Olympiads in Informatics, 2025, Vol. 19, 35–44

DOI: 10.15388/ioi.2025.03

Strategy and Tactics for Introducing Generative Artificial Intelligence into the Instrumental Distance Learning System DL.GSU.BY

Michael DOLINSKY

Faculty of Mathematics and Technologies of Programming, F. Skorina
Gomel State University, Sovetskaya str., 104, Gomel. 246019. Republic of Belarus
e-mail: dolinsky@gsu.by

Abstract. This paper provides the strategy and tactics for introducing generative artificial intelligence (GenAI) into the instrumental **distance learning** system DL.GSU.BY (hereinafter **DL**). The strategy consists of sequential implementation of the following stages of development: creating opportunities for convenient work with GenAI in the DL system; launching electronic GenAI students to automatically complete training courses in the DL system and comparative analysis of the achievements of various GenAIs among themselves and with real students; accumulation and dissemination of students' experience working with GenAI; improving the quality of training using GenAI by developing a system of preprompts for tasks and subjects; further personalization of training through the implementation of advanced techniques for using GenAI (active GenAI, Learning By Teaching). GenAI implementation tactics consistently and in detail describe the practical steps to implement the strategy.

Keywords: generative artificial intelligence, instrumental system for distance learning, DL.GSU.BY.

1. Introduction

The goal of initial programming training at the university is to develop Computational Thinking skills (Kaleem, 2024), which includes the following components: abstraction, decomposition, pattern recognition, algorithmization, debugging.

A significant increase in learning efficiency comes from the use of online platforms for teaching programming, such as EduCoder (Zhang, 2023; Li, 2023), HTProgramming (Figueiredo, 2021), Code4brownies (Phan, 2018).

A new stage in increasing the efficiency of initial programming training is associated with the introduction of generative artificial intelligence (GenAI). Text-based generative systems have a chatbot as the front end to interact with the user and are based on LLMs – generative models that can produce new content based on the data they are trained on.

The most famous LLMs: GPT (OpenAI), Gemini (Google), Llama (Meta), Claude3 (Anthropic). User interaction with a chat bot is implemented by requests. Herden (2024) formulated tasks for such chat bots during the initial programming learning process: explaining basic knowledge, constructing code, explaining code, refactoring code, formatting code, checking coding style, commenting on code.

The most effective seems to be the integration of GenAI into online systems for training and testing programs. TutorBot+ (Martinez-Araneda, 2023, 2024) is integrated into a WEB training system and an online program testing platform: it allows you to take the problems, obtain solutions in dialogue with GenAI, and check solutions. KOGI (Kuramitsu, 2023) is a learning support system that integrates ChatGPT and the Jupyter framework. KOGI helps the student get advice from ChatGPT in response to errors and questions. KOGI is implemented in two introductory courses: Algorithms, Data Science. As a result, there is a significant reduction in the number of unresolved student errors and teachers receive information about questions and errors.

IPSSC is an intelligent programming scaffolding system based on ChatGPT (Liao, 2024), where instead of direct interaction, students work through three structured modules: Solution Assessment, Code Assessment, and Free Interaction. In the Solution Assessment module, students decompose a complex problem into several simple ones and abstract them so that the solution is suitable for similar problems (thus, this module focuses on decomposition and abstraction). In the Code Assessment module, students design algorithms, write code, and continuously debug code, using the module to identify errors in the code, then fix them. ChatGPT helps improve algorithms and code. In the Free Interaction module, students can interact with ChatGPT directly, discussing topics that are not covered in other modules.

Coding Step (Sarshartehrani, 2024) is a web application designed to teach the basics of Python programming.

LearningProgrammingWithGPT (Abolnejadian, 2024) – an environment that is used with students in the CS1 course while studying Python. The course integrates ChatGPT as a means to support students and instructors in the classroom. The platform acts as a medium between students and their GenAI instructor, providing personalized educational material based on preprompts built into student requests.

