Tutorbot: A Chatbot for Higher Education Practice

Jonas Sjöström(✉) and Maritha Dahlin

Uppsala University, Campus Gotland, 62167 Visby, Sweden
jonas.sjostrom@im.uu.se

Abstract. In this paper, we present the design of Tutorbot – a chatbot software to support learning and teaching in higher education. We account for the implementation of the design as a proof-of-concept and share reflections from experiences in the design and implementation process expressed as design considerations for the design of chatbots in a higher education setting.

Keywords: Chatbot · Design · Architecture · Higher education · Practice · Design science research

1 Introduction

While chatbot technology has matured over time, there is still need for research on how to appropriately add value to human practice through the use of chatbot technology, including challenges to design effective dialogue between humans and bot technologies [3, 11, 14, 17].

In this work, we focus on chatbots in higher education (HE) practice. Potentially, the use of chatbots may influence educational flow to be more interactive and dynamic [1, 6]. A potential advantage of using a chatbot in an educational setting is the facilitation of instant retrieval of information for the learners [4]. Chatbots have also been proposed as a means to estimate learning styles [10, 20], and to stimulate student feedback in e-learning environments [12]. Chatbots may be part of the motivation for continued communication for educational purposes [3]. However, there is a need to factor in expectations from teachers and other stakeholders when designing bot technology in an education setting [18].

While there is a lot of research on chatbots in education, there is little or no research adopting a broader educational practice perspective [16]. A practice approach entails that we take into account multiple stakeholders in the learning situation, and investigate the emergence of social practices and stakeholder interactions given the introduction of chatbot technology. In this paper, we present Tutorbot – a software based on a theoretically and empirically grounded conceptual design for chatbots in HE practice. We provide a set of design reflections for chatbots in HE practice based on our experiences from the design process.

S. Hofmann et al. (Eds.): DESRIST 2020, LNCS 12388, pp. 93–98, 2020.
https://doi.org/10.1007/978-3-030-64823-7_10

2 The Tutorbot Software and Its Rationale

We searched in the web of science core collection for publications where either the topic or the title contained the word “chatbot” or the word “conversational agent”, rendering 374 hits. A refined search within the search result where the title or topic contained the word “education” or “learning” rendered 99 results. We read through each abstract to further narrow down our search. After the reading of abstracts, we ended up with 50 articles that were considered relevant. In addition to those articles, we identified another 13 articles while reading or from suggestions by peers. In total, 63 articles from the years 2001–2017 were identified – informing the rigor cycle both with foundations and contextually relevant methodologies. Also, we factored in literature on general design topics including software architecture [2, 15] and communication as shown below. Also, repeated group discussions with teachers informed the design process.

