support [37].

Both students and faculty use e-mail and other “classic” technologies for communication, new instruments are being introduced. E-mail is still the most popular way of communication between students and faculty at a Higher Education institution. However, modern technologies are gaining popularity, including Social Media. [18]
2.3 Social Media in Education

Social Media is an umbrella term which covers various technologies and services. Obar and Wildman in [25] compile definitions from different sources to result with four distinctive features to define this term:

- Social Media are built around WEB 2.0 ideology [26] which presume a user to be a participating creator rather than just a consumer of content;
- Social Media revolve around user-generated content;
- Social Media allow users to build and express both private and group identities;
- Social Media facilitate social networking.

Social Media are popular among both students and faculty. Moran, Seaman, and Tinti-Kane in [21] report that over 90% of all faculty in the U.S. employ Social Media services for personal needs and roughly 80% have experience using Social Media for professional needs. Social Media are even more popular among students and are being used by over 90% of college students for their personal needs [14, 33, 28, 11]. About 24% of students use Social Media in education. At the same time, the majority of students expressed an opinion that “it would be convenient” to use Social Media for studies [14].

Social Media are communicational tools and used by students mostly for direct communication via instant messaging functions. SM is also used to organise group work via text and video chats [17].

Even though SM is widespread, they are not often accepted as an educational tool. The most popular reason for that is the lack of privacy which SM employment implies. [21]

2.4 Personal Learning Environment

Social Media and other ICT and virtual tools employed in the learning process constitute Personal Learning Environment (PLE). PLE denotes user-centered approach, recognising learning as a personal activity and suggesting that a user (a learner) should be able to organise their learning activities including
communication with peers and teachers. PLE aims to narrow the gap between students as consumers and teachers as producers of education-related content [4, 2, 34].

PLE allows more flexibility for a learner than traditional LMS. It allows a learner to employ available ICT, such as Social Media, to shape their experience according to their learning goals. This flexibility results in a higher acceptance of technology comparing to course-oriented systems. [5].

People have a need to make sense of the world which they experience. In the case of students, there is a need to understand knowledge and skills which they try to acquire in their education. This “need to structure relevant situations in meaningful, integrated ways” is called Need For Cognition (NFC) [10]. NFC is associated with the higher acceptance of PLE approach among students [5] as well as the employment of Social Media as tools in the learning process [3].

PLE might be implemented as a single piece of software, trying to solve a variety of issues and problems which a learner might encounter. For instance, Bogdanov et al. in [7] describe an implementation of PLE Graasp which is a “multi-purpose collaborative platform.” It aims to satisfy various educational needs through extensions, widgets, and other virtual instruments. In other words, the platform can be “shaped” to match the needs of a user. Bogdanov et al. report that Graasp was evaluated in several universities and participating students were satisfied with the experience. One of the contributing factors was the possibility to control the level of privacy and to facilitate a student’s learning process independently.

### 2.5 Social Affordances

Software which facilitates learning process is characterised by certain properties allowing users to engage in social contact. These properties are called social affordances. Looking back at Gaver’s notion of “technology affordances” (see [16]) Kreijns and Kirschner in [19] define “social affordances” as properties of a Computer-Supported Cooperative Learning (CSCL) environment which “act as social-contextual facilitators relevant for the learner’s social interactions.”
Social affordances constitute social fabric. As an example, Kreijns and Kirschner present a design idea of a Group Awareness Widget (GAW) as a communication tool to be employed in a CSCL environment. GAW provide group awareness information such as messages by group members and notifications of various events (e.g., user login, user logout, new message).
3 Background

3.1 Data Science Minor

This work is originated from attempts to provide a toolkit to non-STEM students, who decided to study Data Science (DS), a compelling and technically sophisticated subject. The students come from various non-STEM backgrounds, from Oriental Studies to Sociology and Economics undergraduate programmes.

The Data Science Minor (DSM), which students can select out of several available options, is taught for two years (second and third years of studies) and contains several sections, e.g., Data Aggregation, Data Mining, Text Mining, and Social Network Analysis.

Currently (spring semester of 2017), there are 202 students in the first and 177 of the second year of education, three teachers, and 16 teaching assistants (TAs) in the DSM.

