# Bringing Automation to the Classroom: A ChatOps-Based Approach

Eueung Mulyana
STEI
Institut Teknologi Bandung
Bandung, Indonesia
eueung@stei.itb.ac.id

Rifqy Hakimi STEI Institut Teknologi Bandung Bandung, Indonesia rifqyhakimi@stei.itb.ac.id Hendrawan STEI Institut Teknologi Bandung Bandung, Indonesia hend@stei.itb.ac.id

Abstract—In this paper we present the design and implementation of a chatbot-based virtual assistant called LTKA-Bot. Its main function is to streamline and to automate manual and administrative tasks while supporting other course-related activities. It differs from other recent approaches in that it is based on the ChatOps paradigm instead of on some AIbased schemes. LTKA-Bot introduces a case of automation and demonstrates its potentials in the area of higher education which is steadily transformed to cope with technological progresses and administrative policy dynamics including accreditation. International accreditation body such as ABET requires fulfillment of certain criteria which in turn also requires appropriate course design and quite a lot of document works. LTKA-Bot borrows the idea of automation which is very common in the modern tech companies to cope with such challenges. It facilitates more efficient course-related activities while satisfying all document requirement with minimal effort.

Index Terms—chatops, chatbot, automation, ABET, virtual assistant, bot in education

#### I. INTRODUCTION

Courses belong to the most important activities in education since they are building blocks of any educational programs. Courses shall be designed according to goals for given subjects for intended group of participants. In the past, lecturers had a high degree of freedom in managing their courses, which were also relatively stable over time. Recent changes in the landscape of educational demands and regulations have been forcing many institutions to shape their programs (thus the corresponding courses) according to some standards and accreditations. In most cases, this implies additional works required for compliance documents. In some cases this requires a complete redesign of the courses. While the changing landscape and its consequences are inevitable, reducing administrative burden introduced by this phenomenon is highly relevant, especially for institutions with limited resources.

In the fields of engineering and computing, ABET (Accreditation Board for Engineering and Technology) is the most adopted accreditation. ABET prescribes a number of student outcomes that students should attain by the time of graduation. Since the work of equipping students with the attributes specified in program outcomes must be done at the individual course level, all faculty members involved in teaching required courses must understand and be involved in

the accreditation process on a continuing basis, not just in the months preceding each visit [1].

Fig. 1 shows elements of course design as discussed in [1]. Creating a course to achieve specified outcomes requires effort in three domains: planning, instruction and assessment. Planning efforts include identifying course content and defining measurable learning objectives for it. Instruction efforts cover selecting and implementing the methods that will be used to deliver the specified content and facilitate student achievement of the objectives. Assessment elements deal with selecting and implementing the methods that will be used to determine whether and how well the objectives have been achieved and interpreting the results. As indicated in Fig. 1, the three stages are not purely sequential i.e. the information collected in each of them feeds back to each of the others in a cycle that leads to continuous improvement. If the assessment reveals that an objective has not been satisfactorily achieved, the nature of the failure may suggest reframing the objective or modifying the instruction used to address it. Similarly, as the quality of the instructional program improves, new objectives may be formulated to encompass higher levels of achievement and the course instruction and assessment modified accordingly [1]. All these processes establish a continuous improvement scheme but also simultaneously introduce more overheads for course management.

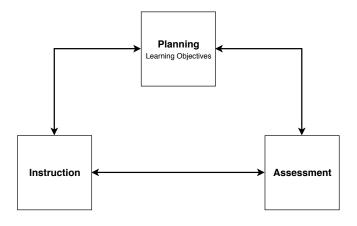


Fig. 1. Elements of course design.

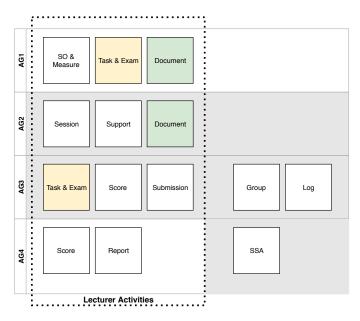


Fig. 2. Modules in LTKA-Bot.

While accreditation requirement (which is not only limited to ABET) was a major motivation, our work has a broader perspective i.e. to streamline process by introducing automation for every possible aspect of the courses. This way, we can spend less time managing organizational challenges and more time on the tasks that really matter (and require more thoughts), like careful lesson planning and assessment writing [9]. Also worth noting, that teaching presents a lot of organizational and communication challenges, which can be overwhelming without the appropriate systems and routines to address them.

