ISSN 1870-4069

Da Capo: A Proposal for a BDI-Based Tutorial for Teaching Music

Alberto Nieto-Rocha, Wulfrano Arturo Luna-Ramírez

Universidad Autónoma Metropolitana,
Cuajimalpa,
Mexico

2163071736@cua.uam.mx, wluna@correo.cua.uam.mx

Abstract. The use of digital educational tools allows more people to have access to educational content, they have a wide range of resources and help to meet the demand for education. Learning music can provide various health benefits, such as improving mood. One of the ways to create a tool to help teach music is thanks to AI, in particular with tutoring agents. This paper presents a proposal of a multi-agent system making use of cognitive agents under the well known Belief-Desire-Intention paradigm based on a successful integration of web development tools such as *Flask* to create a web-based system for teaching basic music concepts.

Keywords: E-Learning, AgentSpeak, music tutor agents, BDI agents, multiagent system design, python, flask.

1 Introduction

Technology tools have solved some issues generated by social distance, noteworthy is teaching and e-learning. Most of the schools in the world have opted for virtual activities due to the increase in the use of internet and digital platforms that allow contents to be taught, homework to be assigned, homework to be reviewed, among other educational tasks.

Online activities proved to be very attractive to the population was those related with arts. Is in this context where we identify the need to create a music tutor agent for both to exploit the digital resources and connectivity, and to overcome the limited capacity of live human interaction.

Additionally, learning music can be beneficial for health, in a study done by [13], basic music theory and piano lesson were given for four months to a group of older adults who had no knowledge of the subject. The research determined that learning music and the cognitive processes involved can help combat depression and promote a better mood.

This paper presents one way to make use of AI and in particular tutor agents to support music students who would like to reinforce their basic music knowledge and skills. Using a web-based system, this present concepts and practical exercises. This tool is an intelligent music tutor called *Da Capo*.

2 Preliminaries and Related Work

Musical formal education has been oriented to traditional teaching, where teachers and students meet to realize the act of teaching/learning. Technology has let innovation on this type of act. For example, some music school has given formal and informal course of music during summer. Activities pursued in this course imply to make video and audio recording while students do some exercises. With this file, the teacher can do the evaluation and provide feedback to their students.

One of the problems of this kind of course, as mentioned in [2], arise with the high demand and the difficult to serve so many students, teachers can be overwhelmed by the workload. This can result in not providing adequate feedback to each student. One solution to this difficulty is the use of software that allows students to make use of tools that are capable of acting as tutors [1]. These tools have the particularity of being able to provide personalized and instant attention to each student, which allows students to be able to advance at their pace and helps to prevent gaps in the understanding of any subject.

Considering this, we can contemplate for a first version of the Da Capo system, the possibility of showing different types of content such as texts, audios, images, and videos. With these, the student can be presented with various exercises such as identifying concepts, improving his musical ear and learning about the history of music. It should be clarified that this article does not contemplate the design of exercises to teach music theory, since to do so, at least one person with expertise in music pedagogy is needed.

In this article, we present a system which, with adequate music content, can be used by people who wish to reinforce their musical knowledge and skills. In this section, we discuss the technologies that take on the role of the teacher and the methods used to develop them [1, 4, 11, 7].

2.1 Agents

Agents are informatics system that are located in an environment, they are capable of perceiving changes and interacting autonomously and persistently according to these changes. Their existence summarizes in the following characteristics:

Reactivity can be considered as the perpetual interaction with its environment, which forces it to respond to the changes that arise. To be able to respond adequately to changes, it needs to be endowed with initiative; this characteristic is known as *proactivity*. Since their behavior cannot be a product of chance, agents must be *autonomous*, i.e., they must base their behavior on the measure of their experiences. This implies that the agent does not rely solely on the knowledge with which it was created; the more it bases its behavior on learning and on the knowledge incorporated, the more autonomous it will be.

Social ability in agents is described as their ability to interact with other agents and even humans, they must use certain languages and protocols to establish accurate communication, since with this, they can achieve certain goals that can only be achieved with their peers, sets of these are known as multi-agent systems.

