The work of adopting third-party bot applications for online community governance

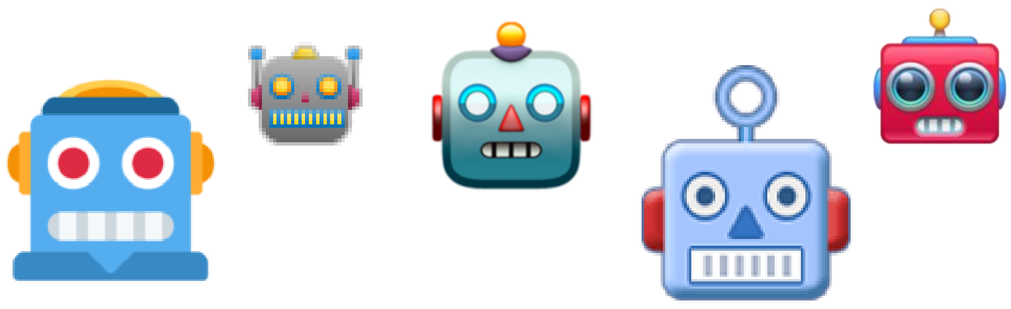
Community Data Science Collective

Sohyeon Hwang[1], Charles Kiene[2], Serene Ong[1], and Aaron Shaw[1]

[1]Northwestern University; [2]University of Washington

Questions can be sent to the corresponding author: sohyeonhwang@u.northwestern.edu

Introduction



Online communities often rely on bots to automate governance of their spaces [1], creating custom bots via platform APIs in examples of **end-user innovation** [2].

But not every community is able to develop a custom bot and instead adopt *existing* bots created by **third-party** developers, leading to potentially **misaligned or imperfect technical solutions** [3].

In lieu of custom bots:

- **How do community leaders decide to use the third-party bots that they use?**
- **What are the consequences of this decision-making process?**

Research Design

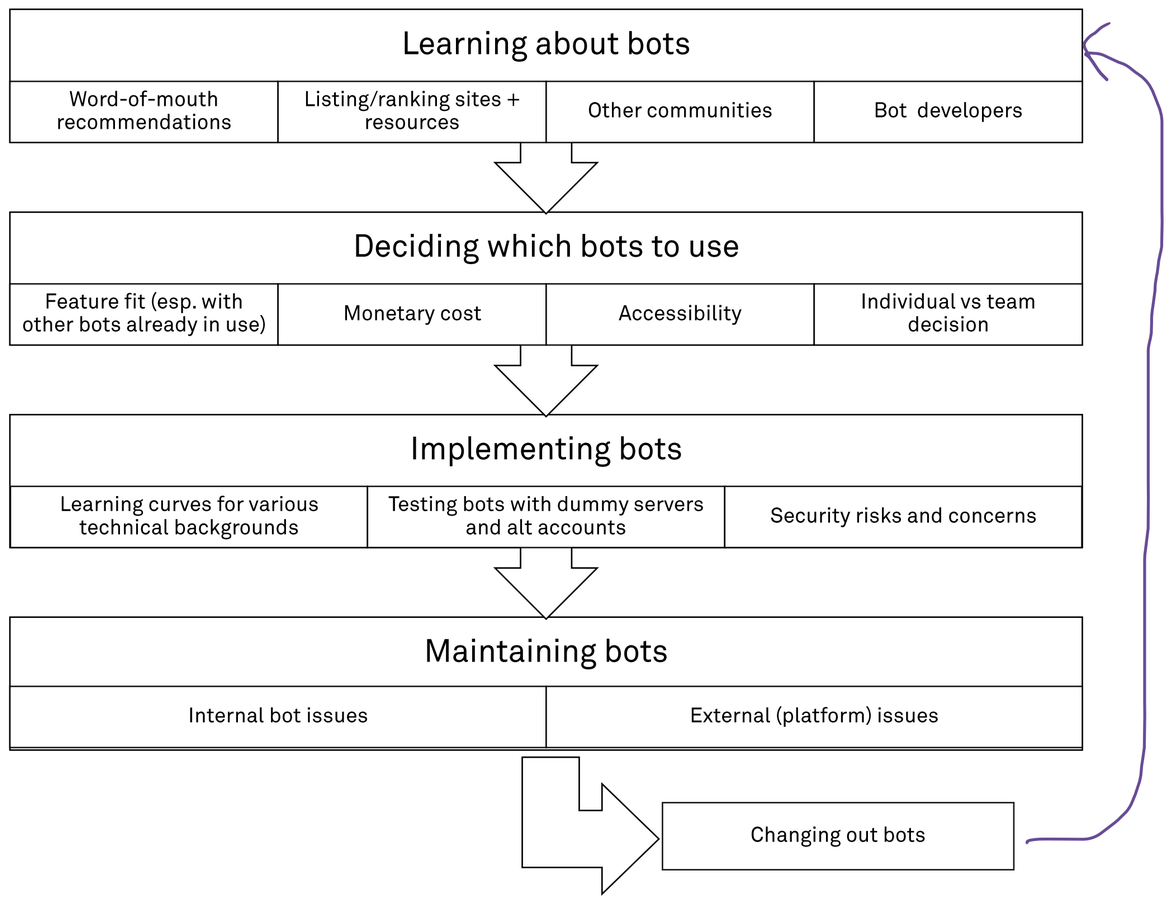
Empirical Setting. Discord, a popular VoIP and synchronous chat-based community platform made up of communities called "servers".