3.2 Virtual Learning Environment

The Virtual Learning Environment (VLE) is a set of tools which provide instrumental support for the DSM. It consists of three components: Rstudio Server, Biostars Q&A platform, and custom software modules. All software pieces are installed on a physical server which allows me and my colleagues to employ centralised authentication system and to manage users’ home and shared directories.

3.2.1 Rstudio Server

The main educational tool is Rstudio Server, a web-based integrated development environment (IDE) for R programming language [31]. Rstudio Server
and R language were chosen as the primary instrument for DSM for several reasons:

- R is one of the main languages/platforms for Data Analysis, especially for non-STEM specialisations;
- Rstudio provides a neat and full-featured IDE with various built-in tools suitable for both simple scripting and complex projects;
- server version of Rstudio allows for centralised management of user accounts and additional software, R packages, etc.

Rstudio interface (see Fig. 3.1) consists of four panels:

- file editor;
- interactive console to run R code;
- objects explorer which elaborates the contents of R objects present in the environment;
- file explorer.

![Rstudio interface](image)

**Figure 3.1: Rstudio user interface**

Rstudio provides a mechanism to include custom functions into its interface called “addins.” We employed this mechanism previously to provide better integration with Biostars Q&A platform and enhance students’ learning experience [22].
R programming language and associated technologies (including Rstudio) are famous for various options and software packages to produce different kinds of digital artefacts, e.g. graphics, tables, and reports. Rstudio user can employ the Rmarkdown language\textsuperscript{1} to easily author a report, presentations, or a manual which could be rendered as HTML, PDF, or a Word document. This feature is extensively used in the learning process in the DSM, as it enables knowledge exchange practices, including scaffolding.

### 3.2.2 Biostars Q&A platform

Rstudio Server is complemented with a Biostars platform, which is an open-source Q&A forum, similar to StackOverflow and other Q&A platforms (see Fig. 3.2). [27]

The platform was installed in an attempt to provide a place for hundreds of DSM students to communicate and exchange information. Both teachers and students use it to:

- ask R- and Rstudio-related questions, typically trying to identify emerged error;
- ask administrative questions, e.g., about the schedule, homework, midterms, finals;
- post links to websites and blog posts on related topics.

### 3.2.3 VLE custom modules

Two aforementioned systems are united with our custom software. Together, all three parts form a Virtual Learning Environment (VLE). This work aims to develop design ideas which would improve VLE in a way it would allow and promote cooperation and communication between students.

In [22] we suggested two design hypothesis to improve integration between Rstudio and Biostars platform which was implemented. In particular, a “Recent questions” screen was introduced in Rstudio UI (see Fig. 3.3), allowing a user to get an idea of the current Q&A activity and providing affordances to participate in it.

\textsuperscript{1}http://rmarkdown.rstudio.com/
Chapter 3. Background

We did not conduct any formal evaluation of the proposed interventions yet. However, observations suggest that users became more active after these interventions.

3.3 Private Learning Environment

It is not a surprise that students use other software, websites, and services in the learning process.

For instance, the most popular Russian SNS VK.com is widely used by students for communication. Google Docs\(^2\) and other online authoring tools are often used to organise different administrative and learning activities. Students also employ email and other services, which are not relevant to the current project.

Thus, the PLE of a DSM student consists of several components: VLE including Rstudio and a Q&A platform, Social Media, and other services (see Fig. 3.4).

Currently, there is a gap between components of the PLE: Social Media and VLE are not connected in any way other than by the student. These components are connected only by users’ practices as they employ different services.

\(^2\)https://docs.google.com
3.3. Private Learning Environment

in their learning process. Also, these components can differ from user to user, as some of the students (and teachers) might be very accustomed and proficient with ICT and sometimes employ even professional tools (e.g. git and other VCSs) while others might be not.

DSM students work and study in the virtual environment, and the artefacts they produce in the learning process are mostly digital. These include but not limited to:

- scripts, programs, and chunks of source code;
- tables, graphics, and other forms of results of code evaluation;
- notes and messages which students exchange in the learning process;
- written and generated reports which can include the aforementioned elements.
Chapter 3. Background

Figure 3.4: PLE and VLE
4 Methods

4.1 Delimitations

The challenge of the project is in the specific setting which was described in the chapter Background. There are several delimitations which constrain the design process.