Persistence is expressed as the constant functioning of the system. There are different perspectives for designing agents. In the following, we will discuss two approaches in particular, tutor agents and BDI agents [9].

2.2 Tutor Agents

The intelligent tutor agents guide the student in the learning process; providing constant feedback, through evaluations the intelligent tutor agent chooses the topics in which the student presents some learning deficit. Some of the research that has been carried out on the subject indicates that intelligent tutor agents have four components: the knowledge base, the student model, the tutor model and the user interface [4, 5, 1]:

– **Knowledge Base**. Stores the materials that are taught in the agent, these can be: sounds, images, texts, among others. It also contains the solutions and explanations of the topics.
– **Student Model**. Manages the information about students, such as the access credentials, the activities that the student performs, the errors and successes about the topics seen.
– **Tutor Model**. It is responsible for deciding, based on the information in the student's model, the topics that will be presented to the student. These topics are the ones found in the knowledge base.
– **User Interface**. It is the connection between the tutor system and the student. It shows the exercises and materials pleasantly. It is necessary that this component is refined to provide adequate feedback to the student, as this will help the student to understand the failures he has. Ease of use should also be considered, since the user should concentrate on the topics presented, not on how to use the platform.

To create an intelligent music tutor agent, the tutor agent components should be taken as a base, to which other components that the developer considers necessary can be added. The knowledge base should be endowed with contents related to the musical field. For example, in the article related to WMITS, [11] discusses that, in addition to the tutor agent components, they use third-party packages that interface directly with the tutor model.

2.3 BDI Agents

The BDI model, in the words of García et al. [6], is a promising architecture. This is thanks to their capacity for practical reasoning, which in other words is the process of deciding what action to take to achieve goals. The BDI architecture is based on beliefs, desires, and intentions[14], and integrates Dennett's and Bratman's positions. They are: intentional approach and Plans and interactions and practical reasoning, respectively. This architecture has been extensively tested in a diversity of systems ranging from manufacturing to space navigation.

Beliefs are the information that the agent knows about its environment; they are actualized by perceptions and intentions. Desires are the agent's goals. Intentions are stacks (like data structures) of executing plans, the latter are stored in a plan library.

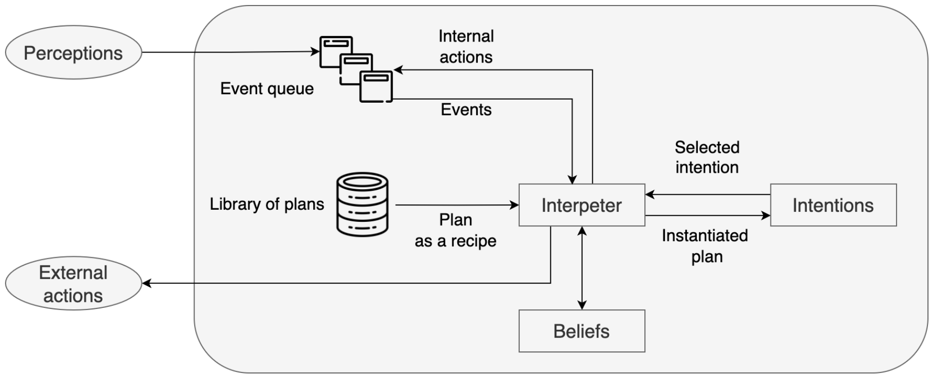


Fig. 1. Architecture of a BDI agent. Adapted from [8].

Plans are skills or procedural knowledge that the BDI agent has at its disposal to achieve certain states. Within plans are actions, which can update beliefs or modify the environment [8, 10]. In Figure 1, the above is shown.

2.4 Related Work

Some related work with agents and education has in common that used agents of type cognitive or reactive. In this section, we present some of the systems that make use of agents to teach (*Slang*, *Maestoso*, *WMITS*) or whose teaching area is music (*EarMaster*).

Maestoso is a tool focused on people without knowledge about music, it's an intelligent tutoring system that using an optical pencil that allows to make different exercises of writing notes. These are automatically graded by the system, which provides feedback and hints on what to do in the exercises [3].