The developers of CodeHelp (Liffiton, 2023; Denny,2024) claim that programming teachers have huge problems being able to devote enough time to each student. CodeHelp provides students with on-demand help without offering immediate solutions, an example of a "Socratic" tutor who avoids revealing solutions directly to the user.

CodeAid (Kazemitabaar, 2024) also avoids direct answers with code, trying to guide the student to solve the problem.

The developers of NotebookGPT (George, 2024) believe that direct use of ChatGPT can interfere with student learning, so NotebookGPT gives access to GPT, but does not return complete solutions, providing: feedback on programming style, explanations of how pieces of code work, help with debugging code, the ability to see alternative solutions to problems.

AI TA (Lee, 2023) helps students perform decomposition – that is, partitioning tasks into subtasks. AI TA supports the Subgoal learning, an effective learning strategy that helps students break down complex problems into subtasks.

StAP-tutor (Roest, 2024) provides recommendation of the next step to solve the problem. TeachYou+AlgoBo (Jin, 2023) offers an LBT (Learning By Teaching – teaching yourself by teaching someone). Here TeachYou is an LBT environment for teaching algorithms, and AlgoBo is a learning chatbot for problem solving.

2. Instrumental System of Distance Learning DL.GSU.BY

Since September 1999 at Gomel State University named after F. Skorina, under the leadership of the author, the instrumental distance learning system DL.GSU.BY (hereinafter DL) has been introduced and developed (Dolinsky, 2022a). The author has been using it all this time to teach programming to schoolchildren (starting from the first grade) (Dolinsky, 2016) and first-year students of the Faculty of Mathematics and Programming Technologies (Dolinsky, 2022b)., as well as to teach first- and second-year students the basics of digital electronics within the subjects Machine-oriented programming, and Computer architecture (Dolinsky, 2022c).

DL allows you to take problems and send solutions to these problems for testing in various programming languages, including Pascal, C++, Python, Java, C#. And when studying the basics of digital electronics, programs are sent in assembly languages and C microprogramming, as well as circuit diagrams of digital devices developed in the high-level design system HLCCAD (Dolinsky, 2022d). To support personalized learning for students with different levels of training, a tree-based learning system has been introduced, in which a student who cannot solve a problem has the opportunity to press a "I don't know" button and receive a tree of auxiliary tasks that will help him ultimately solve the original problem. However, on the one hand, this is a static, predetermined system. On the other hand, there are many test tasks for which there is no such built-in training system. The introduction of generative artificial intelligence can help in learning to solve these problems as well. In addition, for courses related to teaching the basics of digital electronics, there is no such learning tree at all. Further development the system of learning capabilities of the DL is associated primarily with its integration with GenAI.

3. Strategy for Introducing GenAI into the Instrumental Distance Learning System DL.GSU.BY

3.1. *Creating Opportunities for Convenient Work with GenAI in the DL System*

The basis for introducing GenAI into the DL system is a chatbot (#1), which receives student requests and sends them through the API provided by GenAI and receives responses, and then displays the responses to students. The presence of such a chatbot allows the use of special additional training preprompts for tasks, programming languages, topics and subjects, including those taking into account the individual characteristics

of a particular student. To explore the capabilities of GenAI from various companies, they are permanently added to our chatbot, and the student is given the opportunity to select a GenAI and then assess satisfaction with the result of working with GenAI with a score from 0 to 5. Accordingly, at the student's request, he is provided with a list of available GenAI in alphabetical order or in descending order of the average GenAI grade by students. GenAIs, which are worked interacting with our chatbot #1, are called built into the DL system.

Alternatively, students are given the opportunity to work with the same GenAI directly, without using our chatbot, to be able to develop the latter based on student suggestions. Such GenAI are called non-embedded.