The project is done for the HSE’s DSM and must be integrated into the VLE or be a part of the VLE. The solution should be integrated with Rstudio, VLE’s authentication system, and VK.com.

The project’s design should be easy to use for an untrained student who has little or no experience with programming in general and R programming language environment in particular.

The result of the design process should be implementable. It is important to implement and evaluate the design idea in the given setting.

4.2 User-centered design

The project is very limited due to its settings and background. At the same time, the number of potential users is limited to the number of DSM students and teachers, and it is crucial to deliver good user experience for the solution to be adopted easily. Thus, it is extremely important to understand the needs of stakeholders.

Eason in [15] (cited in [1]) identified three types of users:

- primary, who use the artefact;
- secondary, who use the artefact occasionally;
- tertiary, who do not use the artefact but are affected by it.
In the present project, I expect both students and teachers (including teaching assistants) to become primary/secondary users, but not tertiary. Thus, users can be easily involved in the development process.

The design methodology which implies a high level of involvement of users is called user-centered design (UCD). The UCD approach was introduced in 1980’s and developed by Donald Norman and Steven Draper in [24, 23]. In [23] Norman suggested principles for a design process which emphasize the needs of a user:

1. “State and action alternatives should be visible.”
2. “Conceptual model with a consistent system image.”
3. “Interface should include good mappings that reveal the relationship between stages.”
4. “User should receive continuous feedback.”

The principles are self-explanatory and correspond to various analogous sets of principles which could be found in, e.g., [30] and [29].

Following the User-centered design process in this project implies not only the investigation (and satisfaction) of users’ needs but also an affirmation of the final system to be obvious and transparent for the user.

Depending on project specifics, UCD implies various techniques to be involved in the design process, each suggesting high level of involvement of end-users (see, e.g., [30] and [1]). In the present project, I use following methods:

- Background interviews (in the form of contextual inquiry, see, e.g., [6]) in the beginning of the project to gather data on users’ needs and expectations;
- On-site observations during the early stages of the project to collect information regarding the environment in which the solution will be used;
- Interviews to gather data on user satisfaction of the project.
5 Design and Evaluation

5.1 Project Goal

The goal of the project was introduced in the section 1.1 Objective. In this section, I elaborate the project’s goal referring to concepts introduced in the chapter Theoretical framework and project specifics described in the chapter Background.

The goal of the project is to investigate and support practices related to code sharing and other activities which were mentioned in the section 1.1. The project is aimed to support the learning process by providing a tool which would enable scaffolding and the emergence of the Zone of Proximal Development (ZPD). The tool should enable the knowledge transfer between peer students by providing social affordances to share code examples, ask emerging questions regarding a function or a package, convey ideas on current assignments and tasks.

The tool should also serve as a bridge between the VLE and Social Media and other components of the PLE (see Fig 5.1). The tool should utilise the popularity of Social Media to provide affordances to students to utilise their friendship networks.

5.2 Initial Data Gathering

The practical work on this project started with several unstructured interviews with my former colleagues who are now teachers for DSM and my former students who work as teaching assistants (TA) for the DSM.

During interviews, respondents denoted the digital artefact sharing in general and code sharing in particular as one of the biggest technical challenges which teachers, TAs, and students encounter in the DSM. There are various ways to send another person a chunk of code:
• Copy the code and send it directly via Social Media or email.

• Send the code in an attached file via SM or email.

• Employ a service like Pastebin\(^1\) or Gist by Github\(^2\) by creating an online document with the code fragment and sending a link to it via SM or email.

Sending code as a message (on SNS) or email appears to be the easiest and the most popular method to share code among aforementioned. However, this way implies some drawbacks which often cause troubles for users:

• Code formatting (e.g., indentation) often gets crooked causing the code to become less obvious and clear for a reader (see Fig 5.2).

• SNS often automatically parse messages and interpret certain sequences of symbols as emojis.

• Code chunks are often stripped off their context and miss important information or commands which are required to run the given code fragment.

