POLITECNICO DI TORINO

Corso di Laurea Magistrale in Management Engineering

Master of Science Course

in Management Engineering

Master of Science Thesis

Enhancing a Course-Specific AI Chatbot for Higher Education: Technical Improvement and Experimental Design for the AMPS Course



Supervisor

Prof.ssa Giulia Bruno

Candidate

Cadore Edoardo

Table of Contents

Abstract 5

1 Introduction 6

2 Generative AI Technologies and Educational Applications 9

2,1 Historical Evolution of AI 9

2,2 Principles of GenAI Operation 14

2,3 Core Components of an AI Chatbot 16

3 Platforms and Practices in Chatbot Development 20

3,1 Main Platforms for Chatbot Development 20

3,1,1 Hugging Face 20

3,1,2 LangChain 22

3,1,3 Rasa 23

3,1,4 Microsoft Copilot Studio 24

3,1,5 IBM Watson Assistant 25

3,1,6 Goolgle Dialogflow 26

3,2 Criteria for Selecting Language Models 27

3,3 Literature Review on University Chatbots 29

3,3,1 Overview of the sources 31

3,4 Summary of Findings 38

4 The AMPS Course Chatbot: Analysis, Limitations, and testing 42

4,1,1 Prompt 43

4,1,2 Knowledge base 44

4,1,3 Functioning 46

4,2 Comparative testing 47

4,3 Replication of the Chatbot System 52

5 Technical and Pedagogical Improvement 54

5,1 Replacement and Update of Knowledge Base Files 54
5,2 Document Parsing and Extraction 55
5,3 Integration of Course Video Lectures 57
5,4 Update of Prompt, Glossary and Bridge notes 59
5,5 Post Improvement Testing of the Chatbot 60
6 Experimental Design for Evaluating the Educational Impact of the AMPS Chatbot 65
6,1 Literature Review on the Evaluation of AI Supported Learning 66
6,1,1 Search Strategy and Keyword Selection 66
6,1,2 Overview of Selected Studies 67
6,1,1 Synthesis of the Literature 70
6,2 Design of the experiment 73
6,2,1 Objectives of the Experiment 73
6,2,2 Experimental Context and Educational Setting 74
6,2,3 Definition of the Intervention 76
6,2,4 Experimental Design and Group Assignment 77
6,2,5 Measurement of Learning Outcomes 78
7 Conclusions 81
Bibliography 82
Web Resources 86
Appendix 87
Appendix A 87
Appendix B 102
Appendix C 112

Abstract

The use of generative artificial intelligence (AI) in education is growing rapidly and it's offering new possibilities for student learning. This thesis focuses on the development of a domain-specific chatbot for the Analysis and Management of Production Systems (AMPS) course at Politecnico di Torino. The work started from an initial version of the chatbot, built with Microsoft Copilot Studio for another thesis project, and aims to test and modify the system to enhance the chatbot's capabilities.

The central goal of this thesis is to identify the best practices used in AI chatbot creation and use them to enhance the existing AMPS chatbot so that the new version can have a tangible impact on learning.

To achieve this goal the work started from a literature review of the existing papers regarding AI chatbots and agents, thanks to this analysis most used platforms and techniques were identified and presented one by one. Leveraging this acquired knowledge, it was possible to identify the existing chatbot critical issues and convert the best practices into actual improvements. Before that, the chatbot has been tested to assess the benchmark for the improvement measurement. The work presents all the modification applied to the system to improve it both from the technical and pedagogical point of view. These improvements have been compared to the pre modification benchmark to obtain a quantitative measure of the intervention impact.

The final step is to test whether the enhanced chatbot improves students' understanding and problem-solving abilities. The work presents a design of experiment (DOE) that follows a randomized control approach, ensuring a robust comparison between two groups approaching the same exercises with and without access to the AMPS chatbot tool.

This research contributes to the growing body of knowledge on AI tools in education, examining the improvements and customization possibilities of AI chatbots as educational tools. The findings will provide a structured approach to create and optimize AI chatbots for educational purposes and could serve as a reference for applications in other academic courses and disciplines.

1 Introduction

The rapid evolution of artificial intelligence technologies, and in particular the development of large language models, has significantly transformed the landscape of digital tools available for education. In recent years, conversational systems capable of understanding natural language and generating coherent responses have moved from experimental research prototypes to widely accessible instruments that can support learning, problem solving, and information retrieval. The emergence of generative artificial intelligence has therefore opened new possibilities for the design of educational tools that can assist students in navigating complex academic content and interacting with course materials in a more dynamic and personalized way. Within this broader technological transformation, chatbots have attracted increasing attention as potential instruments to support learning processes. A chatbot can be defined as a conversational system that interacts with users through natural language, either in text or voice form, and provides responses based on predefined rules, machine learning models, or generative artificial intelligence. These developments have encouraged the exploration of chatbots as educational tools within university environments. In academic contexts, chatbots can be used to provide support for frequently asked questions, assist students in navigating course materials, clarify theoretical concepts, and guide learners through problem solving processes. The potential advantages of such systems are particularly relevant in courses that involve complex analytical models or technical methodologies, where students often require continuous clarification and support outside traditional classroom hours. By offering immediate access to information and explanations, conversational agents can complement traditional teaching methods and potentially improve the overall learning experience.

Despite the growing interest in the application of artificial intelligence in education, the implementation of course specific chatbots presents several challenges. One of the most significant concerns relates to the selection of appropriate technological solutions for chatbot development. The current ecosystem of artificial intelligence platforms includes a wide range of alternatives that differ in terms of capabilities, costs, scalability, and control over data. Developers must therefore carefully evaluate the tradeoffs between proprietary solutions and open-source models, between fully customizable frameworks and low code development environments, and between systems optimized for conversational flexibility and those designed for reliability and domain specificity. Another important issue concerns the pedagogical value of these systems. While conversational AI tools can potentially assist students in accessing

information more efficiently, it remains an open question whether their use leads to a deeper understanding of the subject matter or simply improves task performance while the tool is available. In other words, it is necessary to distinguish between improvements in assisted performance and genuine learning gains that persist even when technological support is removed. Addressing this question requires not only the development of AI based tools but also the design of rigorous methodologies capable of evaluating their educational impact.

The present thesis addresses these issues through the development and evaluation framework of a domain specific chatbot designed for the university course *Analysis and Management of Production Systems*, taught at Politecnico di Torino. The project aims to explore how recent advances in generative artificial intelligence can be leveraged to create a conversational assistant capable of answering questions related to the course content and supporting students during their study activities. The chatbot is built using Microsoft Copilot Studio and is connected to a knowledge base containing the official teaching materials of the course, including lecture slides and additional explanatory documents.

The thesis is structured around two complementary objectives. The first objective concerns the technological and methodological aspects of chatbot development. This includes the analysis of existing conversational AI platforms, the evaluation of criteria for selecting appropriate language models, and the design of a knowledge base capable of grounding the chatbot's responses in reliable course specific information. Particular attention is devoted to understanding how different development environments, ranging from custom built solutions to low code platforms, influence the flexibility, scalability, and maintainability of chatbot systems.

The second objective of the thesis focuses on the educational dimension of the project. Beyond the technical implementation of the chatbot, it is necessary to investigate whether such a tool can provide a tangible benefit for students. To address this question, the thesis proposes a structured experimental design aimed at evaluating the potential impact of the chatbot on students' learning outcomes. Although the experiment itself is not implemented within the timeframe of this research, the thesis develops a detailed methodological framework that could be used in future studies to assess the educational effectiveness of conversational AI tools.

In order to build a solid foundation for this analysis, the thesis begins with an overview of the main technological platforms used for chatbot development, including Microsoft Copilot Studio, IBM Watson Assistant, and Google Dialogflow. These platforms illustrate the diversity

of approaches currently available for the creation of conversational agents, ranging from graphical no code environments to more complex architectures requiring programming expertise. The discussion then moves to the criteria for selecting language models, examining both technical and non technical factors such as performance, economic sustainability, privacy considerations, and data governance. Following this technological analysis, the thesis presents a systematic literature review on the use of chatbots in university education. The review examines academic research focusing on the design and implementation of conversational agents within higher education settings, with particular attention to engineering courses and domain specific learning environments. By analyzing the technological architectures adopted in previous studies and the educational objectives pursued by researchers, the review provides a broader understanding of the current state of the art in this field.

Building on the insights derived from the literature, the thesis then describes the development process of the chatbot created for the AMPS course, including the preparation of the knowledge base and the configuration of the conversational agent within Copilot Studio. Finally, the work proposes a design of experiment aimed at evaluating the impact of the chatbot on students' learning outcomes through structured group activities and quantitative and qualitative assessment methods.

Through this combination of technological analysis, literature review, system development, and experimental design, the thesis seeks to contribute to the ongoing discussion on the role of artificial intelligence in higher education. In particular, it aims to demonstrate how generative AI technologies can be integrated into academic contexts in a way that supports teaching activities while maintaining methodological rigor in the evaluation of their educational effectiveness.