Slang is an application focused on teaching English, created at MIT. It makes use of the tutor agent components described in section 2.2, to present an education with constant, unique, and customized feedback. The system allows the learner to perform activities such as recording sentences, identifying objects by name, spelling words, matching words with their meaning, etc. [12]

WMITS(Web-based Music Intelligent Tutoring Systems) is a web-based system for teaching music. This system teaches different music concepts such as intervals, chords, and harmony. According to the author of the system, to display the content, intelligent tutor agents are implemented to define the content to be displayed to the user. Since the research was conducted in 2007, the system was somewhat limited by the Web development technologies of the time. The development of the system seems not to have continued, since it is not possible to find it [11].

EarMaster is a paid music theory trainer with more than 2500 interactive lessons, covering the fundamental aspects of ear training, sight-singing and rhythmic training. To use the tool, you make use of the device's peripherals such as the keyboard, headphones, microphone, touch screen and touch screen to carry out the different activities it proposes.

The activities can include saying a note out loud and the system detecting if it is the correct note, identifying chords, identifying rhythms, etc. It should be noted that the system shows constant feedback, which lets you know exactly what you are doing. It is not specified if the system uses some kind of intelligent tutor to select the content that the user will see.

In a test performed I could detect that when doing several exercises wrong, the system put them again, also when doing several exercises correctly the system finished the lesson. This could indicate that there is a certain component that may or may not be an intelligent tutor [5].

Those systems mention before are tools designed to teach, and they use agents to make that, but the principal difference, in respect to Da Capo, is the use of BDI agents for improving the learning process. With BDI agents, Da Capo can display different types of content depending on the user's needs, i.e., it is not just a software to show exactly the same lesson to all students. Other difference it's that Da Capo it's a free software, and use more modern technologies like AgentSpeak or Flask.

3 Da Capo System Overview

This section describes the architecture of the system *Da Capo* and the three agents that conform it, and also explains how the interaction between the components is. For the development of the system, different technologies were used: Python, *Flask*, *MySQL*, *Python-Agentspeak* (alternative to *Jason*), *WSGI*, *HTML*, *CSS*, *JavaScript*.

We use *Prometheus* methodology [10] to design and develop the system *Da Capo* as it stands out among other software development methodologies for supporting the creation of intelligent and BDI agents. Figure 2, shows a diagram elaborated as part of the *Prometheus* methodology.

It shows the three agents that compose the system (`Teacher`, `Librarian`, and `School Control`) with their respective actions (rectangular trapezoid) and perceptions (star), according to [10], the actions are ways in which the agent can operate in the environment (in *Da Capo*, the environment is *Flask*) and the perceptions are the relevant information coming from web requests. A brief description of Da Capo agents is presented as follows:

- **Teacher**: It is responsible for interacting with students, providing content according to their needs. It constantly applies evaluations on the academic achievement of students. He also communicates with $SchoolControl$ to send a record of the student. It establishes a connection with $Librarian$ to request the contents to be shown to the student.
- **Librarian**: It consults the database, but acts under the orders of the teacher agent. When changes in the content needs to be done, this agent will be modified in the database.
- **School Control**: It has the academic information about the students, it will identify and indicate to the teacher agent which student arrive and will provide relevant data to this for the request of a lesson.

