The Present and Future of Bots in Software Engineering

Emad Shihab, Concordia University

Stefan Wagner, University of Stuttgart

Marco A. Gerosa, Northern Arizona University and University of São Paulo

Mairieli Wessel, Radboud University

Jordi Cabot, ICREA

Digital Object Identifier 10.1109/MS.2022.3176864
Date of current version: 22 August 2022

SOFTWARE ENGINEERING BOTS are applications that are able to react to external stimuli, such as events triggered by tools and messages posted by users, and run automated tasks in response, working as an interface between users and services. Bots often include conversational capabilities to interact with end users through textual messages (in chatbots) and speech (in voicebots) in the same communication channels as their human counterparts. Bots can support technical and social activities in software engineering, including communication and decision making.

We are witnessing a massive adoption of bots in a variety of domains, including e-commerce, customer service, and education. Software development is no exception.[1,2] Given the essential complexity of software projects and the large community of people around them (stakeholders, designers, developers and, let's not forget, end users), there are plenty of opportunities for bots to jump in and tame this complexity by (semi) automating repetitive tasks. We often see bots working on software repositories, e.g., to manage pull requests; acting as Q&A bots, e.g., for information retrieval; and integrated in software development environments, e.g., automating bug repair.[3]

Automation is even more relevant for open source projects, which typically face sustainability issues. The adoption of bots may help relieve some responsibilities of open source maintainers and allow them to focus on the most critical tasks, benefiting the long-term health of open source. In open source (and inner-source) projects, bots can leverage the public availability of software assets, including source code, discussions, issues, and comments, to target more significant contributions. This special issue offers a perspective on the current role of bots in software engineering.

Overview of Articles in This Special Issue

Zimmermann et al., in their article "The Advantages of Maintaining a Multitask, Project-Specific Bot: An Experience Report," report their experience developing and maintaining a custom bot, Coq bot, which was built to support the Coq team (with circa 40 developers and hundreds of contributors). The bot was initially developed to automate the synchronization between pull requests opened on a GitHub repository and branches on a GitLab mirror. Based on user feedback, the bot evolved to execute other tasks, including merging a pull request, keeping track of pull requests with merge conflicts, and backporting pull requests. The authors note that relying on familiar technology and straightforward and extensible architecture choices can ease the maintenance of a bot by facilitating the onboarding of new bot maintainers.

The adoption and characterization of bots in open source projects is the topic of "From Specialized Mechanics to Project Butlers: The Usage of Bots in Open Source Software Development," by Wang, Wang, and Redmiles. In this article, the authors sample the top 1,000 most popular (using the number of stars as a popularity metric) software development repositories on GitHub and study whether projects employed software bots and, if so, what types of tasks those bots were helping with. As part of their conclusions, they highlight that over 60% of open source projects do use bots, even though these bots often focus on automating simple tasks. The authors note that these bots are typically rule based, reacting to certain events they have subscribed to, and show very limited interactive capabilities.

We are witnessing a massive adoption of bots in a variety of domains, including e-commerce, customer service, and education.

Cogo and Hassan focus, in "Understanding the Customization of Dependency Bots: The Case of Dependabot," on a popular bot used on GitHub: Dependabot. This bot is used for checking and updating dependencies to libraries used in a project. The authors analyze almost 500 projects that use Dependabot and have corresponding configuration files. They conclude that customizing a bot's behavior can help in reducing noise but might also limit the bot's usefulness, as certain features might stop working. Therefore, designers of bots should be careful in considering the tradeoff between allowing bot users to configure a bot and the interaction with features that the bot offers. In general, configurations should be as simple as possible since users interact with many different

bots and therefore are not able to spend significant effort on maintaining their configurations. Cogo and Hassan even suggest that there could be sharing platforms for distributing tailored configuration files for certain project characteristics.

Markusse et al. study the use of benchmarking bots in their article, "Using Benchmarking Bots for Continuous Performance Assessment." The authors show that bots are rarely used to continuously benchmark performance but that the situation is changing, with the newly introduced GitHub Actions. Based on their findings, the authors encourage developers of performance-sensitive projects to consider adopting bot-based benchmarking. Specifically, adopting such bots can help with performance testing, the detection of performance regressions, and providing confidence to maintainers about complex changes.

Golzadeh et al. make a call for better bot identification techniques in their article, "Recognizing Bot Activity in Collaborative Software Development." The authors show that bots are among the most active accounts in open source GitHub projects, yet they are rarely well identified. This widespread presence of bots can impact certain analysis techniques that give credit based on activity. Hence, the authors argue that although current manual techniques tend to be the best option for maintainers, future work should examine the use of machine learning and artificial intelligence to detect bot activity.

Future Challenges

The growth in popularity and contribution of bots is undeniable. The number of libraries, platforms, and reusable bots keeps mounting up. Nevertheless, to fully unleash the potential of bots in software engineering, we would like to draw attention to several technical and socioeconomic open challenges. Regarding technical challenges, we need better systems to facilitate the coordination and collaboration of bots in the same project, as right now, each bot behaves in an independent way, and bots can have conflicting actions. This challenge requires defining bot-specific coordination and integration policies.