2 Generative AI Technologies and Educational Applications

The rapid development of artificial intelligence technologies, and generative artificial intelligence in particular, has profoundly transformed the ways in which computational systems interact with users and produce complex content. In the educational context, these developments open up new possibilities for learning support, introducing tools capable of assisting students and instructors in a continuous, personalized, and scalable manner. However, in order to fully understand the potential and the limitations of such applications, it is necessary to situate the underlying technologies within a structured theoretical and technical framework. This chapter aims to provide such a framework by first examining the historical evolution of artificial intelligence, from its earliest theoretical formulations to the emergence of modern deep learning models. The objective is to illustrate how the transition from symbolic and rule-based approaches to data-driven models has progressively expanded the capabilities of intelligent systems, ultimately enabling the development of increasingly sophisticated conversational applications. The chapter then explores the operating principles of Generative AI, with particular attention to Large Language Models and the Transformer architecture, which today constitute the technological foundation of next-generation chatbots. The training and alignment phases of language models are also analyzed, highlighting the central role of data, fine-tuning, and human feedback in shaping the system's final behavior. Finally, the focus shifts from model-level technology to application-level architectures, describing the fundamental components that make up a complete AI chatbot. From this perspective, the chapter establishes the conceptual foundations necessary to understand how a language model can be integrated into a conversational system designed for university-level education, thereby setting the stage for the analysis of development platforms and for the case study addressed in the subsequent chapters.

2,1 Historical Evolution of AI

In 1950, the famous British mathematician Alan Turing published an article in the journal Mind titled Computing Machinery and Intelligence, in which he asked himself if machines could think. Turing proposed the "imitation game" later known as the Turing Test, a thought

experiment in which a machine is considered intelligent if it can interact with a human being without that the person is able to distinguish it from another human. Although many scholars have criticized this test for focusing exclusively on linguistic behavior without considering real cognitive processes, the Turing Test is still regarded as a benchmark for evaluating machine intelligence, and until today any software has passed it (Copeland, 2000).

Artificial intelligence as an academic discipline emerged in 1955, when the American mathematician and computer scientist John McCarthy coined the term. The following year, together with other colleagues, he organized the Dartmouth Summer Research Project on Artificial Intelligence, a conference whose explicit objective was to explore the possibility of simulating cognitive functions within machines (Toosi et al., 2021). The project proposal contained a statement that later became a foundational theoretical reference for artificial intelligence:

"Every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

In the years following Dartmouth, research on artificial intelligence experienced a period of enthusiasm and rapid expansion. Two distinct approaches emerged among researchers. On one hand, symbolic AI, based on the use of symbols to represent objects and their relationships and on behaviors defined by a set of predefined rules; on the other hand, connectionist AI, inspired by the functioning of the human brain and based on artificial neural networks composed of simple interconnected units. These systems are able to learn and adapt their behavior without following explicit defined rules.

Public investment in the United States, increased significantly leading to the creation of first dedicated research laboratories at institutions such as MIT, Stanford University, and Carnegie Mellon University. Toward the end of the 1950s and during the 1960s, the first applications of artificial intelligence began to emerge. In 1957, the American psychologist Frank Rosenblatt introduced the perceptron, one of the earliest artificial neural network models, a machine capable of learning from examples (Toosi et al., 2021). Its functioning was based on an algorithm that given a set of input data assigned a weight to each input; using an activation function, it then computed the sum of the products of each input and a corresponding weight; if the sum exceeded a threshold value, the model produced a binary output. This mechanism allowed the system to learn over time by updating the weights after each classification error, improving its decision-making abilities with time. The major limitation of this system was that

it could provide correct outputs only for linearly separable problems, problems for which a line or a plane separates all data combinations into two classes corresponding to the two possible outputs.

In 1966, the German computer scientist Joseph Weizenbaum wrote the code for ELIZA, the first chatbot in history. ELIZA simulated a conversation by extracting keywords from the user's input and inserting them into predefined sentence templates. Its most famous script simulated a Rogerian therapist, that is, a non-directive therapist whose role is to help the patient clarify and process their feelings, often through reflection or paraphrasing of the patient's statements. To emulate this behavior, ELIZA reflected the user's sentences in the form of questions or invited them to elaborate further on the topic they were discussing.

Between 1966 and 1972, Shakey the Robot was developed at the Artificial Intelligence Center of Stanford University. This project represented the first attempt in history to combine artificial intelligence and robotics in order to enable a robot to move autonomously within an environment containing obstacles. Shakey did not operate following a fixed sequence of actions but it was equipped with cameras, sonar sensors, and bump detectors that allowed it to perceive its surroundings and adapt its movements. Its behavior followed a reasoning process: starting from a goal expressed in natural language, the system broke it down it into a sequence of elementary actions, which were then executed one after another while updating its internal representation of the environment based on detected changes. Shakey showed all the limitations imposed by the technology of the time. the robot was able to operate only in controlled environments, the thinking process was slow and expensive and the system wasn't able to learn or adapt to unexpected situations (Kuipers et al., 2017). Even with these limitations Shakey represents an important step in the evolution of artificial intelligence.

From the second half of the 1960s, however, the enthusiasm generated by the Dartmouth conference began to fade as unmet expectations regarding the advancement of artificial intelligence technologies accumulated. In 1973, the British Parliament commissioned the mathematician Sir James Lighthill to evaluate the state of AI research in the United Kingdom. He produced a document called: "Artificial Intelligence: A General Survey", later known as the "Lighthill Report", in which he stated that research failed to achieve the goals promised by its supporters. After these conclusions the British government to reduced funding to AI research, and the effects of the report extended outside United Kingdom, creating a climate of skepticism for AI. Globally research slowed, funding declined, public interest decreased and many

researches were abandoned. This period became known as the first AI winter (Toosi et al., 2021).

The end of this period happened in the early 1980s, thanks to the growing popularity of a new type of AI program capable of producing concrete results: expert systems (Brock, 2018). An expert system is a program designed to solve problems or answer questions in a specific domain using rules based on if–then logic, derived from the knowledge of human experts in that field. They focused on a narrow domain and were relatively easy to develop because the rules dictating their choices were represented explicitly. This explicit representation also made it possible for users without programming skills to configure and operate these systems. The most famous expert systems were DENDRAL, developed by Edward Feigenbaum at Stanford University, which was capable of identifying unknown organic molecules by analyzing data from a mass spectrometer, and MYCIN, used to identify bacteria responsible for infections and to recommend antibiotic treatments adapted to the patient's weight (Lederberg, 1990). In this period the first cases of expert systems in corporate contexts also started to appear. The R1 system, developed at Carnegie Mellon University for Digital Equipment Corporation, was capable of automatically selecting components in the process of computer configuration based on customer specifications. R1 automated this process with an accuracy of 98%, significantly higher than the rate of human management; it generated annual savings close to 40 million dollars for the company.

Despite these successes, which attracted considerable attention and funding to the symbolic approach, some scientists became convinced that this type of AI would never be able to solve problems related to perception or robotics, where learning and the ability to make assumptions are essential. As a result, a small group of researchers revisited earlier approaches and developed new ones. During this period, artificial neural networks experienced a new popularity, and in 1986 the backpropagation algorithm, originally discovered in the 1960s, was successfully applied for the first time. This algorithm enabled the training of multilayer neural networks: networks composed of one or more hidden layers between input and output, allowing them to make more accurate predictions and to solve problems that are not linearly separable, a major limitation of single layer networks like perceptrons (Rumelhart et al., 1986).

The enthusiasm of the 1980s proved to be a bubble. Following the failure of numerous companies, the belief spread inside the business world that AI technologies were not yet ready to have a significant impact on real-world applications. An emblematic case was the Japanese Fifth Generation Computer Systems project, launched in 1982 by the Ministry of International

Trade and Industry (MITI) with the aim of developing a new generation of intelligent computers based on logical reasoning and parallel processing, capable of supporting natural language dialogue, machine translation, and deductive reasoning. The project turned out to be a commercial failure: the machines developed did not achieve the promised levels of intelligence, due to the inefficiency of logical programming languages and the lack of learning capabilities. Applications in natural language dialogue, reasoning, and translation never reached satisfactory performance levels. The wave of pessimism triggered by this failure, led to a second AI winter, a new period of stagnation characterized by a drastic reduction in funding, academic output, and industrial applications (Toosi et al., 2021).

Throughout the remainder of the 1990s and the early 2000s, artificial intelligence began to provide increasingly important solutions for the technology industry and became a fundamental component of numerous software systems and web applications. Nevertheless, once these technologies became common, they started to be perceived just as standard computing tools, with little emphasis on their belonging to the field of artificial intelligence. As a result, even if AI became essential to the functioning of many systems, its presence was largely overlooked by end users. In May 1997, Deep Blue, a supercomputer developed by IBM, defeated the Russian world chess champion Garry Kasparov for the first time (Seirawan et al., 1997). This event had strong symbolic significance and captured the attention of the media and global public opinion, demonstrating that an artificial intelligence system could outperform the world champion in a discipline traditionally associated with human intellect.

