Q2A-I: A Support Platform for Computer Programming Education, Based on Automated Assessment and Peer Learning

Simone Papandrea[1], Andrea Sterbini[1], Marco Temperini[2](✉), and Elvira Popescu[3]

[1] Computer Science Department, Sapienza University, Rome, Italy
sterbini@di.uniroma1.it
[2] Computer, Control, and Management Engineering Department, Sapienza University, Rome, Italy
marte@diag.uniroma1.it
[3] Computers and Information Technology Department, University of Craiova, Craiova, Romania
popescu_elvira@software.ucv.ro

Abstract. Management and assessment of homework assignments in programming courses is a challenging topic both for researchers and practitioners. In the current paper we propose a solution based on the blending of automated evaluation and peer learning. More specifically, we introduce a platform called Q2A-I, which provides two main features: (1) automated management and assessment of homework submissions; and (2) peer interaction support on the programming tasks, by exchanging questions and answers through dedicated micro-forums. By (1) we try to relieve the teacher of the burden of manual evaluation of assignments, while still providing sufficient insight into the students' knowledge level to discover potential need for remedial interventions. By (2) we aim to capitalize on the benefits of peer learning, allowing students to request and provide help to each other, and rely on teacher support only in a limited number of cases. The paper provides a description of the Q2A-I platform, followed by an illustration of its practical use in the context of an introductory Computer Programming course. Results related to students' participation level as well as their subjective satisfaction with the platform are reported and discussed.

Keywords: Computer programming education · Homework
Automated assessment · Peer learning

1 Introduction

Computer Programming is one of the main disciplines the students get in contact with in Computer Science Education and it is present in both introductory and advanced learning activities and courses [1]. The main path for letting the learners acquire programming skills is in allowing for extensive programming training, with the undertaking of several programming tasks, accompanied by the related, timely

G. Hancke et al. (Eds.): ICWL 2018, LNCS 11007, pp. 3–13, 2018.
https://doi.org/10.1007/978-3-319-96565-9_1

provided feedback [2]. The accomplishment of such training activity takes often the form of homework [3].

The evaluation of homework can be a very rewarding activity for the teacher, as it can unveil misconceptions and may allow to plan acting on them, by means of remedial learning tasks. On the other hand, in a large class, this activity can be very demanding for the teacher, due to the high number of assignments to assess in a short time, in order to provide timely feedback. In this context, tools for automated evaluation of programming tasks can provide significant support in computer science education. Real time evaluation of homework programming assignments, with provision of feedback, is performed by means of static or dynamic analysis of programs (and sometimes by the integration of techniques from both methodologies). This approach has proven fruitful in increasing learners' programming proficiency [4–8].

At the same time, educational systems which provide support for social learning and peer interaction tend to increase students' participation and engagement [9]. In particular, platforms based on the exchange of questions and answers and student reputation have proven useful [10, 11]. This approach is suitable to the sociable, team-oriented nature of the current generation of students, who are open to sharing, seeking interaction, preferring peer-to-peer and group learning [12]. In addition, it answers students' expectations for immediate communication, instant responses, quick reactions (sometimes placing more value on speed than on accuracy), and prompt rewarding - as peer feedback is usually delivered much quicker than teacher feedback.

Furthermore, this approach can make less pervasive the teacher's intervention, with positive effects on the learner's self-guidance and self-esteem: sometimes the student might feel that a misconception, of which she is aware, could be solved just by means of interaction with peers (e.g., by exchange of questions and answers in a forum). This way the teacher could limit her intervention to possible moderation of the forum, while having time to focus on cases when her direct intervention is more needed.

In this paper we aim to merge the two approaches (automated homework assessment and peer learning), in the context of computer programming. More specifically, we propose a web-based educational platform called Q2A-I, which provides two main features: (i) automated management and assessment of homework submissions; and (ii) peer interaction support on the programming tasks, by exchanging questions and answers through dedicated micro-forums.

The rest of the paper is structured as follows: the next section includes an overview of related work. Section 3 provides a description of the Q2A-I platform, followed by an illustration of its practical use in the context of an introductory Computer Programming course in Sect. 4. Students' subjective learning experience with the platform is reported and discussed in Sect. 5. The paper ends with some conclusions and future research directions.

