

Design of CQA Systems for Flexible and Scalable Deployment and Evaluation

Ivan Srba  and Maria Bielikova

Faculty of Informatics and Information Technologies,
Slovak University of Technology in Bratislava, Ilkovičova 2, 842 16 Bratislava, Slovakia
{ivan.srba,maria.bielikova}@stuba.sk

Abstract. Successfulness of Community Question Answering (CQA) systems on the open web (e.g. Yahoo! Answers) motivated for their utilization in new contexts (e.g. education or enterprise) and environments (e.g. inside organizations). In spite of initial research how their specifics influence design of CQA systems, many additional problems have not been addressed so far. Especially a poor flexibility and scalability which hamper: (1) CQA essential features to be employed in various settings (e.g. in different educational organizations); and (2) collaboration support methods to be effectively evaluated (e.g. in offline as well as in live experiments). In this paper, we provide design recommendations how to achieve flexible and scalable deployment and evaluation by means of a case study on educational and organizational CQA system Askalot. Its universal and configurable features allow us to deploy it at two universities as well as in MOOC system edX. In addition, by means of its experimental infrastructure, we can integrate various collaboration support methods which are loosely coupled and can be easily evaluated online as well as offline with datasets from Askalot itself or even from all CQA systems built on the top of the Stack Exchange platform.

Keywords: CQA · System design · Flexibility · Scalability · Askalot

1 Introduction and Related Work

Since the emergence of first Community Question Answering (CQA) systems, they became a substantial source of knowledge online. In the most popular and successful CQA systems, such as Yahoo! Answers or Stack Overflow, communities consisting of millions of users share their knowledge by providing answers on questions asked by the rest of the community. This question answering process is based on knowledge sharing between people and builds on theories of collective intelligence and wisdom of the crowd. More specifically, CQA represents a unique example of online community which utilizes principles of crowdsourcing, human computation and social computing.

In the recent time, motivated by many positive outcomes of open CQA systems on the open web, academy as well as industry became interested in a possibility to adapt CQA systems into additional contexts and environments. At first, potential of CQA systems has been recognized not only in the context of the web, but also in educational domain [1], in crowd-based customer services [2] or in integrated development

environments (IDE) [3]. Secondly, concepts of CQA systems can be utilized not only by large open communities, but also inside organizations (e.g. as a part of company's social platform IBM Connect [4]). The transferability of CQA systems from the web to these new contexts and environments brings several open problems. Especially, their specifics naturally result in many new opportunities as well as limitations which should be taken into consideration when providing users with:

- *essential features* – core functions related to the question answering process (e.g. in educational domain, it is essential to delay teachers' answers to give students enough time to provide answers by themselves [1]), and
- *collaboration support* (e.g. similarly in educational domain, it is necessary to perform precise expertise matching in new question recommendation as students should not be requested to answer questions which they are not capable to address [5]).

While some initial research approaches address these problems, we recognized that implementation of essential features, and integration as well as evaluation of collaboration support methods lacks sufficient *flexibility* and *scalability*. More specifically, we identified two open problems that direct our research presented in this paper:

1. **Low adaptability of essential features to various settings.** A CQA system adapted to a particular context or environment provides a possibility to be deployed in several different instances at the same time (e.g. in several educational or enterprise organizations). In spite of that, design of essential features is not usually flexible enough to handle various different settings.
2. **Ineffective integration and evaluation of collaboration support methods.** CQA systems without appropriate collaboration support would not be so successful. After ten years long research and development, we can take advantage of many collaboration support approaches (as a part of our previous work, we conducted a comprehensive survey in which we analyzed 265 approaches aimed at CQA systems [6]). Achieving loosely coupled integration of the existing collaboration support methods as well as evaluation of novel ones can be, however, quite difficult. Moreover, adapted CQA systems provide a valuable possibility to perform live experiments which can supplement offline evaluation (in our survey, we found out that only 3 out of 169 approaches were evaluated online [6]). Therefore, there is an open question how to make combination of offline and online experiments as effective as possible.

Despite a number of studies providing design frameworks and design guidelines for applications based on collective intelligence [7], human computation [8] as well as for CQA systems themselves [9], there is not any particular study which would tackle flexibility and scalability of CQA systems, especially with focus on adapted CQA systems. Probably, the most similar design guidelines to our aim, are proposed in study [10] which tackles with adaptability of CQA systems to organizational environment.