3.2. *Launching e-Learners in DL Courses*

A special bot is being developed (#2), the parameter of which is name of the GenAI website. After manual registration on DL, this bot permanently performs the following work:

- Goes to the first available unsolved task.
- Takes the problem text.
- Sends it to GenAI with a request to return the solution to the problem.
- Receives a solution from GenAI.
- Sends it for testing on DL.
- If the solution passes, moves on to the next task.
- If the solution does not pass – takes from the DL website, a test on which the solution did not pass.
- Sends a test to GenAI with a request to correct the solution.
- Receives a corrected solution in response.
- Sends the corrected solution to the site DL.

This process continues until the solution passes or until the limit on the number of submissions of a solution to the same problem.

There are educational and control tasks. Educational tasks are open all the time, except for those classes when control tasks are open. The results of all e-students appear in the special result tables: only e-students – to compare GenAI sites with each other based on the success of solving problems; together with students – to compare the results of e-learners with the results of students.

For each problem, the following marks are saved: not solved, solved at what attempt, by which GenAI.

3.3. *Accumulation of Student Work Experience*

Insert call icons to our chatbot, which supports working with built-in and non-built-in GenAI, into the DL in the menu of the site, each course (subject), each task. As students

work with the GenAI chatbot, a protocol is kept that records the date and time of work, the student, the task, and the student's assessment of the quality of the assistance received. In parallel, for the built-in GenAI, the entire dialogue is saved in the file system. For non-embedded GenAI, the student is asked to save the dialogue in the cloud and enter a link to it in a special field. DL will automatically copy the dialog file to its file system. Links to each such dialogue are provided directly on the page of the task for which this dialogue was conducted. In addition, pages are created with links to all available dialogues, both on tasks and on general questions. Search by keywords is provided. In this way, students gain experience working with GenAI. And a student can receive help without directly contacting GenAI, if such help has already been provided to another student before and turned out to be effective.

3.4. *Improving the Quality of Teaching by Developing Preprompts*

A special direction of development is the permanent development and improvement of the system preprompts by tasks, courses, programming languages. Such preprompts are additional information that is sent before the student's request to improve the quality of the response to the student. Such preprompts are developed primarily by the students themselves. In addition, students are provided with direct access to all preprompt information so that they can use it interactively when working with non-embedded GenAI.

3.5. *Personalization of Learning*

Currently, the author sees two directions for the development of personalization:

(1) The "active GenAI" mode, when, having received the condition of the problem, it does not immediately give a solution, but tries to consistently lead the student to a solution, asking clarifying questions that help GenAI find out what exactly the student does not know and, accordingly, eliminate this ignorance. At the same time, the student, under the guidance of GenAI, consistently performs abstraction, decomposition, pattern recognition, algorithm development, coding and debugging for the task.
(2) Mode LBT (Learning By Teaching) – when a student is asked to teach a specially tuned GenAI to solve problems of a given type. The same mode can be used to research and develop pre-prompts on tasks, topics and programming languages.

3.6. *Entry from the other End*

Since 1999, the DL system has been accumulating problems, their solutions by students, and the corresponding entries in the protocol: pass or fail, and if not pass, then on what test. These solutions are of high-level programming languages, Intel 8086 assembler,

C-MPA microprogramming language, digital device diagrams – represented by projects – in graphical or text form. This data can be used for direct LLM training.

4. Tactics for Introducing GenAI into the Instrumental Distance Learning System DL.GSU.BY

1. Create opportunities for convenient work with GenAI in the DL system:

- Launch the page `http://dl.gsu.by/ai/chat` – this task has already been completed.
- Create and expand the list of sites accessed via API (currently, there are four such sites).
- Create and expand the list of sites where students can work interactively after logging in through `http://dl.gsu.by/ai/chat`.
- Develop a page for adding new sites to the second list (by entering a URL).
- Allow students to rate their experience with sites from both lists (using a 0–5 star rating system).
- Display site lists both alphabetically and sorted by descending ratings.