2 Related Work

Automated analysis, verification, and grading of programming tasks in education is a long standing research topic [13], especially ranging over applications to training in introductory programming, with the aim to support writing programs according to syntax and semantics of a given programming language, and to foster computational thinking and the development of problem solving skills [2, 14].

In the scope of this paper, a homework is a collection of programming tasks to be assessed. In general, programming tasks are analyzed to determine their correctness with respect to the problem at hand; other qualities of the programs are sometimes considered, such as the suitability of the implemented algorithm and the efficiency of the submitted code. In general, the program analysis is based either on Static Analysis (SA) - which gives feedback just by a syntactic and static-semantic examination, without actually running the program, or on Dynamic Analysis (DA) - which is basically done by executing the program, and measuring its success on significant test-cases.

SA approaches are addressed in [14, 15], where compiler generated errors are attached with explanations, in [16], where programs are evaluated based on the structural similarity with correct ones, and in [17], where detection of plagiarism is also managed.

In DA, the dynamic semantics of the program and its logic are considered, and the tests are designed to catch significant aspects of the expected behaviour of the program. In [6] a web-accessible system for programming task assessment is described, based on the application to programs of pre-determined sets of tests. Safety of the execution environment, plagiarism, and data privacy are also considered among the features of the system. Tools for competitive programming use the DA approach to manage well known programming contests [18, 19]. Kattis [4] proposes "programming exercises" aimed to offer both theoretical and practical learning experiences, in an environment providing automated testing; this is also used to support the programming competitions in ACM-ICPC finals. An evolution of the test-based approach managed through DA appears in systems where the students are called to devise the tests to be used for the program assessment. This approach has been shown fruitful in [7, 20]. A slightly different approach is described in [8], where the student proposes her own tests, and the program is assessed on them and also on a set of reference predefined tests. Finally, combining SA and DA is also possible, although not very frequent; [21] provides a noteworthy example.

Beside the automated assessment functionality, our Q2A-I system also provides a social learning space for the students, in which they can actively engage with each other. Indeed, "today, learning is at least as much about access to other people as it is about access to information" [22] and learning environments should be conducive to peer interactions, which trigger learning mechanisms [23]. In particular, providing opportunity for peer learning and peer tutoring is highly beneficial for the students, whether receiving or offering help [24]. Asking questions and providing answers to peers is a worthwhile learning activity, as understanding is socially constructed by means of conversations and interactions around specific problems [25, 26]. Thus, learning is seen as a product of participation in a community; online social settings and in particular question and answer support platforms foster this active participation and collaboration, engaging students and increasing interactions.

3 Q2A-I Platform Description

Q2A-I[1] is a web application built by adding a plugin into the Question2Answer system [27]. The plugin we implemented is aimed to support the administration of homework, with automated grading, and management of micro-forums based on question/answer interaction. The system is implemented in PHP on a Linux based server (*LAMP* architecture).

A Q2A-I homework, H, is a set of programming tasks $H = \{T_H^i, i \in [1, n_H]\}$. In the system, each homework has a dedicated area, showing its definition, and allowing to submit solutions for each one of the tasks. After submission, the student can see the related (automated) assessment. For a homework, and for the related tasks, the assessment consists of a sequence of tests. Tests for each task are predefined by the teacher, at homework definition time. This testing provides feedback on the homework tasks, based on a variety of aspects (*correctness*, i.e. success in the tests, *intricacy*, computed as the cyclomatic complexity, and *efficiency*, based on the average time of execution for the tasks, provided that all of them can run). The system features an anti-plagiarism tool, used to uncover and discourage cheating.

Beside the management of homework submission and assessment, Q2A-I provides the students with various means for social interaction and peer learning; in particular, the following functionalities are offered:

- Student s can propose a question Q^s related to a given task T_H^i (i-th task in the homework H);
- Student s can submit an answer A^s, for a question Q (proposed by any student, including s);
- Student s can comment on an answer A (proposed by any student, also s);
- Student s can comment on a comment proposed by any student $s' \neq s$;
- Student s can mark (*upvote*) a question $Q^{s'}$ (with $s' \neq s$);
- Student s can mark (*upvote*) an answer $A^{s'}$ (with s' ≠ s);
- Student s, who proposed question Q, can select an answer, A, among all the answers submitted to Q, and declare it to be the best answer for Q.