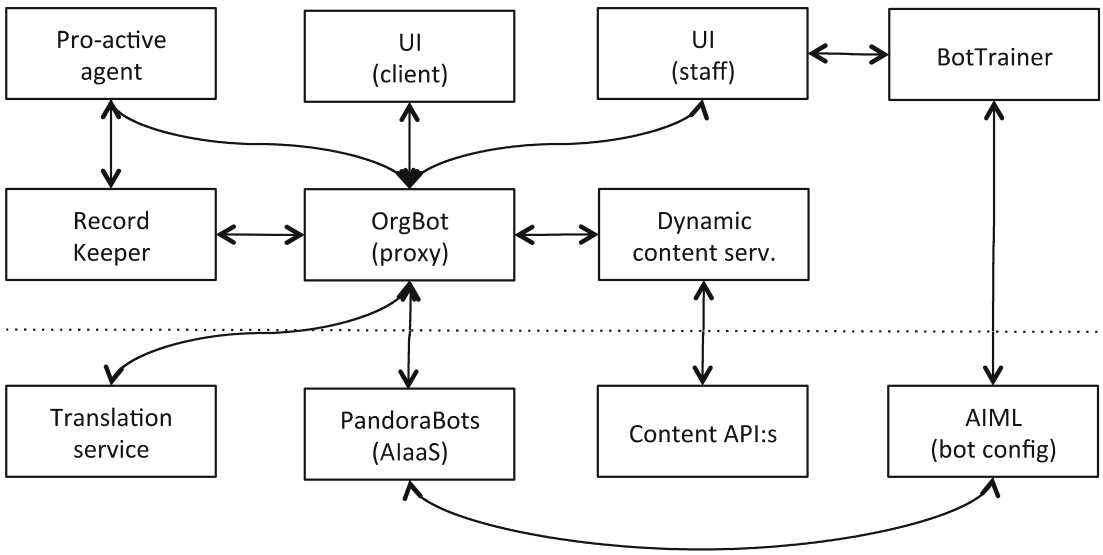

Fig. 1. The Tutorbot architecture.

Figure 1 shows a conceptual architecture for chatbots in higher education, drawing from the literature review as well as from interviews with a teacher group. The core of the architecture is the *OrgBot*, acting as a proxy with built-in logic for every interaction in the system. The OrgBot receives a question from learner through the *client UI* and passes it on to the AI as a service (AIaaS). The AIaaS reads the AIML configuration to find a suitable response for the question. If there is no response, a ‘fallback’ feature is activated, asking the learner if the question should be sent to the teacher. If yes, the OrgBot forwards the question to the *staff UI*. All the steps in the process are recorded by the *RecordKeeper* component, which logs data for future analysis (e.g., for research or education improvement purposes).

The OrgBot and its interaction with the AIaaS correspond to the basic idea of chatbots: Responding to questions. The teacher interviews showed that the chatbot should be able to answer common questions from students, e.g., regarding course content, prerequisites and requirements for a course, and information regarding exams deadlines. From a student point of view, the chatbot may simplify access to important information.

From a teacher point of view, the chatbot's ability to answer such questions could reduce administrative overhead. That is; we view the bot as support and supplement, rather than a substitute for the learner-teacher or peer. The fallback function makes the bot act as a doorway to the teacher when needed. While many studies of bots focus on the interaction between a learner and a bot, we are interested in the bot as an integral part of learning practice. A lack of such a perspective creates a risk that bots decrease interaction between students and teachers, which may cause negative feelings and consequently affect learning negatively.

Sometimes, the AIaaS returns control codes to the OrgBot. If a return message includes such a control code, it is passed along to the dynamic content service, which fetches data using external content API:s, and injects them into the response message. The mechanism allows for control codes in the AIML definitions that translate into dynamic content at run-time. The design supports fetching data about the syllabus, schedule, assignments, et cetera. By providing such content dynamically, the AIML definitions remain useful over time.

The *BotTrainer* subsystem allows teachers to provide answers to questions from the 'fallback' scenario above, while at the same time allowing for supervised training of the chatbot. The chatbot needs to evolve under the supervision of humans, to align its behavior with institutional norms, and to promote the quality of responses. Even though it is technologically feasible to automatically train bots [e.g., 7], we believe that it is risky to do so in higher education. In the Swedish context, bot actions in a University context are a form of agency, likely comparable to exercising public authority.

The *pro-active agent* subsystem contains the logic to initiate conversations, e.g., quizzes, course evaluations, or reminders to log on to the course Intranet. The main idea is to facilitate a mechanism to promote student activity, in keeping with the idea of supportive accountability in eHealth [13]. Conceptually, the subsystem exists in the conceptual architecture to allow for chatbot features beyond question-answers, i.e., design for mutability as suggested by Gregor and Jones [5]. Continued research may, for instance, include designs where data analytics methods are employed to identify when and how to trigger conversations with learners based on quiz results, inactivity, et cetera.

Finally, the idea of a *translation service* is still in its infancy, but it may prove very powerful to integrate a cloud translation service to facilitate interaction with international students, and provide them with an automatic translation of essential course information into their native language.