2. Accumulate of student work experience:
GenAI call icons are built into the menu of the DL website, DL course, and tasks in the DL course. Students are provided with the opportunity to:

- Select GenAI for dialogue from the list provided (ordered alphabetically, the current rating of students, by the number of tasks passed using this GenAI).
- Replenish the AI list with new URLs with starting comments (the system administrator has the ability to edit and delete information entered by students).
- Ask questions about tasks.
- Ask any questions (for clarification).
- Leave your comments/evaluations of the work – search and view dialogues on tasks and answers to questions.

The system automatically accumulates links to task dialogs, providing ordering and search by task number. CC also accumulates lists of dialogues on general issues, sorted alphabetically and searched by keywords. Dialogs are stored as doc files in a special folder Dialogs.

GenAI is called by following the link `http://dl.gsu.by/ai/chat` with the transmission of any combination of the following parameters, depending on the already available information:

AI_ID = GenAI identifier
Course_ID = course (subject) identifier
Task_ID = task identifier
Language_ID = programming language identifier (the default is taken from the task page, the student is also given the opportunity to select another programming language from the proposed list).

To support the operation of the chatbot, information about courses, programming languages and users is copied from the database (DB) of the DL system:

Courses: <Number> <Name>
Languages: <Number> <Name>
Users : <Number> <personal information as in DL>

In addition, special database tables are created

AI_ID <Number> <URL><Indicator built-in/non-embedded)><Name>
Themes <Number> <Subject of the request to ID>
(for subsequent search and/or display
in the list of topics by alphabetical order/relevance>
References <Number> <file name in the Dialogues folder>
AI_LOG <Date><Time><AI_ID> <Refs_ID (dialogue link)> <(0–5)
User Rating> [User_ID]
[Course_ID] [Task_ID] [Theme_ID]

During work in the database, a log of work with sites is kept in the format

<Date><Time><AI_ID> <Refs_ID (link to dialogue)> <(0–5)
User Rating> [User_ID] [Course_ID]
[Task_ID] [Theme_ID]

[] mean that this information may absent

All dialogues are saved in the file system in the Dialogues folder, in subfolders

<Task_ID> <Language_ID>

The dialog file names are as follows:

<UserID> <AI_ID> <Date><Time><User Rating>.doc

When working with built-in GenAI, these files are created automatically by the system.

When working with non-embedded GenAI, the student is asked to manually create a dialogue doc file in the cloud, and then enter a link to this file, as well as select dialogue metadata (Task_ID, Language_ID) or enter the topic of the dialogue.

If a student requests a task with previously saved dialogues, a list of them is first shown for viewing. (together with user rating in descending order of ratings). In addition, it is a supporting list of saved dialogs by task / topic / language.

3. Improve the quality of teaching by developing preprompts

A folder Preprompts is created in the file system with files Course_ID.txt, Task_ID_.txt, Language_ID.txt.

When working with the built-in GenAI, suitable preprompts are automatically sent to GenAI, anticipating student requests. When working with unbuilt GenAI, students have access to all preprompts to manually send preprompts.

Resources are available for editing preprompt files, developing and accumulating preprompt files for tasks, languages, courses by students.

In order to provide the chatbot with access to task texts, folders containing them are copied from the DL to the chatbot:

Tasks – html task texts
Images\original – pdf and doc – task texts

Make and add to the folder

Tasks-ext – txt-texts of tasks.
Replenish task txt texts by extracting from pdf and doc, if not found in Tasks and Tasks-ext.

Automatically include the task texts into the student's request for the task.

Preprompts are also automatically inserted depending on the task (language, course).

4. Launch e-learners in DL courses
The goal is to support the continuous operation of GenAI on DL: solving training problems, tests, team olympiads, possible with a limit on the number of attempts, followed by comparison of the GenAI results with each other and with the results of students. To ensure the principle of "do no harm" to the operation of the DL system, tests for tasks (Archives folder) and the DL database are copied to the chatbot. At first there will be a routine copying of information (for example, once a day at night), and in the future it will be incremental (that is, as the information changes).

Next development of DL API is needed:

- Log in with your account.
- Go to the course.
- Get a list of tasks to be solved (during training, control, by options).
- Get the problem text.
- Send the solution to the problem.
- Find out the verdict on the sent solution.
- Take the first failed test for the solution.
- Skip task.