Based on above considerations, we design LTKA-Bot. To some extend, it is a virtual teaching assistant, but it differs to currently popular approaches such as Jill Watson [2]. LTKA-Bot is not meant to replace real teaching assistants by doing free-form conversation with or answering subject matter questions from the students, but it does provide services related to course activities. These services cover many aspects, ranging from group or task management for students to automatic document generation for reporting and accreditation purposes. Details will be discussed in Section III.

LTKA-Bot is based on so-called ChatOps paradigm. ChatOps (Chat and Operations) can be seen as a model where people, tools, process and automation are connected in a transparent flow. It is about conversation-driven interaction by placing tools directly in the middle of the conversation, so that everyone is pairing all of the time [7], [8]. This approach shall perfectly complement recent AI-based approaches for virtual assistants that might boost application of automation in education.

#### II. RELATED WORK

The landscape of research in the field of virtual teaching assistant (TA) is mostly dominated by the emergence of the

use of artificial intelligence (AI). AI is currently progressing at an accelerated pace, and this already impacts on the profound nature of services within higher education. An overview of the impact of AI on teaching and learning in higher education is discussed in [4]. A notably work in this area is Jill Watson (JW) bot presented in [2]. JW was tested in a real class to autonomously respond to student introductions, post weekly announcements and answer routine, frequently asked questions. JW was so valued by students that one wanted to nominate her to the outstanding TA award.

Other interesting applications of chatbots in a broader context (i.e. including non-educational) can be found in [3], [5], [6]. The authors in [3] used a hybrid crowd-AI architecture and introduced a crowd-powered conversational assistant built to automate itself over time. [5] presented a chatbot-mediated task management to support team collaboration. Lastly, [6] proposed a customer service chatbot for e-commerce systems.

From our extensive literature surveys, there is currently no work similar to our proposal, especially in the following aspects. First, we are designing a virtual bot assistant with a ChatOps paradigm to support and streamline course-related activities. Second, we are creating and automating a course management system to comply with (ABET) accreditation requirement. It is worth noting that our work is no replacement of those mentioned earlier. In fact they complement each other e.g. LTKA-Bot and JW-like bots can be applied together in a course to bring more benefits and take automation in the field of education to the next level.

### III. LTKA-BOT

LTKA-Bot consists of functionalities which are depicted as blocks in Fig. 2. It generally covers four activity groups (AG1 - AG4). The first three activity groups (AG1 - AG3) are related to elements of course design mentioned in Section I:

- AG1 is planning stage, includes Student Outcome (SO),
   Task and Document modules.
- AG2 is instruction-related activity, includes Session, Support and Document modules.
- AG3 is assessment-related activity, includes Task, Score, Submission, Group and Log modules.

The last activity group AG4 provides supports for reporting purposes and other course-end administrative tasks. Note that a module may provide services for different stages / activity groups. From users' perspective, services provided by LTKA-Bot is available to three kind of roles i.e. student, lecturer, admin. Admin role may have access to all services (but it is still context-dependent). In Fig. 2, modules accessible from lecturer role are given in the block marked with dashed-line while modules accessible from student role are marked by grey area.

The system architecture of LTKA-Bot is illustrated in Fig. 3. The system consists of several flexible services glued together with the bot instance as the central component. This non-monolith approach offers flexibility to scale across multiple base infrastructure or to adapt to more complex features/requirement in the future. The bot instance is based on Hubot

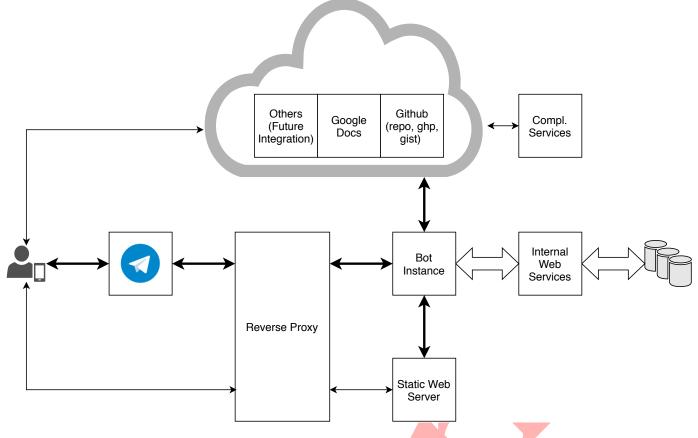


Fig. 3. LTKA-Bot system architecture.