In this paper, we propose several design recommendations how to tackle the identified open problems by means of design of our educational and organizational CQA system named Askalot. Thanks to its universal design of essential features, we can

deploy it at two universities as well as in MOOC system edX. In addition, its experimental infrastructure allows us to easily implement and experimentally evaluate various research approaches offline as well as online directly in Askalot.

2 Case Study on Educational and Organizational CQA Askalot

In order to achieve our main goal, we draw upon case study on our educational and organizational CQA system named Askalot¹. Askalot represents a novel concept of an organization-wide educational CQA system that fills the gap between open and too restricted class communities of learners [11].

In contrast to the standard CQA systems (e.g. Yahoo! Answers or Stack Overflow), in the design of Askalot we took into consideration especially educational specifics (e.g. presence of a teacher or different levels of students' knowledge) and organizational specifics (e.g. lower number of users or users' familiarity), for more information see [11]. As a part of our previous work, we provided design recommendations, which reflect these specifics in the adaptation of CQA concepts, and we divided them into five categories: dialogue and action, teachers' assistance, workspace awareness, students' self-regulation or guidance, and finally community level management.

Source code of Askalot is provided as an open source². It is implemented in Ruby on Rails with Bootstrap that ensures a responsive design. The quality of our code is assured by employing test driven development (TDD) and regular code review process.

The first version of Askalot was developed for use at our faculty only. Motivated by positive outcomes as well as feedback from the involved students and teachers, we have recently started a cooperation with:

1. *Harvard University* in order to transform Askalot into a plugin to MOOC system edX, which would be suitable for performing A/B experiments (following MOOClet formalism [12]). Our main goal is to replace the standard unstructured forum with an effective tool that can be used by students to share their knowledge and thus solve various course-related questions.
2. *University of Lugano* in order to deploy Askalot at their university as a part of cooperation project in the SCOPES program.

The original design of Askalot was proposed specifically for our university (e.g. it supported only simple non-hierarchical categorization of questions which reflected our subjects' structure). Therefore, it did not provide sufficient flexibility and scalability which is necessary to deploy Askalot in additional various settings. In spite of the same educational domain, edX differs significantly from university environments as well as both universities differs from each other (in terms of their formal educational process, structure, etc.). As the result, we had to rebuild the original system design and following this process, we provide several design recommendations in the following section.

¹ Demo of Askalot is available at: <https://askalot.fiit.stuba.sk/demo>.

² Source code of Askalot is available at: <https://github.com/AskalotCQA/askalot>.

3 Designing Essential Features for Various Settings

Some of CQA essential features are natively flexible and scalable. On other side, other ones required to be redesigned what leads to identification of several design recommendations, which we divided into four groups.

Modular System Architecture. At first, following the requirements, we identified that it is necessary to distinguish two main configurations of our system which we codenamed *Askalot @university* (which is supposed to be deployed at our university and at University of Lugano) and *Askalot @mooc* (which is supposed to be deployed in edX). Consequently, we created three modules. Into the first one, we separated all core features that are common for both configurations (e.g. posting questions and answers or a *global view* containing lists of all questions, categories, tags and users). The remaining two components inherit all features from the core module and add specialized features for a university or MOOC respectively (e.g. in MOOC besides a global view, Askalot provides also a *unit view* – a list of questions asked about a particular learning unit). To achieve the best possible integration with other learning systems (including edX), we adopted LTI (Learning Tools Interoperability) standard. It allows Askalot to obtain data (e.g. information about a student) as well as to provide data back to the learning system (e.g. grades for quality of posts) in the standardized way.

Flexible User Management Integration. Secondly, to face high diversity of educational environments, Askalot can be integrated with several user authentication services. Many universities have their own LDAP servers, and thus Askalot provides a possibility to configure LDAP authentication. Similarly, users can be authenticated by LTI protocol. In both cases, if a user signs into the system for the first time, his/her account is automatically created and filled with data provided by LDAP/LTI. In other words, Askalot does not require any particular import or configuration of users. Another available possibility is to sign up for an account directly in the system. In this case, a user account can be completely anonymous. This option is important especially in situations when students might hesitate to ask questions because their identity is revealed.