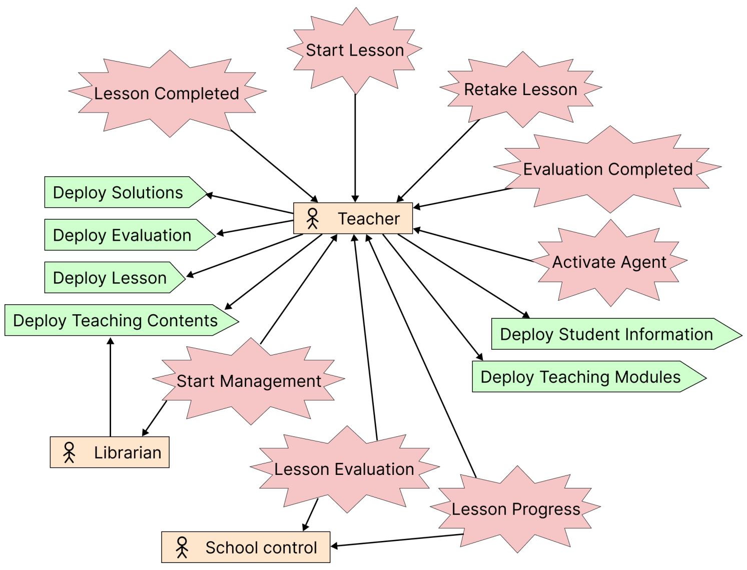
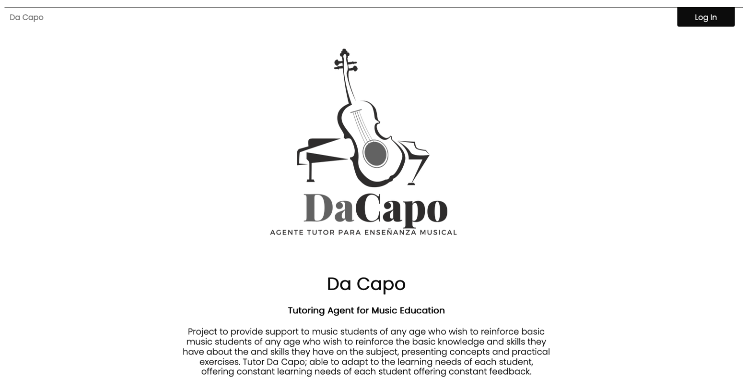


Fig. 2. System Overview Diagram from Prometheus methodology [10].



Da Capo

Tutoring Agent for Music Education

Project to provide support to music students of any age who wish to reinforce basic music students of any age who wish to reinforce the basic knowledge and skills they have about the and skills they have on the subject, presenting concepts and practical exercises. Tutor Da Capo; able to adapt to the learning needs of each student, offering constant learning needs of each student offering constant feedback.

Fig. 3. Main page of *Da Capo* displayed into a Flask environment.

Flask, a framework developed in Python, was used to build the website. It uses *Jinja2* as a web template engine and the *WSGI* interface for the web server to call a Python program.

Three Python codes were written, app.py for *Flask* related, $agent.py$ for *Agentspeak* related, and $database.py$ for database queries. An *ASL* file was also written, which contains the agent procedures. In addition to the three mentioned files, there is a fourth one called variables.py, this one contains global variable declarations, which are used to communicate *Flask* with *Agentspeak*. In Figure 3 the main page of Da Capo system is displayed.

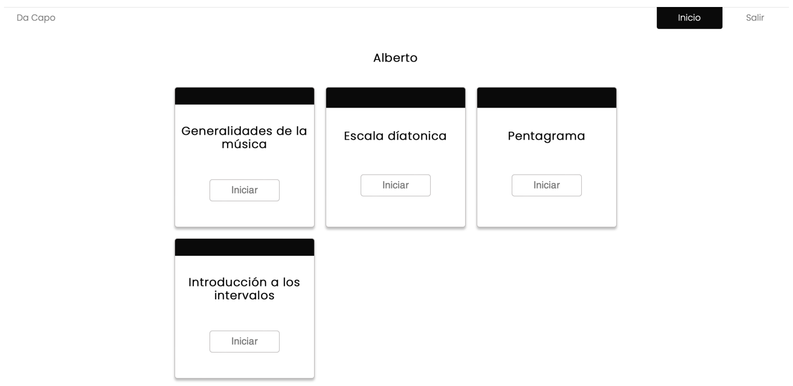


Fig. 4. Student's profile, with their teaching modules.

Figure 4, shows a user's profile, in which you can see the educational modules that the user can access. When entering these, the lesson (in Spanish) is presented.