---

\(^1\)https://pastebin.com/
\(^2\)https://gist.github.com/
The second method (sending a file via email or SM) implies for both users (sender and addressee) to go through too many steps:

1. User A exports (downloads) a file from Rstudio server;
2. User A writes a message (email) to User B;
3. User A attaches the file to the message (email);
4. User A send the message (email);
5. User B receives the message (email);
6. User B downloads the attached file;
7. User B uploads the attached file to Rstudio server.
Chapter 5. Design and Evaluation

The aforementioned algorithm, according to interviews, is acceptably complex in the case when there is a need to share a file only once. In the case of continuous exchange of code chunks (e.g. during the project work), users are forced to repeat the process from the first to the last step in each iteration.

The third method of sharing which implies employment of third-party services. Thus these services are present in the PLE of some students but to a different extent. This method is practised unsystematically by both students and teachers and has several drawbacks. First, it requires a user to have an account in the system. Second, it implies a number of steps like downloading a file from Rstudio server and uploading it to the system which, similar to the second method of sharing, is too complex in the case of continuous exchange of code fragments.

As it was mentioned in the chapter Background, DSM students are not specialising in STEM, and DSM teachers try not to overload them with technical information or details. Thus, the employment of Git or other version control systems (VCS) was rejected as a possible solution to the problem as it implies mastering additional technical skills and tools which are not directly related to the subject of the DSM.

The context of use for the present project was investigated in a series of observations which were conducted in computer rooms at HSE during DSM classes.

In the classroom, there are several rows of computer desks, enough for each student in the class (see Fig. 5.3). Thus every student has own space that they can organise which they do.

As it can be seen in the Figures 5.4 and 5.5, students tend to put their bags and personal belongings on the table, designating their private space. Students often use mobile phones and tablets (see Fig. 5.4) or even laptops (see Fig. 5.5) in the classroom for various activities:

- To search for information regarding the topic of the current lesson. DS (and related terms, e.g., Data Mining and Machine Learning) is a buzzword and a popular theme on the Internet which results in numerous materials which could be and are used in the DSM.

- To communicate with classmates. Since verbal communication interferes with the lesson, it is a norm not to create a buzz in the classroom. Students employ SM applications to communicate with their
classmates silently, even if the addressee is in close proximity.

- To communicate with friends, not in the classroom. Sometimes students are getting distracted from the studies by SM applications and other ICT. It is hard to estimate the frequency and the effect of this kind of events. However, it will be incorrect to presume that it does not happen.

- Other reasons. As it was mentioned in the chapter Background, DSM students employ different applications and services for administrative and study needs. Code sharing is one of them.

5.3 Design Idea

In interviews, it was continuously proposed by teachers to employ a service similar to Pastebin or Gist by Github to share chunks of code. However, a third-party service could not be easily integrated into the VLE. Thus the employment of a third-party service will introduce a new component to the PLE which will not be connected to others.
In one of the interviews, it was suggested to design and develop the similar (to Gist by Github specifically) service or tool which would be integrated into both PLE and VLE. The tool should be integrated with Rstudio and allow a user to share code with minimal effort. The new tool is called Gist as it is an analogue to Gist by Github and have similar purpose and design.

Thus, my first design hypothesis is the following:

**Hypothesis 1 (H1):** The introduction of an integrated code sharing tool will enable and support teachers’ operations in students’ ZPD and the peer-learning process among students.

It was also decided to add a possibility for users to comment chunks of code via Social Media tools. Many SNS provide a possibility to include their custom blocks (“widgets”) into websites to enable users to comment or share pieces of information. It is hypothesised that this intervention would result in a change of students’ communication practices. Namely, students would move their discussions regarding current work (e.g., study projects) to the commentaries block.

**Hypothesis 2 (H2):** The introduction of Social Media components in the code sharing tool in a form of commentaries block will result in enabling and encouragement
5.4 Development

As I mentioned earlier (see section 4.1 Delimitations), the tool should be implemented or be easy to implement. Thus, considering the specifics of the user environment, it was decided to create a working prototype which could be used and evaluated in real-life scenarios by actual users.

Similar to the tool which we described in [22], Gist employs Addins \(^3\) functionality of Rstudio which allows introducing custom functions to Rstudio user interface.

The Gist system is a web application consisting of a server (backend) and a client (frontend, UI) parts. The server part of Gist is developed in Python, a programming language suitable for fast web development and testing.

---

\(^3\)https://rstudio.github.io/rstudioaddins/
Chapter 5. Design and Evaluation

The UI is developed with Bootstrap\(^4\) - a framework which is suitable for rapid prototyping and web development. Bootstrap provides a developer with a set of standard UI elements (e.g., buttons, input fields) which are recognisable and distinguishable.