Associated to the above listed social interactions, is a concept of reputation for a given student s: *rep(s)*. As each interaction produces a value (*interaction points - IPs*), *rep(s)* is computed as the sum of all the *IPs* associated to s's interactions.

More specifically, interaction points are gathered by means of active participation:

- Asking questions, answering to questions, commenting (*participation by content creation*)
- Upvoting questions, upvoting answers, selecting the best answer for a question (*participation to content quality selection*)
- A student receives other *IPs* based on votes obtained by her questions or answers, and based on having her answers selected as the best for a question (*usefulness to others*).

[1] Q2A-I platform (*Q2A-Informatica*), available at: https://Q2A.di.uniroma1.it.

Overall, Q2A-I entails two very important features for computer programming instruction. The first feature is that the automated dynamic assessment of programming tasks provides the student with meaningful and real-time feedback about her solution. This may allow the student to learn from errors and, over time, to be able to produce more correct solutions.

The second feature of Q2A-I is the possibility to support and measure the social interaction of a learner in the online environment. Such interactions happen in dedicated micro-forums, each one related to a specific homework task. The contributions coming from the students in these forums are focused on the definition of questions, answers to such questions, upvoting (declaring the usefulness of a question, or the agreement on an answer), and commenting. The significance of this targeted learning community is twofold. On one hand, the students are helped to stay focused on individual homework topics. On the other hand, the amount of interaction in a given forum, represented by the number of questions, the number of answers, the quality of such answers, and the frequency of upvoting, can help the teacher assess the educational value of a homework, or reveal its inadequacy to its aim.

4 Using Q2A-I in Practice

The Q2A-I system was used in a course on *Basics of Computer Programming (with Python)*, for Bachelor Program in Computer Science, Sapienza University of Rome, fall semester 2017–2018.

Students had 5 homework assignments proposed by the system. The final grade (a value between 18 – lowest, and 30 – highest) would be the average grade of the submitted assignments; so a student had to submit at least 3 fully successful homework assignments to reach at least the minimum grade, otherwise the student would fail. For a homework, H, the evaluation is the average of the (usually three) associated tasks, T_H^i, whereas a task's assessment is as follows:

- 0–30 points according to the average percentage of successful tests for that task
- 0–2 points according to the output of the cyclomatic complexity computation (intricacy of the code: the less, the better)
- 0–2 points for the efficiency of the proposed solutions (measured as the average execution time during the tests, only if all the tests are successful).

As mentioned in the previous section, an anti-plagiarism mechanism is in place in Q2A-I, to discourage cheating behaviors. In particular, during our trial we applied the following rules: given a solution for homework $H = \{T_H^i, i \in [1, n_H]\}$, submitted by student s, a textual anti-plagiarism analysis is performed on each task solution T_H^i:

- if an act of substantial cheating is unveiled, and this is the first time for s that such an event occurs, then a *yellow card* is presented to s;
- if cheating is unveiled for a student s who already has a *yellow card*, then a *red card* is issued for s.

The student who has been "red-carded" on homework *H*, has *H* canceled, and will have to perform a special examination, in presence, invigilated, consisting basically in producing a homework similar to *H*.

Our study was conducted in a class comprising of 432 students; 112 of them did not register with Q2A-I and did not submit any homework, dropping out of the course. Out of the remaining 320 students, only 111 had a level of participation characterized by more than 500 *IPs*, which we considered to be the threshold for a fair engagement. Figure 1 shows a comparison between the level of participation in Q2A-I and the final grade obtained for the homework assignments; a general correspondence between higher participation and better grades can be observed. A total of 183 students failed, corresponding to a very low average degree of participation in Q2A-I.

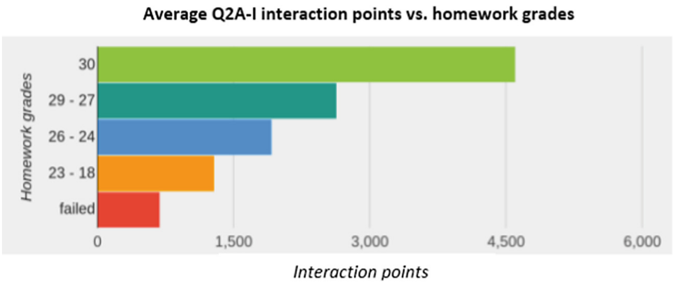

Fig. 1. Relation between students' participation in Q2A-I and homework grades; each bar represents the average *IPs* value obtained by the students in the corresponding grade bracket.