The early years of the twenty-first century marked a technological turning point. During this period, access to vast amounts of data and the increased computational power of computers made it possible to successfully implement machine learning algorithms for the first time. After 2010, artificial intelligence entered a phase of exponential growth, driven primarily by the rise of deep learning, a subset of machine learning based on the use of artificial neural networks composed of many layers of interconnected nodes, known as artificial neurons. The year 2017 marks the beginning of the so-called AI boom. Researchers at Google published the paper Attention Is All You Need, which introduced the Transformer architecture, a completely new structure that abandoned the traditional sequential mechanisms previously used. Instead of processing text word by word in a sequential order, the Transformer analyzes all words simultaneously and computes how much each word should influence the others, improving output quality and reducing training time (Vaswani et al., 2017). This innovation represents the technological foundation that, starting the following year, enabled the emergence of Large

Language Models (LLMs): artificial intelligence models designed to understand and generate natural language, trained on enormous quantities of text from which they learn linguistic structures and semantic relationships between words and context.

2,2Principles of GenAI Operation

The term Generative AI (GenAI) refers to a branch of machine learning whose objective is the creation of original content in various formats, such as text, images, audio, and others (Goodfellow et al., 2014). The functioning of this family of AI systems is based on learning patterns and statistical structures from vast amounts of pre-existing data, enabling them to generate new content in response to a specific input prompt (Zhao et al., 2023).

Within the broader set of generative artificial intelligence systems, a particular group of models stands out as the most widespread and widely adopted on a global scale: Large Language Models (LLMs). As previously discussed, the logical foundation of these systems is the Transformer, an architecture based on the attention mechanism in which text is converted into numerical representations called tokens (Vaswani et al., 2017). A token does not correspond to a single letter but rather to a group of characters, which may belong to different words; in general, it is estimated that 1,000 tokens correspond to approximately 750 words [1]. Each token is converted into a numerical vector called an embedding, which captures semantic and syntactic information. A positional encoding is then added to each embedding to allow the model to understand the order of the sequence, since Transformers process text in parallel rather than word by word (Vaswani et al., 2017).

At this point, the core of the architecture starts operating: the attention mechanism calculates relevance weights that determine how much each token influence the others. The model get the vector sequence and, through the multiple layers of the Transformer it keeps evaluating which parts of the context are most relevant for understanding each element. In this way, the model can learn conceptual and grammatical relationships between words that are apart in a sentence, gaining the ability to produce responses that are coherent with the context. Text generation is based on sequence of successive predictions, where each step is influenced by the representation of the topic constructed by the Transformer (Brown et al., 2020).

Before these models can function correctly, they must be a trained with a process that transforms them from simple numerical structures into systems able to produce coherent text. During this process, the model is shown large quantities of text and analyzes them; absorbing the mechanisms of language. It’s able to learn all the aspects of human writing like: words that

appear together more frequently, how gramatical rules work, and the context influence of context in the meaning of a sentence. The training of LLMs consists of three main phases. The first, called pre-training, involves exposure to massive amounts of data from diverse sources, such as books, conversations, and articles. This is anon specialized training stage: the model is not instructed on specific tasks but just analyzes data to develop writing competence. The central mechanism of pre-training is next token prediction. The model receives a sentences with missing characters and it's asked to predict the missing element or the token that should follow. At each iteration, the model produces a prediction, which is compared with the actual token in the original data; the resulting error is used to update its parameters. It is important to emphasize that this procedure does not give to the model a deep understanding of language; instead it enables it to learn the statistical distribution of language patterns.

Pre training phase is followed by the fine tuning. In this phase, the general linguistic capabilities acquired earlier are adapted to the specific task for which the LLM is intended. The learning process is still based on token prediction, but the model is now exposed to a much smaller and more specialized dataset, containing text consistent with the function the LLM is designed to perform. A further difference is that, at this stage, data are often supervised, meaning they include examples of input and output that explicitly demonstrate the expected behavior of the model (Touvron et al., 2023). If pre-training teaches the model how to generate text, fine-tuning trains it to produce responses with a style consistent with the target context and oriented toward a specific objective.

In the last phase, the model must still learn which responses are useful, correct, and appropriate, in order to produce safe and reliable dialogues. This process is known as alignment and works differently from the other stages. The most used method is Reinforcement Learning from Human Feedback (RLHF), which uses judgments and rankings given by real people as reward to optimize model responses. First reward model is created: a group of human annotators is asked to choose between two or more output generated by the model and to indicate which one is more appropriate. These choices are associated with numerical scores, forming an incentive system in which responses judged as better receive higher rewards. The preferences do not directly improve the language model but they are used to build a reward function that represents what persons consider a correct behavior. In the next phase, the model generates responses that are evaluated by the reward model, it assigns them a score that represent the vote that a human judge would give (Christiano et al., 2017). Higher scores correspond to more desirable responses, and the model uses this feedback to adjust its parameters using reinforcement

learning. The result is a transformation in the model's behavior: it keeps the predictive structure learned during pre-training but it adapts it to the usefulness of its outputs. After this process the model's logic does not depends only on statistics but also on preferences expressed by real human beings.

2,3 Core Components of an AI Chatbot

Having analyzed how generative language models learn, represent language, and produce text, it is now necessary to shift the focus from the model itself to the way it is integrated in a complete conversational system. An LLM Is only the cognitive core of a chatbot, but it is not sufficient alone to guarantee an effective interaction with the user. In order to obtain a tool that can be effectively used in practice, the model must be integrated within a broader architecture composed of multiple components capable of managing information sources and enabling the user experience.

The tangible component of the chatbot, the one with which the user interacts directly, is called the front end, and it serves as the point of contact between the system and the human experience. It consists of a set of visual and elements that make conversation possible and aim to make it as similar as possible to interaction with another human being. Its structure typically revolves around a chat window that allows the user to ask questions and receive answers immediately. In addition to input and output fields, the front end may include further components that enhance usability, such as message send buttons, indicators showing when the model is generating a response, side panels that display related materials, and the history of conversations. In some cases, functionalities may also be provided to upload documents or integrate multimedia elements, increasing the chatbot's informational capacity. Although, from a technical standpoint, it may appear to be the simplest component, the front end has a substantial impact on the chatbot's accessibility and on the quality perceived by users (Haugeland et al., 2022).

The message submitted by the user in the front end is passed to the orchestration level, which is responsible for preparing it for processing. This preparation includes basic operations such as text cleaning, character normalization, and verification that the input format is correct. In some cases, it may also involve preliminary checks, such as detecting if the message contains a prohibited request: the so called guardrails, or if clarification is required before forwarding it to the language model. Once this procedure is completed, the orchestration level determines the logical flow of the request, deciding if it should be sent directly to the model, trigger an

information retrieval module, access a database, or request additional input. When passing information to the model, this component is also responsible for constructing the actual prompt: the model does not receive just the user's sentence, but a package of instructions, context, retrieved documents, and rules of behavior. At this point, the model starts generating the response, and throughout this process the orchestration level performs a control role, verifying if the content complies with specific guidelines, whether it contains incorrect information, or whether it needs to be reformulated before being delivered to the user. This phase may include semantic filters, safety checks, or consistency validations that reduce the risk of errors and increase the reliability of the chatbot. The orchestration level's final intervention happens immediately before the response delivery stage, where it adapts the output format and may enrich it, for example by adding links, references, and suggestions (Xi et al., 2025).

Although language models possess a large amount of implicit knowledge learned during training, they do not have direct access to external sources, and they can't guarantee that generated information is always correct, updated, or consistent with a specific file sources. For this reason, chatbots require a dedicated system to identify the most relevant information in the materials available. Its operation follows a sequence of steps. The reference documents, which may exist in multiple formats such as PDFs, Word files, HTML pages, or SQL databases, are analyzed and divided into smaller units, called chunks. By using an embedding algorithm, the text contained in the chunks is converted into digits, creating a vector composed of a list of numbers that identifies a point in a multidimensional space. The key characteristic of this space is that the distance between two vectors reflects the similarity between the topic of the two texts. Texts with similar meanings are mapped to nearby points, semantically distant texts are located far apart (Lewis et al., 2020). This approach enables searches based on the meaning of texts rather than on word matching; in this way, the system can recognize that two sentences express a similar concept even when they use different words. The ability to recognize information based on meaning instead of characters makes it possible to identify relevant content also if the user chooses different terms compared to the original text.

Once the retrieval module that enables the chatbot to access external information and write responses based on verified content, has been described, another requirement is needed for guarantee a correct interaction: the ability to maintain coherence over time and continuity during the dialogue. LLMs are stateless systems: each request sent to the system is treated like an isolated transaction, independent of previous request. The server does not store session data or state variables between calls. To solve this problem, the model is complemented with a

conversational memory. This component is responsible for managing the state of the conversation and providing the language model with the information needed to maintain coherence in dialogue turns. This component sits between the backend and the model and is able to store and retrieve relevant information that is used during the exchange between the user and the chatbot. In simpler systems, this takes the form of session memory that keeps track of key elements like: the last topic discussed, parameters provided by the user or variables necessary to complete a logical flow. Simpler systems uses information contained only in the recent dialog; these are relatively cheap and easy to manage, but they have clear limitations: they do not allow the system to recover information from distant parts of the dialogue, these systems can't analyze the importance of the information. More complex systems adopt a selective approach based on identifying the most relevant portions of the chat. After selecting information useful for the conversation, the memory system performs a vector transformation and stores information in vectors that support retrieval based on contextual, using the mechanism similar to embedding retrieval.