> Beyond technical aspects, we need to better understand users' perception of bots and how to optimize human–bot collaboration.

The quality evaluation of bots is another key area. Generally, when bots include conversational capabilities, bot testing implies redefining many of the classical evaluation concepts, as we need to assess the behavioral part of a bot, the conversational component, and the combination of the two.[4,5] Finally, security and privacy also pose relevant challenges. Since we must be able to trust the bots we add to our projects, we need techniques that ensure that bots will not perform malicious activities and leak data and that they request the bare minimum permissions.

Beyond technical aspects, we need to better understand users' perception of bots and how to optimize human–bot collaboration. Bots will need to get better communication and cognitive skills. For instance, when interacting with users, bots should be able to show empathy and react differently depending on the result of their sentiment analysis of the conversation they are having. Learning and mimicking the specific idiosyncrasy of a project (including its vocabulary and natural language use) would increase bots' chances of being accepted. At the same time, bots could help in promoting social diversity in a project. As an example, they could identify and better support contributions from community minorities. Finally, they should be able to explain their behavior to improve their trustworthiness.

The economic impact of bots in a project also deserves special attention. We do not have good economic models to evaluate the return on investment of adopting a certain bot. If we could estimate the value of a bot for a project, it would be much easier to have rational discussions with project owners, considering the cost–benefit analysis of integrating the bot. Even if some bots are released as open source software, there may be costs to adopt them. For example, developers often disregard the cost of learning how to use a bot properly.

So far, we have mostly discussed the impact of bots on software engineering. But since bots are software components themselves, bot development could and should benefit from well-grounded software engineering

practices. What the best practices are for this specific type of software component remains to be seen. For example, it is still unclear how bots, especially collaborative and cognitive bots, will be tested. Bots are becoming smarter, and we know that the creation of smart software applications poses a specific set of additional challenges.[6] We hope the community can benefit from this special issue's articles and keep working on innovating in this increasingly important field.

References

1. M. D. Storey and A. Zagalsky, "Disrupting developer productivity one bot at a time," in *Proc. 2016 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng., (FSE)*, T. Zimmermann, J. Cleland-Huang, and Z. Su, Eds. New York, NY, USA: Association for Computing Machinery, pp. 928–931, doi: 10.1145/2950290.2983989.
2. L. Erlenhov, F. G. de Oliveira Neto, and P. Leitner, "An empirical study of bots in software development: Characteristics and challenges from a practitioner's perspective," in *Proc. 28th ACM Joint Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng., Virtual Event, USA (ESEC/FSE '20)*, P. Devanbu, M. B. Cohen, and T. Zimmermann, Eds. New York, NY, USA: Association for Computing Machinery, Nov. 8–13, 2020, pp. 445–455, doi: 10.1145/3368089.3409680.
3. S. Santhanam, T. Hecking, A. Schreiber, and S. Wagner, "Bots in software engineering: A systematic mapping study," *Peer J Comput. Sci.*, vol. 8, p. e866, Feb. 2022, doi: 10.7717/peerj-cs.866.
4. V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella, "Testing machine learning based systems: A systematic mapping," *Empirical Softw. Eng.*, vol. 25, no. 6, pp. 5193–5254, 2020, doi: 10.1007/s10664-020-09881-0.
5. J. Cabot, L. Burgueño, R. Clarisó, G. Daniel, J. Perianez-Pascual, and R. Rodríguez-Echeverría, "Testing challenges for NLP-intensive bots," in *Proc. 3rd IEEE/ACM Int. Workshop Bots Softw. Eng. (BotSE)*, Madrid, Spain, Jun. 4, 2021, pp. 31–34, doi: 10.1109/BotSE52550.2021.00014.
6. I. Ozkaya, "What is really different in engineering AI-enabled systems?" *IEEE Softw.*, vol. 37, no. 4, pp. 3–6, 2020, doi: 10.1109/MS.2020.2993662.

ABOUT THE AUTHORS

EMAD SHIHAB is a full professor and research chair at Concordia University, Montréal, H3G 1M8, Canada, where he leads the Data-Driven Analysis of Software lab. Contact him at emad.shihab@concordia.ca or https://das.encs.concordia.ca.

STEFAN WAGNER is a full professor of empirical software engineering and the director of the Institute of Software Engineering, University of Stuttgart, Stuttgart, 70569, Germany. Contact him at stefan.wagner@iste.uni-stuttgart.de.

MARCO A. GEROSA is a full professor at Northern Arizona University, Flagstaff, Arizona, 86011, USA, and a Ph.D. advisor at the University of São Paulo, São Paulo, 05508-220, Brazil. Contact him at marco.gerosa@nau.edu or http://www.marcoagerosa.com.

MAIRIELI WESSEL is an assistant professor at Radboud University, Nijmegen, 6525EC, The Netherlands. Contact her at mairieli.wessel@ru.nl.

JORDI CABOT is an ICREA research professor at the Open University of Catalonia, Barcelona, E08035, Spain, where he leads the Software and Systems Modeling Lab. Contact him at jordi.cabot@icrea.cat or https://jordicabot.com.