3 Proof-of-Concept Evaluation

The architecture discussed above was implemented into software both (1) as a proof-of-concept and (2) in preparation for continued evaluation in a Java course. The architecture consists of a set of interacting subsystems, which resonates well with the design of a microservices architecture [2, 15], i.e., small, loosely coupled components that interact over the Internet via the REST lightweight protocol. A microservices architecture and its loose coupling allows different subsystems to be implemented using different programming languages, and in different server environments.

We used PandoraBots as an AIaaS, due to two reasons. It allows us to define bot behavior using AIML, a 'de facto' standard that is reasonably convenient and works well with supervised learning. Also, it was available as a cloud service via a REST API, making it seamless to integrate into the architecture.

We implemented the software to operate in communication channels that the students already use, to make it easily accessible and to avoid the risk of non-use. We justify this idea by drawing on design thinking [9] – thinking of Tutorbot as part of an ecology of artifacts as experienced by the learners. The principle is also supported by the teacher feedback, which clearly shows the importance of bot accessibility, both from a student and from a teacher point of view. Following a survey among the students, we decided to implement the Tutorbot UI in the Facebook Messenger environment.

The other services in the architecture were all quite trivial to implement. The dynamic content service and the pro-active agent are only stubs at his point, but they are integrated into the architecture and ready to develop further when needed.

One lesson learned from the implementation work is the need to address privacy issues at an early point. Not only is it necessary *per se* when we use educational technology for education and research, but it was also needed in this case due to requirements from the Facebook Messenger API. Facebook requires us to upload a privacy policy to open up the chatbot for other users than invited testers. Also, requirements from the cloud services in use demand encrypted communication channels. The privacy issue may prove problematic in a scenario where this technology is used in a larger scale, still running in a cloud setting. While legislation differs between different parts of the world, we suspect that student questions may sometimes be rather sensitive in nature, thus not always suitable for cloud storage. Privacy issues needs to be thoroughly factored in to the design work, from the very inception of the process.

4 Concluding Discussion

In this paper, we have presented a conceptual architecture for chatbots in higher education. The architecture has been implemented into a chatbot software accessed via Facebook Messenger. Through the design process, we have identified a set of tentative design considerations: (i) A recommendation to deliver HE chatbot functionality within the existing *student ecosystem* of applications to promote use – useful to promote the use of the chatbot. (ii) A call to build HE chatbot technology that *promotes human interaction* in the learning context – rather than considering it a substitute to human interaction. (iii) An argument for *supervised learning* – due to the demand for quality controlled responses from the chatbot, and a potential role of the chatbot as an agent exercising public authority. (iv) A reflection about the *multitude of privacy norms* that govern design – in this case both educational norms, research ethics, and regulations to comply with third party cloud services regulations.

Future work includes the process of defining chatbot behavior for a particular course. The behavior will be defined through the creation of AIML documents drawing from both literature and an analysis of student questions from previous instances of the course. Clearly, the behavior of the chatbot is an essential part of design. Previous research, insofar as feasible, will be factored in to the AIML design process, such as the concept of academically productive talk [e.g., 19]. During the course, we promote students to the Tutorbot as a first resort when asking questions. Chatbot log data will be used to produce descriptive statistics of the use of the chatbot. In addition, we will conduct interviews with students and teachers to obtain qualitative (and possibly quantitative) data about their experience of using the chatbot.

We see an interesting future field of research in combining the practice-oriented approach suggested here with collaborative learning environments based on multiple interacting agents [8]. In such a setting, various agents with different roles would intervene in discussions among learners and teachers. Great challenges lies ahead for social science research to understand how to design and employ conversational agents to facilitate an education practice effectively supporting students' learning.