In many cases, interaction does not end with the production of a textual response, but instead requires the system to perform concrete operations or to interact with external resources. In order to enable this feature, it's necessary to introduce a layer dedicated to action execution and integration with external tools, usually referred to as the action layer. In practice, this component allows the chatbot to go beyond text generation by acting as a bridge between natural language and the execution of concrete actions, transforming a user's request into an operation (Xi et al., 2025). Through this layer, a chatbot can perform a variety of actions: real time web searches, creating files in different formats, accessing external databases, sending e-mails, consulting or updating an online calendar, using tools that perform simulations or mathematical computations that the model would not be able to execute internally, and also accessing basic functions such as retrieving the current date and time; data that are not available to the model itself.

Practically all these actions are based on API calls (Application Programming Interfaces), formalized sets of protocols, procedures, and tools that define how distinct software components interact. Form thr operational point of view, an API call consists of sending a message containing the required parameters, formatted according to a standard such as JSON or XML, to an endpoint: an access point exposed by an external service. The client platform receives the request to this endpoint and after interpreting it, returns an output. This process is done automatically, without human intervention, and makes possible for different systems to

communicate even if they have been developed using different technologies (Schick et al., 2023).

3 Platforms and Practices in Chatbot Development

After analyzing in the previous chapter the historical evolution of artificial intelligence and the operating principles of Generative AI models, this chapter shifts the focus from the theoretical level to the applied one, concentrating on the tools and practices used for the concrete construction of chatbots based on language models. The aim is to understand how the technologies described can be translated into operational systems that are effectively usable, with particular reference to academic and educational contexts. The chapter examines the main platforms currently available for chatbot development, highlighting the differences between open-source solutions, orchestration frameworks, and commercial low-code or no-code platforms. Alongside the description of these platforms, the chapter introduces the criteria for selecting language models, emphasizing that the choice of the most appropriate technology cannot be based solely on model performance, but must also take into account economic considerations, data governance, privacy, and long-term sustainability. Finally, the chapter concludes with a literature review on university chatbots, which situates the technological choices within the state of the art of academic research. This analysis provides a comparative overview of the solutions adopted across different studies, highlighting recurring trends.

3,1 Main Platforms for Chatbot Development

This section presents the main online platforms currently used to build chatbots, describing them as development environments that reflect different design choices and usage approaches. These platforms are among the most widely used today in both academic and professional settings, as they offer ready-made tools that make it easier to create conversational systems. The focus is not only on the features they provide, but also on how they are typically used, the level of technical skills required from developers, and the degree of dependence they create on the technology provider. The goal is to offer a clear and comparative overview that helps to understand how different platforms influence development practices, experimentation possibilities, and the long-term sustainability of a chatbot. This overview provides the necessary background to critically interpret the technological choices made in the AMPS course case study.

3,1,1 Hugging Face

Hugging Face Inc. is an American company with headquarter in New York, founded in 2016 by the French entrepreneurs Clément Delangue, Julien Chaumond, and Thomas Wolf. The name

originates from the initial concept of the company's first product: a mobile application featuring a friendly chatbot, described as an "AI best friend forever (BFF)" designed for adolescents. The chatbot provided companionship, support, and entertainment, using a communication style rich in emoji, from which the name Hugging Face was derived. Over time, the company made the chatbot model opensourced and progressively shifted its objective toward building a collaborative machine learning platform aiming at making AI accessible and open. Hugging Face began collecting and distributing natural language models and experiencing a rapid growth of this ecosystem. In 2022, the company released BLOOM, a large language model with 176 billion parameters made publicly available under an open science approach, representing the first open-source LLM of that scale [2].

The company's main product is the Hugging Face Hub, the flagship platform of the open-source AI ecosystem. It is a collaborative online platform that serves as a centralized repository of machine learning models, datasets, and AI applications shared by the community. Today it hosts more than two million pre-trained models and approximately half a million datasets. The Hub works like a social network for AI where users can publish tools, use models created by others, and collaborate with researchers and enthusiasts. Offering pre trained models, the hub is able to guarantee easy and cheap access to LLMs by saving users the computational and financial costs required to train a model from scratch[3]. This central role played by the community has significantly simplified the sharing of scientific results in the research, reducing the distance between academic laboratories and commercial applications.

Another initiative implemented by the company to increase access to AI, is the introduction of Transformers Python library, a tool that allows all models available on the Hub to be integrated into custom Python code. This gives programs the possibility to access models via APIs and facilitates the creation of interfaces known as pipelines. These pipelines automate the steps required to use a model without additional coding: they handle input processing into a format understandable by the model and perform post-processing to render the output human-readable. Pipelines therefore allow users to deploy a chosen model for a specific task with a single command and obtain usable output in just a few lines of code.

What makes Hugging Face unique is its open source approach, which allows users to choose form a database of thousands of pre trained models. The importance and the priority given to transparency from Hugging Face makes the platform ideal for academic and educational contexts. However, being a technical tool, it requires strong ability in Python programming and

does not provide a graphical interface or intuitive user experience, every component of a chatbot must be manually defined and managed inside the code.

3,1,2 LangChain

LangChain is an orchestration framework designed to facilitate the integration of large language models in the development of complex AI applications using a modular approach. It was launched as an open source project in October 2022, the company was founded the following year by Harrison Chase,CEO, and Ankush Gola. The project began in a fundamental moment for genAI: a few weeks later, ChatGPT was released, this event that immediately created widespread interest among AI enthusiasts. LangChain rapidly established itself as a reference toolkit for create AI applications. After an initial rapid growth, the project was formalized into a startup: LangChain Inc. in 2023. The company collected significant investments from venture capital and reached a market valuation of hundreds of millions of dollars [4].

The core strength of LangChain is its modular architecture which allows developers to construct chains composed of multiple components, each performing a specific role in the workflow. Unlike more vertically integrated solutions, LangChain is not a ready to use platform but rather a toolkit of building blocks that integrated into Python code, allow developers to define instructions and operational rules for an agent. In practice, each component of the agent is defined through dedicated classes, allowing users to freely select any existing language model and combine it with retrievers, tools, and other components developed by external providers. The primary advantage of LangChain is the flexibility derived from its modular structure: once a chain encoding the chatbot's behavior has been built, individual blocks can be modified independently without affecting the rest of the architecture. For instance, the underlying language model of a chatbot can be replaced in a matter of seconds while keeping the remainder of the structure unchanged.

In 2023, the company launched LangSmith, a platform that enables developers to monitor and test applications built with language models. LangSmith allows tracking every interaction between the user, the model, and any external tools. The platform summarize all the action and choice take by the model and also the document these are based on, the result are shown via an intuitive interface making it possible to reconstruct the process that led the model to produce a specific response [5].

LangChain give the possibility to develop a chatbot using framework specifically designed manage LLMs with a modular approach, and supporting the development with advanced

functionalities. To use this platform strong programming abilities and a deep understanding of GenAI functioning are require. If a user is able to overcome this technical complexity, the platform gives the possibility of building a chatbot that does not generate only text, but is can reason over content, querying documents, maintaining conversational memory, and evolve into an intelligent assistant.

3,1,3 Rasa

The Rasa project was founded in 2016 by Alex Weidauer and Alan Nichol with the objective of creating an open source infrastructure for conversational AI. Initially, the company was called Lastmile, and in 2016 it released Rasa NLU, an open source library for natural language understanding. The aim was to provide developers with greater control and transparency over models compared to the systems available at the time. In 2017, Rasa expanded the project by introducing Rasa Core, a dialogue management engine based on machine learning; this component was also released as open source (Bocklisch et al., 2017). Over the following two years, Rasa experienced rapid growth and progressively integrated these two systems into a single framework called Rasa Open Source 1.0, while also announcing the launch of a new product: Rasa X. This tool provides a graphical interface designed to visualize and filter conversations, easily transform them into training data, and share the bot with testers. In recent years, Rasa has repositioned its product offering, and today the framework is presented as a platform for generative conversational AI in enterprise environments. This evolution is supported by an infrastructure called CALM (Conversational AI with Language Models), which combines the flexibility of language models with deterministic logic and proprietary knowledge data [6].

Rasa is designed to be hosted on servers owned by the user organization; it can be installed on-premise or on private cloud infrastructures, making sure that conversation data remain under the direct control of the organization. This feature is particularly crucial for sectors with strict privacy and data governance requirements. Due to its flexible and self-hosted nature, Rasa requires a greater initial effort compared to turnkey solutions. Setting up the environment requires technical expertise, and developing an agent involves writing code and manually defining training data. Unlike platforms that provide pre trained models, using Rasa requires constructing conversational examples and datasets of questions and answers used to train the models. Training the models is labor intensive and expensive; also the model tuning that follows requires specific machine learning skills.