E-learners are launched by universal bot #2, which receives AI_ID as a parameter.

5. Handmade
This work is done until it is automated in order to accumulate relevant experience:

- Create – expand the list of AI sites where you can ask for solutions to problems.
- Register AI sites on DL (such as AI URL instead of last name and first name).
- Comparison of the effectiveness of different GenAI (by manually sending solutions to the same tasks).
- Accumulation of dialogues and links to them – for solving problems.
- Accumulation of preliminary notes on tasks, languages, courses.
- Accumulation of tasks and GenAI most suitable for LBT (Learning By Teaching).
- Solving problems using these sites (every problem on each site).
- Write on the forum about students impressions.
- Posting a link to the task on the DL, the URL of the AI site, a link to the entire dialogue with the AI site.
- Accumulate a helper file for preprompts before sending the task conditions.

- Systematic work on pre-prompts for the most problematic tasks for students/e-students
- Training AI sites to "consistently teach students how to solve problems" – by individual types.

5. Conclusion

This paper presents the strategy and tactics for introducing GenAI into the instrumental distance learning system DL.GSU.BY. As a result, it is expected to create opportunities for convenient work with GenAI in the DL system; launching electronic GenAI students to automatically complete training courses in the DL system and comparative analysis of the achievements of various GenAIs among themselves and with real students; accumulation and dissemination of students' experience working with GenAI; improving the quality of training using GenAI by developing a system of preprompts for tasks and subjects; further personalization of training through the implementation of advanced techniques for using GenAI (active GenAI, Learning By Teaching).

References

Abolnejadian, M , Alipour, S., Taeb K. (2024). Leveraging ChatGPT for Adaptive Learning through Personalized Prompt-based Instruction: A CS1 Education Case Study. In: *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. ACM, New York, NY, USA, Article 521, 1–8.

Denny, P., MacNeil, S., Savelka J., Porter, L., Luxton-Reilly, A. (2024). Desirable Characteristics for AI Teaching Assistants in Programming Education. `https://arxiv.org/pdf/2405.14178v1`

Dolinsky, M. (2016) Gomel Training School for Olympiads in Informatics. *Olympiads in Informatics*, 10, 237–247.

Dolinsky, M. (2022a) Instrumental system of distance learning DL.GSU.BY and examples of its application. *Global Journal of Computer Science and Technology Interdisciplinary*, 22(1), 45–53.

Dolinsky, M. (2022b). Teaching Algorithms and Programming First Year University Students on Base of Distance Learning System DL.GSU.BY. *WSEAS Transactions on Advances in Engineering Education*, 19, 52–57.

Dolinsky, M. (2022c). Experience of Blended Learning the Fundamentals of Digital Electronics for First/Second Year University students On Base of Distance Learning System DL.GSU.BY. *International Journal of Education and Learning Systems*, 7, 59–64.

Dolinsky, M. (2022d). Tool HLCCAD for Blended Learning the Fundamentals of Digital Electronics. *International Journal of Circuits and Electronics*, 7, 47–55.

Figueiredo ,J., Garcia-Penalvo, F.J. (2021). Teaching and Learning Tools for Introductory Programming in University Courses. In: A. Balderas, A. J. Mendes and J. M. Dodero, Eds., *Proceedings of the 2021 International Symposium on Computers in Education (SIIE) (23–24 September 2021, Malaga, Spain)*. USA: IEEE.

George, S.D., Dewan P. (2024). NotebookGPT – Facilitating and Monitoring Explicit Lightweight Student GPT Help Requests During Programming Exercises. In: *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces (IUI '24 Companion)*. ACM, NY, USA, 62–65.

Herden, O. (2024). Integration of Chatbots for Generating Code Into Introductory Programming Courses International Conference on Future of education.