[10]. It is the same chatbot framework used and developed by GitHub Inc., a tech company that has introduced and pioneered ChatOps. While the bot instance is generally chatplatform independent, in our case it is configured to connect with Telegram service. All users despite of its role (admin, lecturer, student) are connected via Telegram client. In the future versions, the system may have other components such as a dedicated mobile application and an administrative web interface. The bot instance accesses and persists data via internal web services. A static web server is used to host images and documents resulting from rendering actions, some of which require external services e.g. GitHub Pages, GitHub Gist or Google Docs.

Each module in LTKA-Bot provides services. These services may be used for certain workflows. An example of workflows for task assignment is given in Fig. 4. In this case, lecturer shall first compose and publish a task. The task must be conceived at AG1 stage and prepared considering target measures and outcomes. Once it is published and assigned, students might check the task, complete it and fulfill all requirements. During this process they might utilize Group module and Log module services. In the next step, they submit the work via Submission module so that it can be checked by the lecturer. Some minor variations are not displayed in the figure which include system validation, cancelation and re-

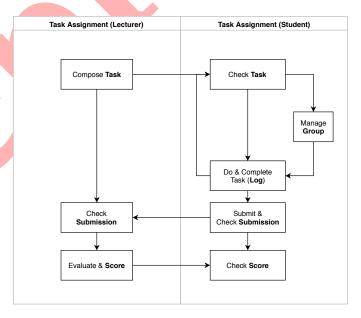


Fig. 4. Example of workflows implemented for task assignment.

submission. The last step involves the Score module, where the lecturer entry and approve the scores and the students can check their achievement.

From implementation perspective, each module consists of

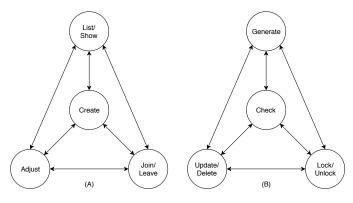


Fig. 5. Diagram of actions provided in the Group module.

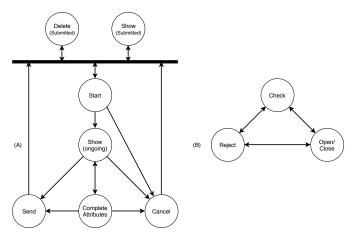


Fig. 6. Diagram of actions provided in the Submission module.

actions (group of services). Those can be either independent to each other or require certain workflows. An example is given in Fig. 5. Fig. 5 (A) shows Group actions available to students. The actions are used for self-service group management. Every students can create groups. Once a group is created, other students can join. If it is decided to rearrange groups, students may leave their current groups and joint another ones. Every group members may change/adjust group parameters (e.g. name, category). Fig. 5 (B) gives additional actions available for admin/lecturer to manage the groups. Lecturer may generate a number of groups with random members as a seed, which can later be modified and rearranged by the students themselves. At any time, lecturer can check categories, groups and their members, which can be useful for in-class activities. If groups in a category is used for task assignment and submission, lecturer can lock them to prevent rearrangement.

Another example for Submission module is given in Fig. 6. Fig. 6 (A) shows actions available to students. In opposite to the previous case for Group module, the diagram implies an internal workflow. The submission process is stateful and the system only accept one open submission process at a time i.e. if a student of a group has started the process, the system will block/reject other submission request from other groups till the active submission is sent or cancelled.

Similar to the previous case, Fig. 6 (B) displays additional actions for lecturer to manage submissions. Lecturer can check, reject certain submissions and open or close submission for particular tasks.

#### IV. DEPLOYMENT STUDY AND RESULT

A beta version of LTKA-Bot has been operating on the Telegram group of LTKA class in H1 2018, following a successful prototype test in late 2017. LTKA stands for Layanan Tersambung dan Komputasi Awan (Connected Services and Cloud Computing). It is a course offered to final year students at the School of Electrical Engineering and Informatics, ITB. Content of LTKA course covers many aspects in modern connected services, so that it fits perfectly for an application of advanced, state of the art tools such as LTKA-Bot. Many features implemented in the bot's Support module are intended for demonstrations and experience-based learning purposes. In the following we briefly highlight some issues and insights resulting from our trials and beta deployment.