Adaptable Self-managed Content Organization. Topic structure plays in CQA an essential role for content organization, navigation and collaboration support (e.g. we can analyze in which topics a student is interested in). At the same time, structure of topics differs significantly across universities or MOOC courses. In addition, as topics reflect actual information needs of students, they are really dynamic and cannot be prepared in advance. Therefore, we proposed to create two-level organization of topics.

1. At first, an asker is requested to select a *category* which reflects the formal structure of a university or a MOOC course (e.g. a subject or a course section, see Fig. 1).
2. Secondly, an asker can add any additional *tags* to describe particular question topics.

This solution provides two main advantages in terms of flexibility and scalability. Deploying Askalot in new settings at a university is quite effective, because it is necessary just to prepare a list of subjects what is quite straightforward. Askalot deployed in

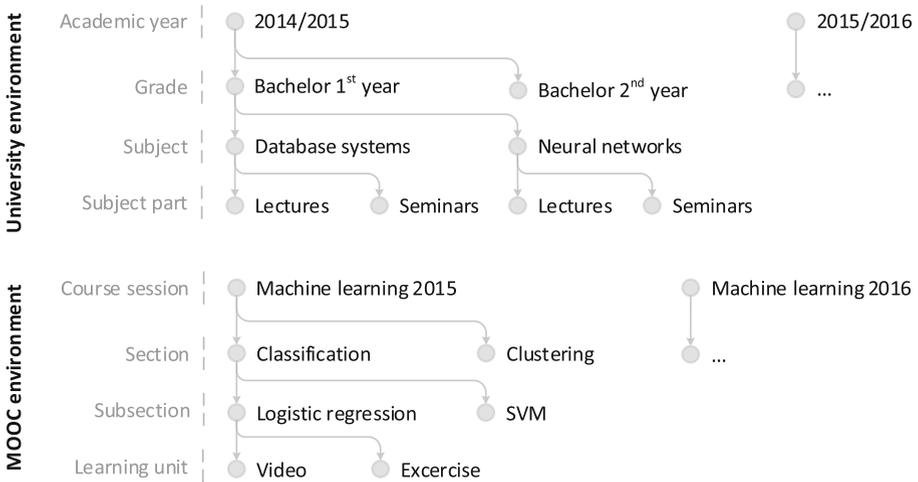


Fig. 1. Example hierarchies of categories in university and MOOC environments.

edX is able to even parse the course structure automatically and create categories on the fly. At the same time, students create a *folksonomy* by means of tags assigned to questions which can be easily adjusted to actual students' needs.

We recognized that categories at universities as well as at MOOC courses need to be organized into a hierarchy. In addition, it is necessary to capture repeating sessions, which are typical for educational domain (i.e. academic years, course sessions), and thus we can easily display user only content from his current user context (e.g. currently opened course session). We solved this requirement by means of a tree structure for each repeating session (an example is provided in Fig. 1). Each node in this tree has four attributes:

1. *domain-specific ID* (e.g. subject code) – to identify the same categories across all academic years or course sessions;
2. *askable flag* – whether students can ask questions in this category (e.g. it is possible to disable asking question for previous academic years);
3. *shareable flag* – whether users with rights to access this category can see also questions from the previous academic years or course sessions (the same categories are identified by means of domain-specific IDs); and finally
4. *roles* – it is possible to give users special roles (i.e. a teacher, an administrator) which assign them special rights, such as assess quality of content or edit/delete posts.

Ubiquitous Activity Awareness and Notifications. As it is necessary to keep students as well as teachers informed about the activity in the system, Askalot provides several ways how to achieve it. Besides notifications displayed directly in the system, users can receive notifications by an email or even let them be sent to their Facebook account.

We have already evaluated the flexibility and scalability of Askalot @university configuration by deploying it as a supplementary tool to the formal educational process

at our university. It involves 1 092 users, who have asked 379 questions and provided 517 answers so far. In addition, Askalot @mooc configuration is also already initially deployed in edX and it will be used in the selected courses during the spring 2016.

4 Designing Universal Experimental Infrastructure

We addressed the second open problem, namely how to effectively and flexible implement and evaluate methods for collaboration support, by designing a universal experimental infrastructure. Its main benefits lie in (1) a modular approach, where all methods are loosely coupled from other methods or system itself, and in (2) a possibility to simply combine training/evaluation of methods on offline datasets (from Askalot or even from other datasets from standard CQA systems) with live experiments.