Figure 5, shows how the different parts of the system interact to display a complete lesson. In general terms, the user must choose the module to be studied, the agent queries it and displays the corresponding lesson. Once the user has finished the lesson they should request the evaluation, again the agent queries and displays the request. At the end of the evaluation, the agent receives the user's answers, grades them and updates the user's profile.

Figure 5, shows the interactions between all the components of the system. But in terms of the BDI agent, the process to consult a lesson is the following. Using as a reference, Figure 1. The agent perceives an event (generate by the user in the web page) this event enters to the event queue, the interpreter deliberates the best option (plan) to achieve the activity (desire), this process is called intention. Once the agent chose what to do, the plan begun to execute. In this case, the plan is called "lessonConsultation", this is conformed for two steps: consult lesson, and show lesson.

The first step executed an internal action in *AgentSpeak* query *"x"* lesson. As mentioned in section 2.2, one of the components of Tutor Agent is *the knowledge base*, this stores the contents that the agent will teach, in our case materials related to the field of music.

This knowledge base are the beliefs of our agent. Once the results are ready, the agent makes the step two *show lesson*, in this case it's an external action. The agent sends the lesson to *Flask* and this last one is responsible for displayed the lesson to the user. The latter is the logic behind the BDI agent, which can be applied to different activities such as evaluating the lesson or showing the correct answer of the evaluation, as long as there are plans for them.

4 Conclusions and Future Work

Technology has revolutionized the way that people learn. This article presents the design of a multi-agent system that takes the activities that a teacher performs and makes a system do them.

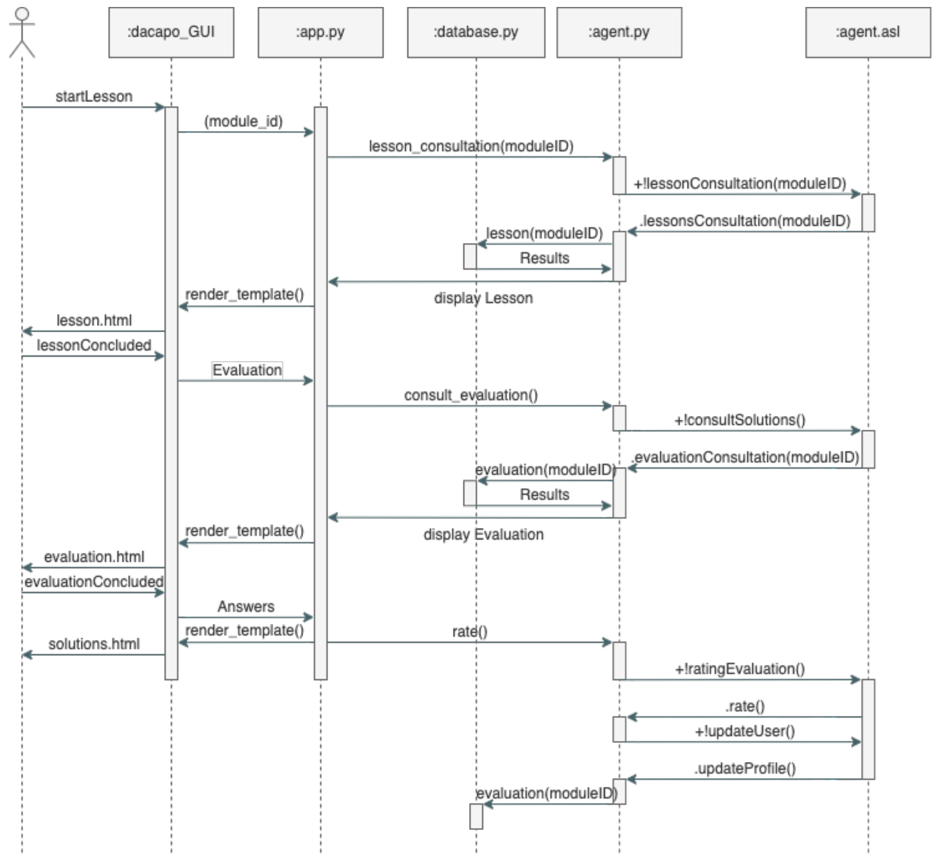


Fig. 5. Interaction Diagram for *Da Capo* Lesson Consultation.