With the stated objective of simplifying and accelerating project development, in September 2025 Rasa introduced Rasa Copilot, an AI-based assistant integrated into the Rasa ecosystem. Through the Hello Rasa development interface, this assistant provides useful suggestions, code snippets, and debugging support during the creation of an assistant. At the moment Rasa Copilot is available to the public in a beta stage; it represents a significant innovation for chatbot development and it's a tool that promises to allow users without technical knowledge to build AI agents [7].

3,1,4 Microsoft Copilot Studio

Copilot Studio is the evolution of Power Virtual Agents (PVA), the low code chatbot platform launched by Microsoft in December 2019. PVA allowed users without technical background to create chatbots based on logical units called topics. User intent was analyzed through a natural language understanding (NLU) system, and specific trigger phrases were used to recognize the context of a request and route the conversation in predefined paths.

Over time the original platform saw increasing integration of AI capabilities and the addition of features going from multilingual support to the possibility of performing API calls. The launch of Copilot Studio took place during the Microsoft Ignite conference on November 15, 2023. The transition to Copilot Studio represents a clear change from the original approach, marking the shift from a rule-based paradigm, typical of early enterprise chatbots, to a generative paradigm.

Copilot Studio is characterized as a low-code platform that makes the design of conversational agents accessible to users without programming expertise. From the platform's development environment, users can create a new agent by defining its identity, purpose and the application context in which the Copilot will operate and the type of users it will address (Dyson, 2025).

A crucial step involves connecting the agent to knowledge bases and corporate data sources, enabling it to provide not only linguistically coherent but also information-grounded responses by drawing on documents, databases, or external applications. Through the Knowledge Base section in the creation menu, it is possible to specify links to databases from which documentation can be retrieved, decide whether the chatbot may use web search, or restrict it to specific sources. To personalize the user experience even more, developers decide which functions add or exclude from the chatbot, for example: interpreting code, analyzing documents in different formats, or generating images. Once configuration is complete, the agent can be tested directly in the development environment using a preview chat that simulates real

interactions, allowing verification of response accuracy and progressive refinement of the model's behavior. After testing, the agent can be shared directly to others or implemented in Microsoft Teams; with Copilot studio is possible to choose who has permission to use the agent to maintain control over security and monitoring. The platform also provides the option to start from pre built example templates representing structured and ready-to-use use cases (Dyson, 2025). By selecting one of these templates before beginning configuration, all underlying fields are automatically populated, giving developers the opportunity to modify only selected components to adapt the model to the specific needs of the project. This represents a further step toward making chatbot creation even more accessible to non-expert users, as it helps overcome one of the main initial obstacles in configuration: the manual completion of fields and definition of basic settings.

3,1,5 IBM Watson Assistant

The origins of this line of projects, which takes its name from the founder and first CEO of IBM, date back to 2011, when IBM Watson succeeded in winning an episode of the television quiz show Jeopardy!. In January 2014, the company announced the creation of a dedicated business unit called the IBM Watson Group, with the objective of commercializing the technologies developed through this first NLP system. In 2013, IBM Watson technology became available as a development platform based on cloud ; in 2016 IBM launched Watson Conversation, and in 2018 the service was renamed Watson Assistant [8].

The entire ecosystem has evolved into the Watsonx platform, introduced in 2023. Watsonx represents a portfolio of AI products that includes: watson.ai, which provides an AI-assisted studio for rapidly and more easily creating chatbots; watsonx.governance, designed to manage risks and simplify regulatory compliance; watsonx.data, which enables the management and integration of data from any source and in any format within AI applications; watsonx Orchestrate, which simplifies the creation and management of AI agents; and watsonx Code Assistant, aimed at increasing developer productivity and reducing development time. Using this platform, users can access proprietary models developed by IBM, known as the Granite models. These models are designed to work in defined domains and use controlled content, with a particular attention to reliability and control. They are optimized for extracting information, classify document and they are designed to base all their answers on verified sources reducing the risk of hallucinations in outputs [9].Through partnerships with Hugging Face, Meta, and

Mistral, Watsons also enables the use of non-proprietary models, with the objective of selecting the model that best fits each client's specific use case and operational requirements.

Watsonx is characterized by an approach that allows the platform to be used by individuals with very different levels of expertise. Users that don't have programming skills can develop AI applications using a graphical interface and configuration tools that can be operated using natural language by explicitly stating the objectives, the content and the rules of the system [9]. If a user has intermediate technical skills he can exploit access to models via APIs and use them in complex applications. The platform offers also the possibility to orchestrate the workflow, integrate databases and monitoring performances to advanced users.

3,1,6 Goolgle Dialogflow

The platform was originally founded in 2012 under the name Speaktoit. It received several rounds of venture capital funding and, in 2016, was acquired by Google. By that time, it had been renamed API.AI and provided tools for developing applications for Google Assistant, the company's virtual assistant. This service was discontinued in 2016, and in 2017 API.AI was rebranded as Dialogflow, becoming part of the Google Cloud Platform [10].

Dialogflow is a platform for natural language understanding that simplifies the integration of conversational assistant into mobile applications and websites. Its main field of application is the development of business instruments, in particular for chatbots that provide customer support services. It can analyze multiple types of user input, including text and audio, and can output both in text and with a synthesized voice. The platform offers two main products designed for users with different needs. Dialogflow ES (Edition Standard) represents the original version of the system evolved from API.AI. It is based on a simple model built around three main concepts: intents, entities, and contexts. Intents are the possible objrctive of the user, entities represent the information that the agent is able to extract from the user's input and contexts indicates the ability to maintain continuity during a prolonged conversation. This product is designed for the creation of small agents that can handle only linear conversations with low complexity. Dialogflow CX (Customer Experience) is a more recent product design specifically for enterprise use; it supports the development of complex assistant that can handle long and intricated converastions. In this version, the dialogue is designed using a visually diagram that makes it possible to represent complex transitions characterized by a nonlinear structure with parallel paths, deviations and interruptions [11].

Both products are characterized by the presence of a graphical interface designed to be accessible even to users with limited programming skills. In the case of Dialogflow ES, the interface is intuitive and entirely no code. Intents can be created through a visual module in which example sentences representing potential user inputs are entered, entities to be automatically extracted from text are defined, and the responses that the agent should provide are configured. Contexts, events, and even simple conditional logic can be managed through menus and fillable fields. Developers can test the chatbot directly one the platform using an integrated simulator which displays the bot's response generated and the internal processing steps that created it.

3,2 Criteria for Selecting Language Models

This chapter will analyze several criteria that must be taken into consideration when selecting an artificial intelligence model, in particular when these language models are used in chatbot development. The context of AI models is experiencing a rapid technological evolution and the number of different solutions available increases rapidly. For this reason choosing the most suitable model for the intended application is a strategic decision that affects performance, costs, and therefore the overall sustainability of a project. This choice can't therefore be based only on technical assessment but it requires a complete analysis that takes into account all the model's characteristics because other than the language capabilities models can differ on data management, costs and implementation needs.

Starting from technical capabilities, the selection of an AI model depends on the level of performance it can guarantee. In the case of language models, a first key element concerns the quality of natural language understanding, that is, the ability to correctly interpret questions, it's important to remark that a model must be able to understand a question when this presents some mistakes or is partially incomplete, also a particular attention must be put on the model's ability to maintain coherence. This is complemented by the model's capability in response generation, which must not only be grammatically correct and fluent but, above all, conceptually accurate, avoiding hallucinations and content unsupported by available sources. The level of intelligence demonstrated by a model doesn't depend on easily measurable factors, so, different consolidated evaluation methods have been created to try to make the comparison as objective as possible. The most adopted method is the use of benchmarks, sets of tests and datasets designed to measure how each model performs on identical tasks. The results obtained

on these tests allow direct to have a direct comparison between different models on the same metrics, making it possible to rate them based on their score. On the internet many public leaderboards and comparison platforms collect and update benchmark results, these are also frequently updated to keep up with new versions released. The most used platforms for comparative analysis of language models are LMSYS Chatbot Arena, the Hugging Face Open LLM Leaderboard, and Artificial Analysis.

After examining technical capabilities and the tools used to assess model performance, the analysis can be extended to the economic dimension, which represents another crucial criterion. Cost analysis is an essential factor in choosing an AI model, especially in application scenarios entirely based on cloud services, where access to models occurs through managed platforms and does not require dedicated physical infrastructure. The cost of these services depends almost entirely on the usage modality and the linked payment scheme; provider can offer consumption based price that offer a fixed cost per token or a subscription price that covers computation until a certain threshold. In this framework, a relevant distinction exists between proprietary and open-source models. Proprietary models are generally offered as turnkey services by providers who centrally define pricing policies, terms of use, and service levels, applying variable costs based on token volume or predefined subscription plans that set maximum usage limits over a given period. This solution ensures ease of access and strong integration with other services within the same platform, but it introduces economic dependency on the provider and limited ability to negotiate or optimize costs as usage scenarios evolve. Open-source models don't require the payment of licensing fees and can be accessed on cloud relying on services from third party providers. These services can have subscription or consumption based payment scheme, similar to proprietary models. The possibility to choose the provider and to change it while maintaining the same model gives to the developers a great flexibility and reduces the lock in risks.