## A. Report Documents

As mentioned in Section I, automating tasks for course reporting that is complied with ABET requirement was a major motivation for our work. At the center of this functionality is the Score module. The scoring system has to be structured according to outcomes/measures and applied to every task/exam. Once once score data for every measure is available, documents will only be the matter of a rendering procedure. An increased level of automation can be achieved when we are able to lower the number of manual entry of individual score values. Sure enough, it can not be avoided entirely and may strongly depend on the defined measures. However, some type of exams/tasks for particular measures can be scored automatically once they are done or submitted. This includes multiple-choice based online exam (e.g. via Google Forms) or individual contribution measure by processing activity data recorded via Log module.

As of the current state, LTKA-Bot has a functional Score module which is able to support both manual and automatic scoring. Document rendering procedure is in the work and shall be ready for use in the next LTKA class. Note that portfolio documents might hard to completely automate at a ready-to-submit level since there are some custom texts shall be added by each instructor. The output document of Report module for ABET purpose will be a filled *raw assessment sheet* ready to be included in the portfolio document.

# B. In-Class Activities

For in-class activities, Session and Support modules are of notably used. Session module includes functionalities to record session data (e.g. topic, schedule, venue, etc.) and services related to attendance (e.g. claim, stats, etc.). Support module covers general services, some of which are used for demonstration purposes. For example there are services to instantly render data (entried via Telegram) as a gist or as a web page served at GitHub Pages. While this can be seen as a

general feature, in our context it is an intended demonstration of *integration* and *auto deployment* in the current landscape of modern connected services.

## C. Off-Class Activities

A major benefit of bots is that it can be set to work 24x7. Basically all modules provided by LTKA-Bot are active all the time, so they are available off-class. But some may need activation to be usable. Such modules include the previously mentioned Session module e.g. to record students' attendance. Activation is authorized to the lecturer or selected admins.

Group, Submission and Log modules are intended to support students off-class when they do group assignments or projects. In the following we highlight a case of interaction that might happen when students are using Group services of LTKA-Bot. The commands displayed below are Group services available for student role (invokeable via help group).

```
.group new <GNAME>
.group join <GNAME>
.group leave <GNAME>
.group set <GNAME> category <CAT>
.group rename <OLD> <NEW>

.mygroup
.group show <GNAME>
.group list all
.group list cat <CAT>
.group gist <CAT>
.group gist <CAT>
.group gist <CAT> static
.group gist <CAT> send
```

An example of the conversation involving two students S1 and S2 with the bot is illustrated in Fig. 7. This can be explained as follows.

- Student S2 wants to join a group named mars, which was non-existent yet. The bot replies with the message that mars does not exist.
- Student S1 creates a group named *mars*. As there is no group with this attribute yet, the bot creates ones and replies with the message that *mars* has been created.
- Lacking of coordination, S2 wants to create *mars* group. The bot replies that the group exists already.
- S2 wants to join *mars* group. The bot accepts the request and replies with a success message.
- A new created group is by default assigned to category default. S1 changes this property to planet which is accepted by the bot.
- The students agree to change the group's name to *earth*. S2 sends the message to the bot and it accepts.
- Both students want to check to which group they belong to. They send the message and the bot replies with the corresponding information.

Groups are used for tasks which include measures for teamwork. Groups can initially be arranged by lecturer with a predefined category associated with the task via generator

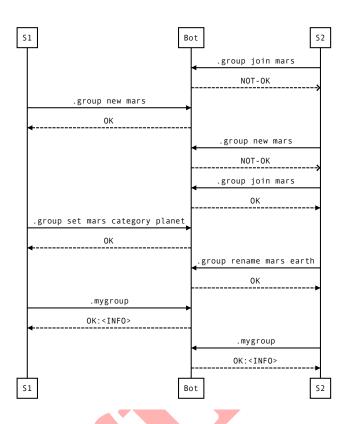


Fig. 7. Example of bot interaction (Group module).

service of Group module. Lecturer may allow students to rearrange and change some attributes other than category. The category attribute is used to track distribution status of students for the corresponding task.