References

1. AbuShawar, B., Atwell, E.: ALICE chatbot: trials and outputs. Comput. Sist. **19**(4), 625–632 (2015)
2. Dragoni, N., et al.: Microservices: yesterday, today, and tomorrow. In: Mazzara, M., Meyer, B. (eds.) Present and Ulterior Software Engineering, pp. 195–216. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67425-4_12
3. Fryer, L.K., et al.: Stimulating and sustaining interest in a language course: an experimental comparison of chatbot and human task partners. Comput. Hum. Behav. **75**, 461–468 (2017)
4. Ghose, S., Barua, J.J.: Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor. In: 2013 International Conference on Informatics, Electronics and Vision (ICIEV), pp. 1–5 (2013)
5. Gregor, S., Jones, D.: The anatomy of a design theory. J. Assoc. Inf. Syst. **8**(5), 312–335 (2007)
6. Griol, D., Molina, J.M., de Miguel, A.S.: The *Geranium* system: multimodal conversational agents for e-learning. In: Omatu, S., Bersini, H., Corchado, J.M., Rodríguez, S., Pawlewski, P., Bucciarelli, E. (eds.) Distributed Computing and Artificial Intelligence, 11th International Conference. AISC, vol. 290, pp. 219–226. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07593-8_26
7. Hassani, K., et al.: Architectural design and implementation of intelligent embodied conversational agents using fuzzy knowledge base. J. Intell. Fuzzy Syst. **25**(3), 811–823 (2013)
8. Hayashi, Y.: Togetherness: multiple pedagogical conversational agents as companions in collaborative learning. In: Trausan-Matu, S., Boyer, K.E., Crosby, M., Panourgia, K. (eds.) ITS 2014. LNCS, vol. 8474, pp. 114–123. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07221-0_14
9. Krippendorff, K.: The Semantic Turn - A new Foundation for Design. CRC Press, Boca Raton (2006)
10. Latham, A.M., et al.: Oscar: an intelligent conversational agent tutor to estimate learning styles. In: 2010 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010). IEEE, New York (2010)
11. Leonhardt, M.D., et al.: Using chatbots for network management training through problem-based oriented education. In: Spector, J.M., et al. (eds.) 7th IEEE International Conference on Advanced Learning Technologies, p. 845. IEEE Computer Society, Los Alamitos (2007)
12. Lundqvist, K.O., Pursey, G., Williams, S.: Design and implementation of conversational agents for harvesting feedback in elearning systems. In: Hernández-Leo, D., Ley, T., Klamma, R., Harrer, A. (eds.) EC-TEL 2013. LNCS, vol. 8095, pp. 617–618. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40814-4_79

13. Mohr, D.C., et al.: Supportive accountability: a model for providing human support to enhance adherence to eHealth interventions. J. Med. Internet Res. **13**, 1 (2011)
14. Neves, A.M.M., et al.: IAIML: a mechanism to treat intentionality in AIML chatterbots. In: Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, pp. 225–231 (2006)
15. Newman, S.: Building Microservices: Designing Fine-Grained Systems. O'Reilly Media Inc, Newton (2015)
16. Orlikowski, W.J.: Sociomaterial practices: exploring technology at work. Organ. Stud. **28**(9), 1435–1448 (2007)
17. Picard, R.W., et al.: Affective learning—a manifesto. BT Technol. J. **22**(4), 253–269 (2004)
18. Tamayo-Moreno, S., Perez-Marin, D.: Adapting the design and the use methodology of a pedagogical conversational agent of secondary education to childhood education. In: Garci-aPenalvo, F.J., Mendes, A. (eds.) 2016 International Symposium on Computers in Education (SIIE). IEEE, New York (2016)
19. Tegos, S., Demetriadis, S., Papadopoulos, P.M., Weinberger, A.: Conversational agents for academically productive talk: a comparison of directed and undirected agent interventions. Int. J. Comput.-Supported Collab. Learn. **11**(4), 417–440 (2016). https://doi.org/10.1007/s11412-016-9246-2
20. Yun, E.K., Cho, S.B.: Learning classifier system for generating various types of dialogues in conversational agent. In: Liu, J., Cheung, Y.M., Yin, H. (eds.) IDEAL 2003. LNCS, vol. 2690, pp. 84–88. Springer, Berlin (2003). https://doi.org/10.1007/978-3-540-45080-1_11