An aspect that is often overlooked in model selection but has a central importance is privacy and data management, which are particularly important in academic and institutional contexts. In the case of subscription based models, user inputs and generated responses pass through centrally managed platforms controlled by the provider, which defines data processing, storage, and potential reuse policies. These platforms usually offer high standards of security and must respect privacy laws, but data control remains external to the organization using the model, introducing potential concerns. Open-source models, even when used via cloud services, allow greater autonomy in defining data management policies, as the model can be hosted by

providers selected by the organization or on the organization's own servers. This ensures a higher degree of control and transparency over the data.

3,3 Literature Review on University Chatbots

This literature review was conducted using a systematic approach aimed at identifying the state of the art in the development and use of chatbots in university settings, with particular attention to engineering contexts. The primary objective was to identify the most adopted technological solutions, recurring platforms, and artificial intelligence models used in the design of chatbots for university courses; the results formed the theoretical basis for the research on practical implementation of AI instruments in academic field. The bibliographic search was performed on academic databases, including Google Scholar and Scopus, these two platforms differ in their management policy but both of them can function as a search engine that collects research paper from both accredited institution and independent researchers. Paper selection was conducted through the combined use of keywords referring to three fundamental dimensions. The aspect concerns technology, including terms such as chatbot, conversational agent, virtual tutor, and question answering. The one relates to the application context and includes references to university education and higher education, with particular attention to engineering courses. The third focuses on educational objectives using keywords like learning support, student engagement and tutoring.

In particular, the analyzed results are derived from the use of the following search strings:

- **'Course specific AI chatbot university'**,
- **'Domain specific chatbot engineering education'**,
- **'AI higher education'**,
- **'Implementation of AI chatbot for university courses'**.

In total 52 papers were selected and analyzed, forming the reference corpus of this literature review. Considering the publication year of the paper, is clearly visible how most of the paper have been publicated in the last two years, this trend testifies the growing interest of research in educational AI. Papers published before 2018 are rare, indicating that conversational solutions applied to education were still marginal and often based on relatively older technologies like logic approach. Starting from 2019 it's possible to observe an initial increase in the number of publications, the rate of publication accelerated and became pronounced in 202. In the last three years, the majority of the works analyzed have been published, with a particular rise in 2024 and 2025. This trend is estimated to continue growing in the next year,

this is another proof of the central role that AI is going to play in the academic system of tomorrow.

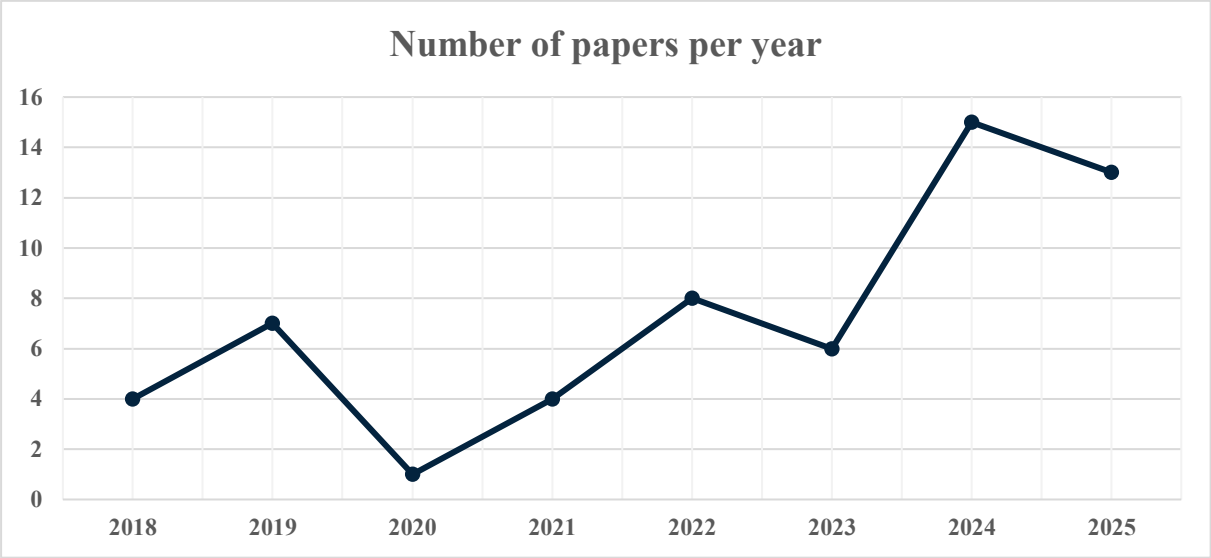


Figure 1: Number of papers published per year (2018–2025).

Of the 57 papers found only a certain percentage is focused directly on the design and implementation of a chatbot in the context of a university course or higher education in general. A group of 50 papers describes the actual development of such instrument with sufficient detail to extract information about components and infrastructure used. The remaining works are relevant for the theoretical framing of the topic but focus on it on a general level without showing a practical approach, most of these remaining paper report comparative analyses between platforms, literature reviews or conceptual discussions, without presenting a concrete implementation.

With regard to the platforms used for the development of the analyzed chatbots, the results show a clear prevalence of custom solutions, designed and implemented by the authors of the individual studies. In most papers, the chatbot is built on architectures developed internally, often combining open-source libraries, machine learning frameworks, and specific software components in order to adapt system behavior to the needs of the course or the university context. This choice is generally motivated by the need for a high level of personalization, especially in cases where the chatbot must interact with complex topics or with educational materials structured in various formats.

Alongside these custom solutions, the literature highlights the presence of established and widely recognized platforms, many of which have been discussed in the previous chapters of this thesis. These platforms are mainly cited and used in studies where the objective is to reduce

development time and complexity by leveraging low code tools for dialogue management, natural language understanding, and integration with knowledge bases. In this paper is highlighted how choosing a commercial tool give researchers the opportunity to focus exclusively on pedagogical aspects minimaxing the effort spent on actual development.

3,3,1 Overview of the sources

Authors	Title	Year	Summary
Benaddi, L., et al.	A Survey on Model-Driven Engineering and Domain-Specific Languages for Chatbot Development...	2025	Survey exploring Model-Driven Engineering (MDE) and Domain-Specific Languages (DSL) to automate and simplify chatbot development, reducing dependency on specific platforms.
Martin, A., et al.	ChEdBot: Designing a Domain-Specific Conversational Agent in a Simulational Learning Environment Using LLMs	2024	A template-based approach to allow teachers without technical skills to create chatbots (CHEdBots) that simulate domain experts or stakeholders for educational interventions.
Fulgencio Sánchez-Vera	Subject-Specialized Chatbot in Higher Education as a Tutor for Autonomous Exam Preparation: Analysis of the Impact on Academic Performance and Students' Perception of Its Usefulness	2025	Evaluation of a specialized AI chatbot as support for exam preparation in an Early Childhood Education degree course, with positive results in clarifying doubts and concept comprehension.
C. A. Fernandez-y-Fernandez, E. Sánchez-Soto, J. R. Aguilar Cisneros, R. Juárez-Ramírez	Exploring the Frontier of Software Engineering Education with Chatbots	2024	Survey of software engineering students and faculty regarding the use, benefits, and challenges of AI chatbots in teaching Object-Oriented Programming (OOP).
Abedi, M., et al.	Beyond Traditional Teaching: The Potential of Large Language Models and Chatbots in Graduate Engineering Education	2023	Examination of LLMs in a fluid mechanics course, evaluating the bot's ability to answer conceptual, analytical, and mathematical questions, highlighting the importance of prompt engineering (e.g., Chain of Thought).
Adwan Mohammad Hasan Bani-Hamad, Mouza Saif Ali Al-Kalbani	Fermi Problem-Based Learning with Artificial Intelligence: Is It Effective to Develop United Arab Emirates	2024	Study on the efficacy of a learning model based on "Fermi Problems" integrated with AI to develop 21st-century skills in science students.