This is not a replacement for a music teacher, it is a tool to help reinforce knowledge. Creating this tool had its difficulties, since there was no precedent that integrated *Flask* and *AgentSpeak*, although both are created in the same language, plus what it means to integrate *MySQL*, *HTML*, *CSS*, *JavaScript*, *WSGI* and the design of the agent.

For future work it is necessary to contemplate other tools that allow to communicate to the agents since *AgentSpeak* does not solve adequately this part, for this reason, the prototype of the system only counts with one agent that pursue all the functionalities.

It is also necessary to add more predicates to the agent to exploit the potential of the BDI architecture. It would also be interesting to enhance agent capabilities for retrieving musical information from dedicated storing sites to create the lessons.

Finally, it is essential to design and create musical content with assistance of professionals in the area, this will allow us to carry out tests with users with which we can pedagogically evaluate Da Capo and what is related to human-computer interaction.

Aknowledgments. Supported by UAM-Cuajimalpa, División de Ciencias de la Comunicación y Diseño, Licenciatura en Tecnologías y Sistemas de Información.

References

1. Apaydinli, K.: Intelligent tutoring systems in music education. In: 2nd International Conference on Future of Teaching and Education (2020)
2. Atiaja-Atiaja, L. N., Guerrero-Proenza, R. S.: MOOCS: Problems and challenges in higher education. In: International Conference on Advances in Education, Teaching & Technology, pp. 82–88 (2016)
3. Barreto, L., Taele, P., Hammond, T.: A stylus-driven intelligent tutoring system for music education instruction. Revolutionizing Education with Digital Ink: The Impact of Pen and Touch Technology on Education, pp. 141–161 (2016) doi: 10.1007/978-3-319-31193-7_10
4. Da, S.: Research on the design of online intelligent tutoring agents. In: 2010 International Conference on Electrical and Control Engineering, pp. 4755–4758 (2010) doi: 10.1109/iCECE.2010.1151
5. Earmaster-music theory and ear training on PC, Mac, iPad and iPhone. https://www.earmaster.com/es/company/press-material.html
6. García, D., Simari, G., García, A.: Planificacion en agentes BDI. In: VI Workshop de Investigadores en Ciencias de la Computación (2004)
7. Latham, A. M., Crockett, K. A., McLean, D. A., Edmonds, B., O'Shea, K.: Oscar: An intelligent conversational agent tutor to estimate learning styles. In: International Conference on Fuzzy Systems. pp. 1–8 (2010) doi: 10.1109/FUZZY.2010.5584064
8. Luna Ramírez, W. A.: Agentes cognitivos, intencionales y BDI (2021)
9. Luna Ramírez, W. A.: Botzoologia: Un bestiario de agentes artificiales (2021) https://www.uv.mx/cienciauv/blog/botzoologiaunbestiariodeagentesartificiales/
10. Padgham, L., Winikoff, M.: Developing intelligent agent systems: A practical guide (2004) doi: 10.1002/0470861223
11. Phon-Amnuaisuk, S., Siong, C.: Web-bases music intelligent tutoring systems. Interactive Multimedia Music Technologies, pp. 231–248 (2007) doi: 10.4018/978-1-59904-150-6.ch011
12. Rojas, N.: Queremos contarte en qué consiste el adaptive e-learning de Slang y cuáles son sus ventajas. https://es.blog.slangapp.com/queremos-contarte-en-qu\%C3\%A9-consiste-el-adaptive-e-learning-de-slang-y-cu\%C3\%A1les-son-sus-ventajas-f62138d69690
13. Seinfeld, S., Figueroa, H., Ortiz-Gil, J., Sanchez-Vives, M. V.: Effects of music learning and piano practice on cognitive function, mood and quality of life in older adults. Frontiers in Psychology, vol. 4 (2013) doi: 10.3389/fpsyg.2013.00810
14. Wooldridge, M. J.: Reasoning about rational agents. MIT Press (2000)