Jin, H., Lee, S., Shin, H., Kim, J. (2023). Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education. `https://arxiv.org/pdf/2309.14534`

Jin, H. , Lee, S., Shin, H., Kim, J. (2023). Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education. `https://arxiv.org/pdf/2309.14534v2`

Kaleem, M., Hassan, M.A., Khurshid, S.K. (2024). A Machine Learning-Based Adaptive Feedback System to Enhance Programming Skill Using Computational Thinking. IEEE Access https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10506466

Kazemitabaar, M., Ye, R., Wang, X., Henley, A.Z., Denny, P., Craig, M., Grossman T. (2024). CodeAid: Evaluating a Classroom Deployment of an LLM-based Programming Assistant that Balances Student and Educator Needs. https://arxiv.org/pdf/2401.11314

Kuramitsu, K. , Obara, Y., Sato, M.., Obara, M. (2023). KOGI: A Seamless Integration of ChatGPT into Jupyter Environments for Programming Education. In: *Proceedings of the 2023 ACM SIGPLAN International Symposium on SPLASH-E (SPLASH-E 2023)*. ACM, NY, USA, 50–59.

Lee, C. , Myung, J., Han, J., Jin, J., Oh, A. Learning from Teaching Assistants to Program with Subgoals: Exploring the Potential for AI Teaching Assistants. https://arxiv.org/pdf/2309.10419

Li, Z., Zhang, S., Sang, X. (2023). Exploration of Machine Learning Teaching based on the EduCoder Platform. *Journal of Education and Educational Research*, 4(3), 130.

Liao, J., Zhong, L., Zhe, L., Xu, H. , Liu, M., Xie T. (2024). Scaffolding Computational Thinking With ChatGPT. *IEEE Transactions on Learning Technologies*, 17.

Liffiton, M., Sheese B., Savelka, J., Denny P. (2023). CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. https://arxiv.org/pdf/2308.06921

Martinez-Araneda C., Gutierrez, M., Maldonado, D., Gomez, P., Segura, A., Vidal-Castro, C. (2024). Designing a Chabot to support problem-solving in a programming course. In: *INTED2024 Proceedings*, pp. 966–975.

Martinez-Araneda C., Valenzuela, M.G., Meneses, P.G., Montiel, D.M., Navarrete, A.S. Vidal-Castro C. (2023). How Useful TutorBot+ is for Teaching and Learning in Programming Courses: a Preliminary Study. In: *42nd IEEE International Conference of the Chilean Computer Science Society (SCCC)*. Concepcion, Chile, pp. 1–7.

Phan, V., Hicks, E. (2018). Code4brownies: an active learning solution for teaching programming and problem solving in the classroom. In: *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pp. 153–158.

Roest, L., Keuning, H., Jeuring, J. (2024). Next-Step Hint Generation for Introductory Programming Using Large Language Models. In: *Proceedings of the 26th Australasian Computing Education Conference (ACE '24)*. Association for Computing Machinery, New York, NY, USA, 144–153.

Sarshartehrani, F., Mohammadrezaei, E., Behravan, M., Gracanin, D. (2024). Enhancing E-Learning Experience Through Embodied AI Tutors in Immersive Virtual Environments: A Multifaceted Approach for Personalized Educational Adaptation. In: Sottilare, R.A., Schwarz, J. (eds), *Adaptive Instructional Systems. HCII 2024 LNCS, 14727*. Springer, Cham.

Zhang, S., Yang, J., Sang X. (2023). Exploring the Applications of EduCoder Platform in Blended Teaching for Computer Major. *Journal of Education and Educational Research*, 4(2), 100.

M. Dolinsky is a lecturer in Gomel State University "Fr. Skoryna" from 1993. Since 1999 he is a leading developer of the educational site of the University (dl.gsu.by). Since 1997 he is heading preparation of the scholars in Gomel to participate in programming contests and Olympiad in informatics. He was a deputy leader of the team of Belarus for IOI'2006, IOI'2007, IOI'2008 and IOI'2009. His PhD is devoted to the tools for digital system design. His current research is in teaching Computer Science and Mathematics from early age.