To enable SM-related activities around the shared code fragments, I employed VK.com widget system\(^5\) which allows inserting widgets (e.g., commentaries block) into a website easily.

The development process was fast, allowing to implement and evaluate suggested features almost instantly. The Gist went through several iterations in which new features were added and tested with real users (teachers and TAs).

5.5 First Implementation

The first implementation of Gist had rather simple and minimalistic user interface (see Fig. 5.6). In the head of the webpage, there is a name of a code block which is provided by the author and the author’s name. Below is the code block.

\[\text{FIGURE 5.6: First UI version}\]

\[^4\]http://getbootstrap.com/
\[^5\]https://vk.com/dev/sites
Below the code block is the social media commentaries plug-in (see Fig 5.7).

![Code block]

**FIGURE 5.7: Social Media commentaries block**

The Gist system works as follows:

1. A user who wants to share code works in Rstudio and presses the Addins button to open Addins menu (see Fig. 5.8).

![Addins button in Rstudio UI]

**FIGURE 5.8: Addins button in Rstudio UI**

2. In the Addins menu the user chooses option “SHARE CODE” (see Fig. 5.9).

3. Rstudio saves the contents of the open file into hidden server directory. This step is completely transparent to the user.
4. Rstudio opens a new page in user’s browser. The address of the page (URL) contains a unique identifier (UUID, see [20] for more details) which refers to the saved chunk of code (see Fig. 5.10).

5. Now the user can share the code with other users by sending the link to the page via email, Social Media, or other ICT.

In discussion with DSM teachers, it was decided that it should be possible to assign a human-friendly name to the code fragment which will appear in the link instead of the UUID. Thus, the usage scenario can include several more steps, starting from step 4:

Figure 5.9: Share code option in Addins menu

Figure 5.10: New gist page
5. User enters the name of the code fragment and presses the “check” button to confirm the input (see Fig. 5.10).

6. Gist system applies the name to the code fragment. The user is being redirected to the new page which has the code fragment name in its address.

7. Now the user can share the code with others using the link with the code fragment name in it.

To get access to all code fragments owned by a user a page was developed with a “list of gists” (see Fig. 5.12). The list contains names or identifiers of each code fragment belonging to a user. The list is public and available to every user of the VLE.

Gists by paul

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>rscript_example.R</td>
</tr>
<tr>
<td>c6757c6a-3cc3-11e7-8d17-005056a4b82</td>
</tr>
<tr>
<td>tc:nixterm.Rmd</td>
</tr>
<tr>
<td>test_report.Rmd</td>
</tr>
<tr>
<td>test_new_1.R</td>
</tr>
</tbody>
</table>

Figure 5.11: Named gist page

Figure 5.12: List of gists
5.6 Adding Sharing Options

During brief evaluation with one teacher, it was suggested to add sharing options or instruments which would eliminate the need to manually copy the the page address (URL) from the browser’s address line and send it.

First, a “copy link” button was introduced (see Fig. 5.13, button 1). This button allows a user to copy the link to the page by clicking on it. After that, the user is to send the link to the addressee via SM or other ICT. Thus, this intervention eliminates the need to highlight and copy the page address in the address line of the browser.

![Gist example image](image)

**Figure 5.13: Gist sharing options**

Second, since the Gist system is a part of VLE, it was decided to provide a possibility to save code fragments as a file in users’ home directory on the server. By pressing the button “Save to home directory” (see Fig. 5.13, button 2) user saves the code fragment as a file to a specific directory which is accessible from Rstudio user interface (see Fig. 5.14).

Third, it was decided to employ a “Share” button provided by SNS VK.com. “Share VK” button (see Fig. 5.13, button 3) opens the dialogue window which allows a user to send the link to the Gist page to one or many friends on VK.com.