	Cycle Three Students' Twenty-First Century Skills?		
V. G. Al Hakim, N. A. Paiman, M. H. S. Rahman	Genie-on-demand: A custom AI chatbot for enhancing learning performance... in occupational health and safety for engineering education	2024	"Genie-on-Demand": A chatbot trained specifically on PDF materials from an electrical engineering health and safety (OHS) course to provide precise and contextual answers compared to generic ChatGPT.
Yu-Hung Chien, Chun-Kai Yao	Development of an AI Userbot for Engineering Design Education Using an Intent and Flow Combined Framework	2020	Development of "AI Userbots" that simulate real product users to help engineering students practice participatory design and gather requirements via simulated interviews.
L. A. González, A. Neyem, I. Contreras-McKay, D. Molina	Improving learning experiences in software engineering capstone courses using artificial intelligence virtual assistants	2022	Implementation of an educational platform with a virtual assistant ("ZIBot") that recommends "lessons learned" from past projects to current software engineering capstone students.
Ramteja Sajja, Yusuf Sermet, Brian Fodale, Ibrahim Demir	Evaluating AI-Powered Learning Assistants in Engineering Higher Education: Student Engagement, Ethical Challenges, and Policy Implications	2025	Evaluation of the "Educational AI Hub" in civil and environmental engineering courses, offering note generation, chatbots, flashcards, and quizzes.
Caccavale, F., et al.	ChatGMP: A case of AI chatbots in chemical engineering education towards the automation of repetitive tasks	2025	Introduction of ChatGPT for a Good Manufacturing Practices (GMP) course, where the bot simulates an auditor during an audit exercise using RAG documentation.
Michael Pin-Chuan Lin, Daniel H. Chang, Philip H. Winne	A proposed methodology for investigating student-chatbot interaction patterns in giving peer feedback	2024	Development of the "EDI" chatbot to guide university students in providing effective peer feedback (peer review) on essay drafts.
Glendamira Serrano Franco	Development of a Chatbot to Improve Tutoring in Engineering Programs: A Focus at the Universidad Politécnica Metropolitana de Hidalgo	2025	Development of the "Leonora" chatbot to support academic tutoring and answer frequent administrative and academic questions for engineering students.

M. De La Roca, M. M. Chan, A. Garcia-Cabot, E. Garcia-Lopez, H. Amado-Salvatierra	The impact of a chatbot working as an assistant in a course for supporting student learning and engagement	2024	Integration of a chatbot in a Java computer engineering course to assist students in code error correction.
Donn Emmanuel Gonda, Beatrice Chu	Chatbot as a learning resource? Creating conversational bots as a supplement for teaching assistant training course	2019	Case study on the use of a simple rule-based chatbot to support a training course for teaching assistants (GTA) by managing repetitive questions and definitions.
Weijiao Huang, Khe Foon Hew, Donn Emmanuel Gonda	Designing and Evaluating Three Chatbot-Enhanced Activities for a Flipped Graduate Course	2019	Implementation of three chatbot-based activities (Quiz, Case study, Dictionary) in a "flipped" course for graduates to improve engagement with pre-lesson materials.
Mikael, K., et al.	A Hybrid Chatbot Model for Enhancing Administrative Support in Education: Comparative Analysis, Integration, and Optimization	2025	Proposal of a hybrid model for university administrative support that combines rule-based, retrieval, and generative approaches using a Naive Bayes classifier.
Neupane, S., et al.	From Questions to Insightful Answers: Building an Informed Chatbot for University Resources	2024	Presentation of BARKPLUG V.2, a chatbot for Mississippi State University using RAG (Retrieval-Augmented Generation) to answer questions while reducing AI "hallucinations".
Yen-Ting Lin, Jian-Heng Ye	Development of an Educational Chatbot System for Enhancing Students' Biology Learning Performance	2023	Development of a chatbot system for biology learning (7th grade) featuring video functionality, a quiz assistant, and more.
Haya Alshayji, Deja Workman, Swapnika Dulam, Lauren A Griggs, Dixon Zor, Christopher L Dancy	Experiences with using an LLM-based Chatbot for a Multicultural Engineering Program Orientation (Experience)	2025	Use of an LLM-based chatbot during orientation for engineering students to support a design competition and teach AI ethics and algorithmic bias.
André Miguel Pacheco Morais	Catmming: An LLM-based Chatbot to Support Self-Regulated Learning in Software Engineering Education	2025	Transformation of a narrative character (Gatoming) into an LLM-based chatbot to support self-regulated learning (SRL) in an asynchronous online software engineering course.

Shameel Abdulla, Yasser M. Al Hamidi, Marwan Khraisheh	Creating and Implementing a Custom Chatbot in Engineering Education	2023	Development of a chatbot for the Mechanical Engineering curriculum to manage FAQs, plan meetings with professors, and provide course materials.
Erik T. Lopes, Diana Mesquita, Rui M. Lima	Developing a Chatbot for Process Modelling Learning: contributions for active learning in engineering education	2024	"Lateco": A chatbot designed to support Business Process Modeling (BPMN) learning by simulating a process mapping interview with a fictitious employee.
Prarthana P. Kartholy, Neil N. Panchal, Thandi M. Labor, Sean H. Wang, Hillary N. Owusu	BIAS-AWARE AI CHATBOT FOR ENGINEERING ADVISING AT THE UNIVERSITY OF MARYLAND A. JAMES CLARK SCHOOL OF ENGINEERING	2025	Development of an AI chatbot aware of bias for academic counseling in engineering, designed to mitigate algorithmic bias.
Dwi Fitria Al Husaeni, Nuria Haristiani, W. Wahyudin, R. Rasim	Chatbot Artificial Intelligence as Educational Tools in Science and Engineering Education: A Literature Review and Bibliometric Mapping Analysis...	2024	Bibliometric analysis and literature review on the use of AI chatbots in education (2017-2024), highlighting research trends, advantages, and disadvantages.
Dimitrios Ramandanis, Stelios Xinogalos	Designing a Chatbot for Contemporary Education: A Systematic Literature Review	2023	Systematic Literature Review (SLR) on chatbot developments in the educational sector over the last 5 years.
Raymond L. Smith III, Emily Fuller Sondergard, Ricky T Castles	The Engineering Ethics Coach: Integrating AI Chatbots to Strengthen Ethical Decision-Making in Engineering Economy	2025	Introduction of "Engineering Ethics Coach," a personalized AI chatbot to support ethical decision-making based on the NSPE Code of Ethics.
Stephen Crown, Arturo Fuentes, Robert Jones, Rajiv Nambiar, Deborah Crown	Interactive Chatbot to Motivate and Engineering Students	2010	Development of "Anna G. Neering," an AI tool where students help train the system by providing answers to thermodynamics questions.
Helgi Thor Ingason, Kirsi Aaltonen, et al.	Personalised learning in project management education: Insights from an artificial intelligence-driven chatbot	2025	Iterative development of "PMTutor" via Action Design Research to support personalized learning and project management.

Paschoal, L. N., et al.	A Chatterbot Sensitive to Student's Context to help on Software Engineering Education	2018	"Usibot": A context-aware chatbot for teaching software engineering requirements that adapts its responses based on the student's knowledge level.
J. R. Aguilar-Mejía, S. Tejeda, C. V. Ramirez-Lopez, C. L. Garay-Rondero	Design and Use of a Chatbot for Learning Selected Topics of Physics	2022	Implementation of "Professor Atom," a chatbot designed to help first-year engineering students understand Newton's laws through tutorials and examples.
Sebastian Hobert	Fostering skills with chatbot-based digital tutors – training programming skills in a field study	2023	Field evaluation of a digital tutor based on a chatbot to support first-year programmers with individualized learning paths and code analysis.
Hien D. Nguyen, Tuan-Vi Tran, Xuan-Thien Pham, Anh T. Huynh, Vuong T. Pham, Diem Nguyen	Design Intelligent Educational Chatbot for Information Retrieval based on Integrated Knowledge Bases	2022	Proposal of the "Integ-Role" model to integrate multiple knowledge domains into an educational chatbot for introductory and OOP programming courses.
Sebastian Hobert	Say Hello to 'Coding Tutor'! Design and Evaluation of a Chatbot-based Learning System Supporting Students to Learn to Program	2019	Design and evaluation of "Coding Tutor," a system supporting programming students with explanations, step-by-step guidance, and automated code evaluation.
F. Colace, M. De Santo, M. Lombardi, F. Pascale, A. Pietrosanto, S. Lemma	Chatbot for E-Learning: A Case of Study	2018	Prototype chatbot to support university students in computer science courses with immediate responses based on domain ontologies and NLP.
Da-Lun Chen, Kirsi Aaltonen, Hannele Lampela, Jaakko Kujala	The Design and Implementation of an Educational Chatbot with Personalized Adaptive Learning Features for Project Management Training	2024	Implementation of "PMTutor" for project management training with Personalized Adaptive Learning (PAL) functionality.
Ilya Kreynin, MohammedShabbar Manek, Chirag Variawa	Creating a Virtual Chatbot to Scaffold Skills Development in First-Year Engineering Education	2019	Study on an SMS-based chatbot ("Reflective Journal Bot") to help first-year engineering students develop the habit of "reflective journaling".
Jeya Amantha Kumar	Educational chatbots for project-based learning: investigating learning	2021	Inquiry into the impact of integrating chatbots in a project-based design course, showing improved learning performance and teamwork.