In our study, interaction points were granted according to the following criteria:

- 100 *IPs* were granted for registering in Q2A-I;
- for Questions: posting = 20 *IPs*; having_it_upvoted = 10 *IPs*; voting = 10 *IPs*;
- for Answers: posting = 40 *IPs*; having_it_upvoted = 20 *IPs*; voting = 10 *IPs*;
- selecting an answer as the best for one's own question = 30 *IPs*;
- having one's own answer selected as best for a question = 300 *IPs*;
- posting comments was not eligible for *IPs*.

The Q2A-I system embeds some facilities supporting data visualization. In particular, it produces a resource in GEXF format (Graph Exchange XML Format), which can be used for automated or manual visual representation and help analysis by the teacher. Figure 2 shows a visualization of the social network built by the students in Q2A-I, while interacting and gaining *IPs*. The figure was obtained by feeding the data in Gephi [28, 29] and selecting only the students that accumulated at least 500 *IPs* (i.e., 111 students). HITS (Hyperlink-Induced Topic Search) algorithm was used, which is aimed to evaluate the rank of nodes, according to incoming links (*Authority*) and referenced ones (*Hub*). For the visualization of students' interaction in the Q2A-I social network, the interpretation of the rank elements is as follows:

- Authority: basically this rate shows the usefulness of the student's contributions (as they elicited further participation, and content production, from the others). Intuitively, this is based on the points derived by the votes obtained by the questions and answers posted by the student, and by the amount of answers obtained by the student's questions.
- Hub: this rate shows how the student participated in the network, by adding contents or votes in relation to contributions of others (answers posted by the student, and student's voting activity, on questions/answers of others).

Overall, we can see that there is a relatively small number of students with very active participation in Q2A-I, who drive most learner interactions. While having both posters and lurkers in the learning community is expected, our goal as instructors is to try to increase students' engagement with Q2A-I; a more active and even contribution to the community is highly desirable.

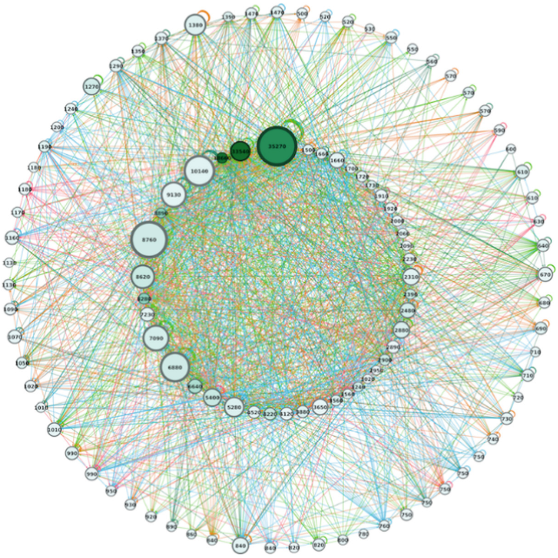

Fig. 2. Visual representation of the Q2A-I social learning network. Each node represents a student; each edge represents one interaction between the students, whereas the edge color depends on the homework related to the interaction. Node dimension indicates how high the authority rate is, while node color indicates the hub rate (the darker the color, the higher the rate). (Color figure online)

5 Students' Experience with Q2A-I

Beside the objective measures of system use, presented in the previous section, students' subjective satisfaction with the platform is an important success indicator. Therefore, at the end of the semester, we asked students to fill in an opinion survey in order to gauge their learning experience with Q2A-I. The questionnaire addressed the following issues: (1) a general evaluation of the course and its lectures; (2) the perceived difficulty of the homework assignments; (3) an evaluation of Q2A-I's usefulness; (4) suggestions for improvements. A total of 132 students filled in the questionnaire and a summary of their answers is reported next.

According to Fig. 3, the answers provided in Q2A-I were considered generally helpful for the comprehension of course topics; in particular, the high appreciation for peers' answers is relevant.