Data. 16 semi-structured interviews w/ former or current Discord server community leaders with experience adding bots.

Method. We conducted inductive line-by-line coding and thematic analysis following Braun & Clarke [4].

Findings

Four key steps in adopting a bot:



We highlight 3 interconnected themes through these steps:

1. Framing **bots as critical governance *infrastructures*** emphasizes the layers of social and technical dependencies upon which they rely.
2. Community leaders using bots showed a **second-order form of innovation in their "bot stacking" strategy**, aka managing patchwork assemblages of multiple bots for communities.
3. We observed **"modularity tensions" in customizing with third-party software**:
   a. Bots were modular but often still highly labor-intensive to adopt;
   b. Functions made available through bots had isomorphic tendencies, limiting feature options.

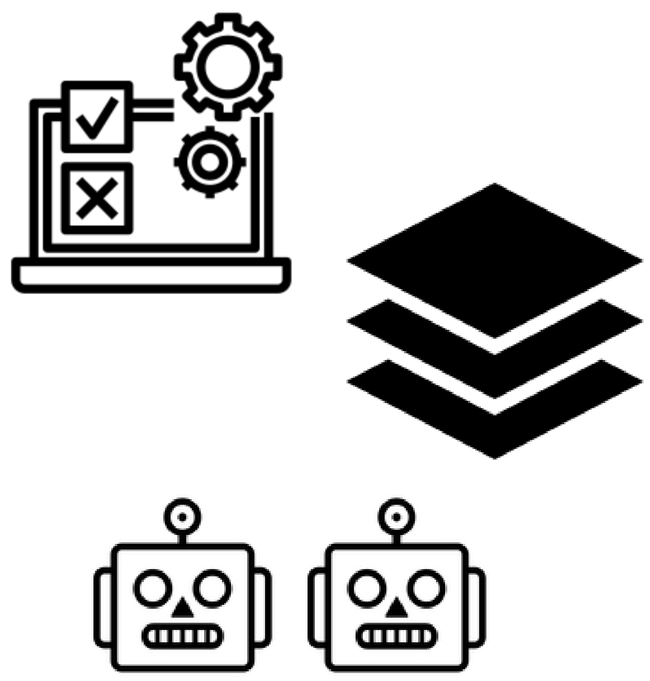
Design Implications and Future Directions

For community leaders.
- Carefully evaluate bots for signals of reliability and good security practices.
- Making simple but flexible testing procedures can reduce start-up costs.

For bot designers and developers.
- Design and release compatible "stacks" of bots, similar to mod packs.
- Make guides accessible to diverse technical levels.

For community platforms.
- Deploy common bot testing environments or standards for end-user and developer use.
- Develop resources to identify novel and unique bots, or likewise, similar bots.

For future research.
- Characterize the ecosystem market that bots "compete" within to be selected.
- Identify mechanisms constraining available bot features, to generate new possibilities for governing with bots.



Acknowledgements + References

Support for this project came from the National Science Foundation (IIS-1910202, DGE-1842165). Special thanks to our participants; Joseph Seering; and members of the Community Data Science Collective.

[1] R. Stuart Geiger. 2014. Bots, Bespoke, Code and the Materiality of Software Platforms. Information, Communication & Society 17, 3 (March 2014), 342–356. https://doi.org/10.1080/1369118X.2013.873069

[2] Eric von Hippel and Ralph Katz. 2002. Shifting Innovation to Users via Toolkits. Management Science 48, 7 (2002), 821–833.

[3] Shagun Jhaver, Iris Birman, Eric Gilbert, and Amy Bruckman. 2019. Human-Machine Collaboration for Content Regulation: The Case of Reddit Automoderator. ACM Trans. Comput.-Hum. Interact. 26, 5 (July 2019), 31:1–31:35. https://doi.org/10.1145/3338243

[4] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. Qualitative Research in Psychology 3, 2 (Jan. 2006), 77–101.