Last intervention implies the employment of R language function `source` which loads and interprets the contents of external files in the current R environment. To support this action, a special function was introduced and made available to a user through Gist UI. By pressing a “Source” button (see Fig. 5.13, button 4) user can load and run code from an external file.
5.7 Interaction and Intelligence Workshop

The Gist system was tested during the “Interaction and Intelligence 2017 workshop” which was organised together with DSM teachers and TAs. The workshop was attended by 25 participants who were divided into five teams. Students spend a week trying to create a prototype of an intelligent application which would rely on Social Media data. Evaluation of the Gist was not the sole (nor the main) purpose of the workshop. During the workshop, I tried to validate the Gist system as a working technology and employed workshop participants as testers who used Gist system to share code examples with their teammates. Several minor errors occurred during the testing and were fixed upon discovery.

During the workshop, students were encouraged to use Social Media commentaries block (see section 5.5 First Implementation) to discuss the projects. However, this method of discussion was merely adopted, and students were keen on employing private messages and real-life conversations for their discussions. This block was also used as a voting tool to determine the best project of the workshop.

6http://mlsoc.io/interaction2017
Besides some technical errors, no other issues were revealed at this point. Thus it was decided to employ the Gist system as an instrument in the ongoing DSM course.

5.8 Group Gists

After the Gist system was introduced to DSM students and used in practice, various ideas on further development emerged. Expressed in the series of interviews and conversations, some of the ideas were implemented while others were postponed or abandoned mostly due to technical reasons. One of the ideas was to turn Gist into an application which would facilitate group work.

The Gist system is designed as a user-centric system. Every code fragment has its author/owner who’s name appears in the link to the page containing the code. To enable group work, it was suggested to enhance Gist system by allowing students to share code fragments with their peers in study and project groups.

![Gist UI: Share with group button](image)

**Figure 5.15:** Gist UI: Share with group button

To share a code fragment with a group a user can push a “Share with group” button in the Gist UI (see Fig. 5.15) which brings up a modal window with a list of study and projects groups which user is a member of.
5.9 Specialized Applications

As the Gist system was employed in learning activities, different other applications were suggested.

5.9.1 Reports and assignment submissions

Originally, Gist system supported only one type of digital artefacts: source code. It was suggested by teachers that the Gist system should also allow a user to share generated reports (which is another common type of digital artefacts in DSM, see chapter Background for more details on digital artefacts).

![Figure 5.16: Gist report addin](image)

To share a report, a user must choose a “SHARE REPORT” addin in Addins menu (see Fig. 5.16) while having an Rmd-file (Rmarkdown source file) open in the editor. The system will render the file and redirect the user to its page (see Fig. 5.17). The page contains the output of the render process (rendered report) which can include tables, graphics and other elements. It also includes the source code of the report and a Social Media commentary widget. Teachers of DSM employed this application. The “gist report” is now a common form for an assignment submission. The students working on an assignment is asked to produce a generated report which would include both
source code and the results of its evaluation. The link to the report could be sent via email or other ICT. Thus the teacher would see (and grade) not only the results (e.g. an output of a statistical model) but also the code which student employed.

It was also decided to implement the submission system which would manage hand ins and allow students to submit an assignment directly from the Gist system UI. This system is still under development and testing.

5.9.2 Presentations

As it was mentioned in the chapter Background, Rstudio allows its users to author not only reports but also presentations which are used by DSM teachers and TAs for lectures. It is also often a requirement for students to make a presentation at the end of a study project.

Since sharing of rendered documents was already implemented to allow users to share reports, it also became possible to share a presentation via Gist.
5.10 Evaluation

To evaluate the adoption of the Gist system I analysed its logs which were recording for three months during the spring semester 2017.

223 users used the Gist system, 192 out of which have uploaded at least one code fragment. In total, 1379 code fragments were uploaded, out of which 269 were named according to instruction in the section 5.5 First Implementation.

Most of the code fragments were not named. Observing students, I noticed that in some cases they tend to enter a name but do not press the “check” button to confirm the action (see Fig. 5.10).

In total, code fragments were viewed by users nearly 8000 times. The view count is distributed according to a power-law: only 35 code fragments were
viewed more than 20 times while 345 were never viewed by anybody except the authors themselves.

The content analysis of a randomly selected sample of code fragments showed that users tend to use the Gists systems for various purposes. Besides sharing and assignments submission, some users seem to employ the Gist system as a “notebook”, saving code and text fragments.

The SM commentaries block appeared to be not in demand, as only a few commentaries were posted during the period of the project (except for commentaries which were posted during the workshop, see section 5.7).