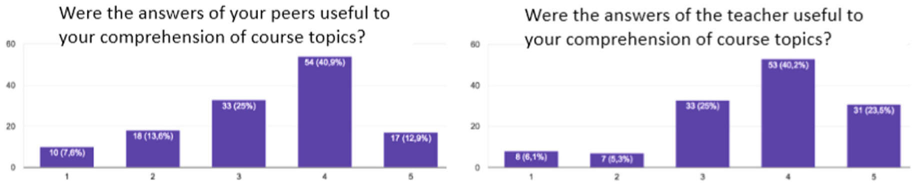

Fig. 3. Students' perceived usefulness of the answers provided in Q2A-I; a 5-point Likert scale was used (1 - Not at all useful; 5 - Very useful). Considering values 3 to 5 as positive, there is a large appreciation by the respondents: 78.8% for the peers' answers and 88.7% for teachers' answers.

As far as the level of participation in Q2A-I is concerned, Fig. 4 suggests a relatively low involvement reported by the students.

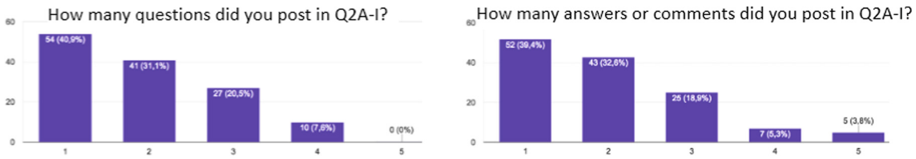

Fig. 4. Students' perceived participation level in Q2A-I, on a 5-point Likert scale (1 - Very few; 5 - Very many); considering values 1 and 2 as a substandard assessment of one's own participation in terms of questions, answers and comments, about 72% of the respondents provide such a negative evaluation.

Regarding the effectiveness of Q2A-I, as perceived by the students, Fig. 5 shows that the overall participation in the system was considered helpful in gaining programming competence.

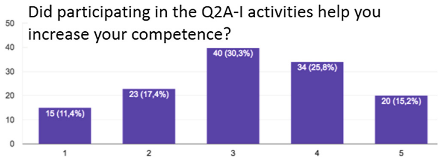

Fig. 5. Students' perceived effectiveness of Q2A-I, on a 5-point Likert scale (1 - Very little; 5 - Very much); 71.2% of the respondents considered their participation helpful at least in a moderate degree.

Finally, the number of homework assignments submitted for evaluation is a relevant measure for student success. Figure 6 shows that 61.4% of the respondents submitted at least 4 assignments and 71.2% at least 3 assignments. This is a relatively good result, given that the minimum number of assignments to be submitted for having a reasonable chance of passing the course is 3.

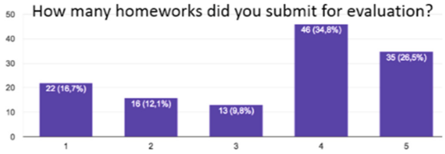

Fig. 6. Number of homework assignments submitted by the students

6 Conclusion

We designed and implemented a platform for the management and assessment of homework assignments in computer programming courses, called Q2A-I. In addition to the automated assessment of programming tasks, the system provides peer learning support, by exchanging questions and answers through dedicated micro-forums. The platform has been successfully used in practice, in the context of an introductory Computer Programming course. Results showed that students with a high level of participation in Q2A-I also obtained the highest grades. Furthermore, students were mostly satisfied with the system: they perceived the answers provided in Q2A-I as generally useful for the comprehension of course topics and the platform was considered helpful in gaining programming competence.

Nevertheless, the number of students with active participation in Q2A-I was not very high, so our goal is to increase students' engagement with the platform; implementing mechanisms for driving a more active and even contribution to the learning community is one of our future work directions. Furthermore, we plan to perform more in-depth analyses regarding students' interactions (including their evolution over time), as well as the impact of Q2A-I system on the learning process.

References

1. Computer Science Curricula 2013 Final Report. The Joint Task Force on Computing Curricula, ACM/IEEE-Computer Society (2013)
2. Pieterse, V.: Automated assessment of programming assignments. In: Proceedings CSERC 2013, pp. 45–56 (2013)
3. Pozenel, M., Furst, L., Mahnic, V.: Introduction of the automated assessment of homework assignments in a university-level programming course. In: MIPRO 2015, pp. 761–766 (2015)
4. Enstrom, E., Kreitz, G., Niemela, F., Soderman, P., Kann, V.: Five years with kattis – using an automated assessment system in teaching. In: Proceedings of FIE 2011 (2011)