The proposed experimental infrastructure is fundamentally based on *publish-subscribe pattern*, which ensures its high modularity and loosely coupling. The experimental infrastructure can be divided into three main parts (see Fig. 2):

1. **Data conversion** – At first, it provides utilities to convert any datasets from CQA systems to a dedicated experimental database which has the same database schema as Askalot system. Currently, the convertor for Stack Exchange datasets (distributed under Creative Commons license in XML format) is implemented, however, it is easily possible to create converters for any additional CQA datasets.
2. **Event dispatching** – The second part of experimental infrastructure is responsible for *dispatching events* to subscribed listeners. Each event is represented by four attributes: (1) an initiator who created the event; (2) an action type (i.e. create, update, delete); (3) a resource which is related to the event (questions, answers, comments, views, votes, etc.); and (4) additional custom options. There are two possible sources of events – a *live system* in online experiments (Askalot is implemented to dispatch an event each time a relevant action in the system happens) and *datasets* (either from Askalot itself or from other CQA systems) in offline experiments.

When using datasets, the experimental infrastructure performs an *event simulation job* which selects from the database all resources and each of these resources is converted to a list of events (e.g. a question which has been updated is converted to two events with actions types *create* and *update*). All generated events are consequently sorted by time when they originally happened (i.e. creation, update or deletion time). Finally, the event simulation job sets the current time in the experimental environment to this event time and dispatch the event. This solution allows us to reproduce events exactly in the same way as they would be created by the live system.

3. **Listeners and Profiles** – The third part is dedicated to implementations of research methods themselves by means of *listeners*. Listeners can select from all dispatched events only those they are interesting in and process them in any possible way. In general, there are two main types of listeners: *profilers*, which can model users and content (e.g. user expertise, question difficulty); and *method feeders*, which can trigger various research methods (e.g. a recommendation of new questions to potential answerers) and also directly evaluate their performance. The results of profilers

can be stored in *user/question/answer profiles* which can be easily used by the proposed research methods. These profiles are universal data structures based on four attributes: a value name (e.g. user expertise), a value (e.g. a numeral expression of expertise level), a probability (e.g. how sure we are about the calculated expertise), and a source (there can be even several profilers for expertise calculation).

The experimental infrastructure has been already successfully implemented and utilized in experimental verification of a question routing method based on non-QA data [13] and a reputation method [14], which we evaluated by means of three Stack Exchange datasets (with 10 to 20 thousands of questions in each of them) and consequently we took advantage of experimental infrastructure and we simply deployed its implementation in Askalot, where it is in production environment since May 2015.

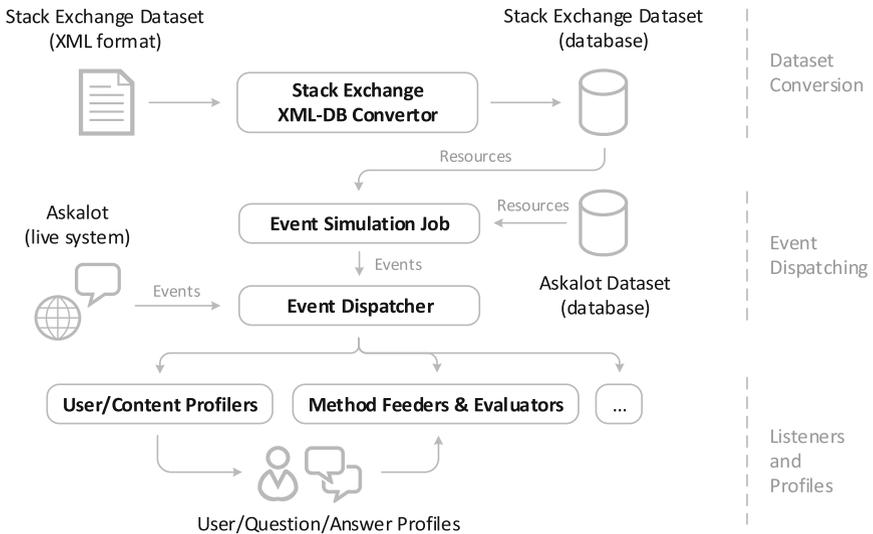


Fig. 2. Overview of experimental infrastructure.