The lack of adoption of the SM commentaries block can be explained in two ways. First, students might not want to expose their private identities from Social Media to public in the VLE (see section 2.3 and [21]). Second, students might treat public discussion and criticism towards other’s work as inappropriate. Third, code fragments are often (if not always) only shards of an ongoing activity which might be already discussed somewhere else.

The issue of privacy raised once again when one of the students complained that all contents of the Gist system (except for group gist) is available to any other user. Initially, every code fragment and lists of them are available to any user of the VLE. However, some students consider it an error, as they perceive the Gist system to be their private space, in which they can store different artefacts of their current work. It was requested to add a possibility to “hide” the contents of code fragments by default and introduce privacy settings into the system.
6 Discussion and future work

6.1 Discussion

The Gist system became an important part of the VLE. Being small and simple, it gave me a possibility to design and develop various specific applications.

According to data and interviews with teachers, the Gist system by itself is not a popular tool among students (as many of students employed it only once). However, it is considered to be a convenient tool by teachers and TAs. Currently, they employ the Gist system to spread instructions (e.g., for project work, midterm exam, homeworks) and other forms of scaffolding among the students.

Thus, it can be concluded that the Gist system solves some problems of educational process by enabling teachers (or, instructors) to operate in students’ ZPD. The low level of students’ engagement in the process is somewhat discouraging, however it still denotes that knowledge exchange practices are present in the DSM and are enabled by the VLE and Gist system. This way, the initial design hypothesis H1 is partially supported, as the Gist system enables teachers and TAs in their attempts to communicate knowledge but knowledge exchange practices between students still require encouragement, probably, in a form of another design intervention.

The UI of the Gist system is simple, clear, and minimalistic. However, according to interviews with students and observations, it is sometimes not obvious, especially in the case of specific applications of the Gist system (e.g., for assignment submission: students not always find the way to do it even though it requires one push of a button). Perhaps, a proper introduction (e.g., in a form of tutorial) to the Gist system (and associated applications) can solve this problem.
The evaluation showed that the Gist system can be used for various purposes, which are not initially considered in the design (e.g., for presentations or as a personal notebook).

It is important to notice the failure of the SM commentaries block intervention (and the failure of the design hypothesis H2). The exact reasons of the failure are not clear and require further investigation. However, literature (see for example [21]) suggests that in the educational context users might be sensitive regarding their private Social Media identities and would like to avoid using them. Another possible reason is that students’ discussion practices are well-established and are hard to change.

The Gist system did not fully replace old practices of code sharing. It is evident that sending code chunks via Social Media is still practiced by students. However, after the introduction of the Gist system, only small chunks of code (one-liners) are sent via SM.

In my opinion, Gist is an example of a small system which can make a huge difference. Being a very simple and basic system it became a platform for complex and sophisticated applications. Even though the Gist system itself did not yet become a popular tool among students, it enabled other applications to emerge and inspired both students and teachers to come up with lots of various and interesting ideas on how would they like to re-shape their virtual educational toolkit.

### 6.2 Future work

I would like to finish the report with some design ideas which emerged during interviews and discussions with teachers and students and were not yet implemented.

Several students suggested that code fragments in the Gist system should be private and unavailable to other users unless the author decides to share them. One of the possible directions of the development of the Gist system is the introduction of privacy settings, e.g., if a form of access control lists (ACL). This intervention might encourage the usage of the Gist system, since students will feel ownership and control over the information they store and share.
6.2. Future work

In this report I almost did not mention “data” as a digital artefacts. DSM teachers and students extensively employ datasets in a form of CSV-files and other formats. Data sharing is an important practice which was not yet supported by the Gist system. Due to peculiar technical properties of data sharing it was not implemented in the course of this project. However, this is definitely an application for future development as it was requested numerously by teachers and TAs.

Currently, the Gist system still relies on third-party services, such as SNS, to enable sharing. However, the experiment on SM commentaries block and previous literature showed that the employment of SM in educational context might fail. I consider a development of an integrated messaging system which would be a part of the VLE and will allow users to share with each other without exposing their private identities, accounts or email addresses.

These are only few of numerous applications which were suggested by students, teachers, and TAs. Yet, they demonstrate that even a small tool such as the Gist system can become a powerful instrument.
Bibliography