5. Brusilovsky, P., Sosnovsky, S.: Individualized exercises for self-assessment of programming knowledge: an evaluation of QuizPACK. J. Educ. Res. Comput. **5**(3), 91–111 (2005)
6. Joy, M., Griffiths, N., Boyatt, R.: The BOSS online submission and assessment system. J. Educ. Res. Comput. **5**(3), 4–31 (2005)
7. Edwards, S.H., Perez-Quinones, M.A.: Web-CAT: automatically grading programming assignments. In: Proceedings of ITiCSE 2008, pp. 328–328 (2008)
8. de Souza, D.M., Maldonado, J.C., Barbosa, E.F.: ProgTest: an environment for the submission and evaluation of programming assignments. In: Proceedings of CSEE&T 2011 (2011)
9. Vassileva, J.: Toward social learning environments. IEEE Trans. Learn. Technol. **1**(4), 199–214 (2008)
10. Sterbini, A., Temperini, M.: Collaborative projects and self evaluation within a social reputation-based exercise-sharing system. In: Proceedings of WI/IAT 2009 (SPeL Workshop), pp. 243–246 (2009)
11. Sterbini, A., Temperini, M.: SocialX: an improved reputation based support to social collaborative learning through exercise sharing and project teamwork. In: Information Systems and Modern Society: Social Change and Global Development, pp. 66–85 (2013)
12. Redecker, C.: Review of learning 2.0 practices: study on the impact of web 2.0 innovations on education and training in Europe. In: JRC-IPTS (2009)
13. Hollingsworth, J.: Automatic graders for programming classes. Commun. ACM **3**(10), 528–529 (1960)
14. Hristova, M., Misra, A., Rutter, M., Mercuri, R.: Identifying and correcting Java programming errors for introductory computer science students. In: Proceedings of SIGCSE 2003, pp. 153–156 (2003)
15. Watson, C., Li, F.W.B., Godwin, J.L.: BlueFix: using crowd-sourced feedback to support programming students in error diagnosis and repair. In: Popescu, E., Li, Q., Klamma, R., Leung, H., Specht, M. (eds.) ICWL 2012. LNCS, vol. 7558, pp. 228–239. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33642-3_25
16. Naudé, K., Greyling, J., Vogts, D.: Marking student programs using graph similarity. Comput. Educ. **54**(2), 545–561 (2010)
17. Rahman, K.A., Nordin, M.J.: A review on the static analysis approach in the automated programming assessment systems. In: Proc. ICSECS'07 (2007)
18. Dagienė, V.: Sustaining informatics education by contests. In: Hromkovič, J., Královič, R., Vahrenhold, J. (eds.) ISSEP 2010. LNCS, vol. 5941, pp. 1–12. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11376-5_1
19. Combéfis, S., Wautelet, J.: Programming trainings and informatics teaching through online contests. Olymp. Inform. **8**, 21–34 (2014)
20. Edwards, S.H.: Improving student performance by evaluating how well students test their own programs. J. Educ. Res. Comput. **3**(3), 1 (2003)
21. Wang, T., Su, X., Ma, P., Wang, Y., Wang, K.: Ability-training-oriented automated assessment in introductory programming course. Comput. Educ. **56**(1), 220–226 (2011)
22. Scholz, R.T.: Introduction: learning through digital media. In: Learning Through Digital Media. Experiments in Technology and Pedagogy. The Institute for Distributed Creativity (2011)
23. Dillenbourg, P.: What do you mean by collaborative learning?. In: Collaborative Learning: Cognitive and Computational Approaches, pp. 1–19. Elsevier, New York City (1999)
24. Topping, K.J.: Trends in peer learning. Educ. Psychol.: Int. J. Exp. Educ. Psychol. **25**(6), 631–645 (2005)
25. Ala-Mutka, K.: Review of learning in ICT-enabled networks and communities. JRC-IPTS (2009). http://ipts.jrc.ec.europa.eu/publications/pub.cfm?id=2721

26. Srba, I., Bieliková. M.: Askalot: community question answering as a means for knowledge sharing in an educational organization. In: Proceedings of CSCW 2015 Companion, pp. 179–182 (2015)
27. Question2Answer homepage. http://question2answer.org/. Accessed 15 Apr 2018
28. The Open Graph Viz Platform. https://gephi.org/. Accessed 15 Apr 2018
29. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: Proceedings of ICWSM 2009, pp. 361–362 (2009)