Finally, Fig. 8 shows some screenshots featuring LTKA-Bot for other usecases. Fig. 8 (A) and (B) highlight a group chatroom where lecturer, students and the bot interact with each other. In Fig. 8 (A) lecturer instruct the bot to show submission summary and use the information provided by the bot to demand an action from some students. In Fig. 8 (B) a student request LTKA-Bot for his final score (this feature can be disabled for group chatroom, if desired). In Fig. 8 (C) the lecturer ask the bot to display summary of all given tasks (private chatroom).

## V. CONCLUSION AND FUTURE WORK

In this paper we introduce LTKA-Bot, a ChatOps based virtual assistant that can help and facilitate lecturers and students in all course-related activities such as: task assignment, submission, group, logbook, session, outcomes/measures, and scoring management. We described our approach to designing LTKA-Bot, presented our usecases and shared the lessons that we learned from our current trial and deployment. It has shown the potential in streamlining and automating many processes and thus lifting up higher education to the next level.

While LTKA-Bot is currently functional and in use, it is still under active development to make it a continuously better tool. Our future work will focus on the following major

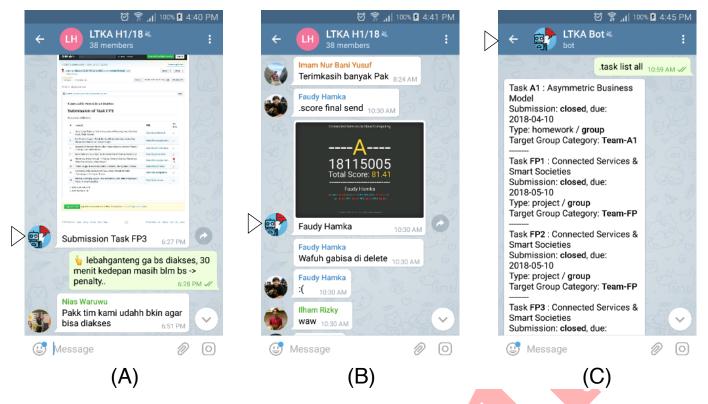


Fig. 8. Example of chat involving LTKA-Bot in group and private chatrooms.

aspects: adjusting existing features based on trial/ deployment results, adding more features to support and streamline more cases, and building more components to increase user and administrative experiences.

# REFERENCES

- [1] R. M. Felder and R. Brent, "Designing and teaching courses to satisfy the ABET engineering criteria," Journal of Engineering Education, 92.1, pp. 7–25, 2003.
- [2] A. K. Goel and L. Polepeddi, "Jill Watson: A virtual teaching assistant for online education," School of Interactive Computing Technical Reports, Georgia Institute of Technology, 2016.
- [3] T. H. K. Huang, J. C. Chang, and J. P. Bigham, Jeffrey, "Evorus: A Crowd-powered Conversational Assistant Built to Automate Itself Over Time," 2018.
- [4] S. Popenici and S. Kerr, "Exploring the impact of artificial intelligence on teaching and learning in higher education," Research and Practice in Technology Enhanced Learning, vol. 12, no. 1, 2017.
- [5] C. Toxtli, A. M. Hernandez, and J. Cranshaw, "Understanding chatbot-mediated task management," Proceedings of CHI Conference on Human Factors in Computing Systems, 2018.
- [6] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, M. Zhou, "SuperAgent: A Customer Service Chatbot for E-commerce Websites,", Proceedings of ACL, 2017.
- [7] J. Newland, "ChatOps at GitHub", Speaker Deck, 2013. [Online]. Available: https://speakerdeck.com/jnewland/chatops-at-github. [Accessed: 09- Jun- 2018].
- [8] R. Zyane, "How ChatOps can help you DevOps better," Chatbots Magazine, 2018. [Online]. Available: https://chatbotsmagazine.com/how-chatops-can-help-you-devops-better-5-minutes-read-507438c156bf. [Accessed: 28- May- 2018].
- [9] A. Guarnera, "Why and How to Automate Your Teaching", Insidehighered.com, 2015. [Online]. Available: https://www.insidehighered.com/blogs/gradhacker/why-and-how-automate-your-teaching. [Accessed: 09- Jun- 2018].

[10] HUBOT, [Online]. Available: https://hubot.github.com/. [Accessed: 10-Jun-2018].