5 Conclusion

Drawing upon the case study on CQA system Askalot, we showed several design recommendations how concepts of CQA systems can be adapted to an educational context and organizational environment with achieving high flexibility and scalability. It allowed us to deploy Askalot in three instances at two universities and in MOOC system edX. Askalot can be characterized also as an open platform based on the universal experimental infrastructure. It can be easily used to implement and evaluate various collaboration support methods (e.g. question recommendation) and even evaluate these methods with data from live system or offline datasets without any code modifications.

Our current primary goal is to deploy Askalot at several edX courses and collect feedback from students in order to make it even more suited for question answering in

MOOCs. In addition, we plan to study specifics of educational question answering process in more details and propose new adaptive support methods for (1) question routing and (2) question retrieval from archives of questions solved in the previous academic years or courses. Another possible direction for our future work is finding a solution how to tackle with performance decrease in experimental infrastructure which naturally appears at the cost of high flexibility.

Acknowledgement. This work was partially supported by grants No. VG1/0646/15 and KEGA 009STU-4/2014 and it is the partial result of collaboration within the SCOPES JRP/IP, No. 160480/2015. The authors wish to thank students participating in AskEd team, who contributed to design and implementation of Askalot and made its deploy in three different settings possible.

References

1. Aritajati, C., Narayanan, N.H.: Facilitating students' collaboration and learning in a question and answer system. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work Companion - CSCW 2013, pp. 101–106. ACM Press (2013)
2. Piccardi, T., Convertino, G., Zancanaro, M., Wang, J., Archambeau, C.: Towards crowd-based customer service: a mixed-initiative tool for managing Q&A sites. In: Proceedings of the 32nd ACM Conference on Human Factors in Computing Systems - CHI 2014, pp. 2725–2734. ACM Press (2014)
3. Ponzanelli, L., Bacchelli, A., Lanza, M.: Seahawk: stack overflow in the IDE. In: Proceedings of 35th International Conference on Software Engineering - ICSE 2013, pp. 1295–1298. IEEE (2013)
4. Matejka, J., Grossman, T., Fitzmaurice, G.: IP-QAT: in-product questions, answers & tips. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology - UIST 2011, pp. 175–184. ACM Press (2011)
5. Yang, D., Adamson, D., Rosé, C.P.: Question recommendation with constraints for massive open online courses. In: Proceedings of the 8th ACM Conference on Recommender Systems - RecSys 2014, pp. 49–56. ACM Press (2014)
6. Srba, I., Bieliková, M.: A comprehensive survey and classification of approaches for community question answering. ACM Trans. Web (2016, submitted)
7. Gregg, D.G.: Designing for collective intelligence. *Commun. ACM* **53**, 134 (2010)
8. Reeves, S., Sherwood, S.: Five design challenges for human computation. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction - NordiCHI 2010m p. 383 (2010)
9. Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., Hartmann, B.: Design lessons from the fastest Q&A site in the west. In: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems - CHI 2011, pp. 2857–2866. ACM Press (2011)
10. Ortbach, K., Gaß, O., Köffer, S., Schacht, S., Walter, N., Maedche, A., Niehaves, B.: Design principles for a social question and answers site: enabling user-to-user support in organizations. In: Tremblay, M.C., VanderMeer, D., Rothenberger, M., Gupta, A., Yoon, V. (eds.) DESRIST 2014. LNCS, vol. 8463, pp. 54–68. Springer, Heidelberg (2014)
11. Srba, I., Bielikova, M.: Askalot: community question answering as a means for knowledge sharing in an educational organization. In: Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work and Social Computing – CSCW 2015, pp. 179–182. ACM Press (2015)

12. Williams, J.J., Heffernan, N.: A methodology for discovering how to adaptively personalize to users using experimental comparisons. In: Proceedings of the Late-Breaking Results at the 23rd Conference on User Modelling, Adaptation and Personalisation - UMAP 2015 (2015)
13. Srba, I., Grznar, M., Bielikova, M.: Utilizing non-QA data to improve questions routing for users with low QA activity in CQA. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pp. 129–136. ACM Press (2015)
14. Huna, A., Srba, I., Bielikova, M.: Exploiting content quality and question difficulty in CQA reputation systems. In: Wierzbicki, A., et al. (eds.) NetSci-X 2016. LNCS, vol. 9564, pp. 68–81. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-28361-6_6](https://doi.org/10.1007/978-3-319-28361-6_6)