	outcomes for a team-based design course		
Bravo, F. A., & Cruz-Bohorquez, J. M.	Engineering Education in the Age of AI: Analysis of the Impact of Chatbots on Learning in Engineering	2024	Qualitative study on the impact of chatbots in five mechanical/electronic engineering courses, analyzing motivation, time savings, and dependency risks.
Joseph Rene Corbeil, Maria Elena Corbeil	Developing AI chatbots to enhance student learning experiences in online courses	2024	Description of Phase 1 of a study on implementing AI chatbots in online university courses to reduce "transactional distance".
Supreetha H V, Sandhya S	Implementation of an Educational Chatbot using Rasa Framework	2022	Development of a chatbot to motivate rural students, track quiz scores, and provide personalized recommendations.
Andrii O. Priadko, Kateryna P. Osadcha, Vladyslav S. Kruhlyk, Volodymyr A. Rakovych	Development of a chatbot for informing students of the schedule	2019	Description of the development of a Telegram chatbot to provide students with information on lecture schedules.
Zia Nadir, Hassan M. Al Lawati, Rayees A. Mohammed, Muna Al Subhi, Abdulnasir Hossen	SQUbot: Enhancing Student Support Through a Personalized Chatbot System	2025	Bilingual (English/Arabic) chatbot for personalized administrative and academic services at Sultan Qaboos University.
Moneerh Aleedy, Eric Atwell, Souham Meshoul	Using AI Chatbots in Education: Recent Advances Challenges and Use Case	2022	Review of chatbot design methods in education and the hubert.ai case study for evaluating student feedback.
Ho Thao Hien, Pham-Nguyen Cuong, Le Nguyen Nam, Ho Le Nhung, Le Dinh Thang	Intelligent Assistants in Higher-Education Environments: The FIT-EBot	2018	Presentation of FIT-EBot for administrative and didactic support at the Faculty of IT, Ho Chi Minh City University of Science.
Alessio Magnolfi, Eugenio Vicario, Leonardo Boncinelli	AI-Powered Chatbot Assistant for Course Organization	2024	Design and evaluation of a chatbot to support organizational information for a university course in Florence.
S. Martinez-Requejo, E. Jimenez García, S. Redondo Duarte, J. Ruiz Lázaro, E. Puertas Sanz, G. Mariscal Vivas	AI-Driven Student Assistance: Chatbots Redefining University Support	2023	Exploration of chatbots for student support and formative assessment in online education.

Thanarat Kingchang, Pinanta Chatwattana, Panita Wannapiroon	Artificial Intelligence Chatbot Platform for Educational Recommendations	2024	Educational recommendation platform based on user attitudes via the "Line" app.
Juliana Ngozi Ndunagu, Christiana Uchenna Ezeanya, Benjamin Osondu Onuorah, Jude Chukwuma Onyeakazi, Elochukwu Ukwandu	A Chatbot Student Support System in Open and Distance Learning Institutions	2025	Development of a chatbot to improve communication and reduce the dropout rate of students in ODL institutions in Nigeria.
Widodo Budiharto, Vincent Andreas, Alexander Agung Santoso Gunawan	AVA: Knowledge-Based Chatbot as Virtual Assistant in University	2022	Proposes a Google Dialogflow-based virtual assistant for university FAQ, achieving 90% accuracy.
Sánchez-Vera Fulgencio	Developing Effective Educational Chatbots with GPT: Insights from a Pilot Study in a University Subject	2024	A case study on developing a GPT-4 based personal tutor for a sociology course using iterative refinement.
Ho Thao Hien, Pham-Nguyen Cuong, Le Nguyen Hoai Nam, et al.	Intelligent Assistants in Higher-Education Environments: The FIT-EBot, a Chatbot for Administrative and Learning Support	2018	FIT-EBot provides administrative and learning support using text classification and entity recognition.
Muhammad Shahroze Ali, Farooque Azam, Aon Safdar, Muhammad Waseem Anwar	Intelligent Agents in Educational Institutions: NEdBOT - NLP-based Chatbot for Administrative Support Using DialogFlow	2022	Proposes NEdBOT for university admission support, evaluated for accuracy and response time.
Akshada Phalle, Saniya Kadam, Sakshi Sonphule, Ila Savant	AI and Web-Based Interactive College Enquiry Chatbot	2021	Identifies requirements for a dynamic college enquiry bot supporting text and voice-based interaction.
K. Shingte, Anuja, Aditee Patil, Anushree Chaudhari, Sharmishta Desai	Research Paper on Chatbot Development for Educational Institute	2021	Developed a college enquiry chatbot using the Chatterbot algorithm to handle admissions and course details.

Naing Naing Khin, Khin Mar Soe	University Chatbot using Artificial Intelligence Markup Language	2019	First university chatbot for school information in Myanmar Language using AIML.
Neelkumar P. Patel, Devangi R. Parikh, Darshan A. Patel, Ronak R. Patel	AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT)	2019	UNIBOT provides university and student info via a dynamic web interface with minimal response time.
Tanya Dinesh, Anala M R, T. Terry Newton, Smitha G R	AI Bot for Academic Schedules using Rasa	2021	Developed an AI bot specifically for academic schedules and syllabus queries using the Rasa framework.
A Kousar Nikhath, Vijaya Saraswathi R, et al.	An Intelligent College Enquiry Bot using NLP and Deep Learning based techniques	2022	Building an intelligent bot for college enquiries using Deep Learning (LSTM) achieving 99% accuracy.

Table 1 Papers Included in the Literature Review

3,4 Summary of Findings

The analysis now turns to drawing conclusions based on the data collected in the literature review, with particular attention to the technological platforms adopted for chatbot development. The results clearly show that, in the vast majority of the studies examined, no-code platforms are not used. This evidence suggests that, within the academic context—and especially in the development of chatbots designed for individual university courses or specific disciplinary domains—solutions that do not require programming skills are either not widespread or are considered insufficiently suitable to meet research and teaching objectives.

The platforms most frequently employed can be grouped into three main categories. First, there is extensive use of fully custom platforms, developed from scratch by the authors of the studies using programming languages and standard web technologies, such as Python for the application logic and HTML and CSS for the user interface. This approach allows for a high degree of design flexibility and enables the chatbot to be precisely tailored to the needs of the course, both in terms of content and interaction modalities with students. Second, dedicated chatbot development frameworks are widely adopted, including Google Dialogflow and Rasa, which provide structured tools for natural language processing while still maintaining a good level of customization and control.

With regard to no-code platforms, their presence in the analyzed literature appears to be decidedly marginal. In particular, only two studies report the use of OpenAI's GPT platform for the creation of artificial intelligence agents. Overall, the findings of the literature review therefore indicate a clear preference for custom or semi-custom solutions, to the detriment of no-code platforms, which are considered less suitable for meeting the requirements of customization, control, and integration that characterize the development of university chatbots.

Platform	**Count of Platform**
Custom	12
Dialogflow	9
Rasa	4
Pandorabots	3
langchain	2
Microsoft Power Virtual Agents	2
Chat GPT	2
Chainlit	2
IBM Watson Assistant	2
Gradio	1
Telegram	1
QnABot	1
Hubert.ai	1
Hugging Face	1
ALICE	1

Chatbase	1
Landbot	1
Flask	1
Chatterbot	1
Textit	1
LINE Messagging	1
Botnoi	1
Moodle	1

Table 2: platform used in papers

It is important to emphasize that not all chatbots analyzed in the studies included in the literature review are designed with the role of a teaching assistant in the strict sense, as is the case with the chatbot developed for the AMPS course. In many contributions, the chatbot instead assumes different and more limited functions, often oriented toward organizational or informational support rather than direct pedagogical guidance on course content. Some studies describe chatbots used as tools for timetable management and operational communication (Dinesh et al., 2021; Priadko et al., 2020), where the system is employed to automatically notify schedule changes, exam dates, and general course information. In other cases, the focus is administrative in nature (Mikael et al., 2025), where the chatbot guides students through the initial phases of enrollment and orientation (Magnolfi et al., n.d.), in which the agent provides information on adopted textbooks and recommended learning materials. Finally, there are contributions that explore the use of chatbots for the development of specific competencies or targeted skills(Kreynin et al., 2019; Smith et al., 2025), where the primary objective is not transversal support for a university course but the enhancement of well-defined skills.

In light of the findings of the literature review, it therefore becomes evident that the possibilities offered by chatbots in the university context are not limited to learning in a narrow sense but extend to a much broader and more articulated set of applications. University chatbots do not necessarily need to be conceived as single-purpose systems; rather, they can evolve into hybrid

solutions capable of combining educational support, organizational management, and administrative assistance within a single conversational interface (Ramandanis & Xinogalos, 2023). Such an approach would allow students to interact with a single point of access for heterogeneous needs, ranging from understanding course content to consulting practical information on schedules, materials, and deadlines, thereby reducing the fragmentation of digital tools and simplifying the user experience. From this perspective, the integration of different roles within a single chatbot represents not only a technical possibility but also a significant design opportunity, capable of increasing the system's perceived usefulness and its overall impact within the complex university environment.

4 The AMPS Course Chatbot: Analysis, Limitations, and testing

Having traced the historical evolution of AI in education, discussed its theoretical foundations, and conducted a structured literature review on university chatbots, this work now shifts its focus to a specific case study. The objective of this chapter is to systematically analyze the chatbot developed for the **Analysis and Management of Production Systems (AMPS)** course. Specifically, it seeks to evaluate how research-driven insights and existing literature can refine a tool that originated as an experimental project, transforming it into a viable solution for real-world educational settings.

The chatbot under review began as an individual thesis project, initially occupying the experimental space typical of many academic prototypes. However, its design is characterized by a rigorous methodological framework that balances technological requirements with pedagogical, organizational, and ethical considerations. In this regard, the AMPS case provides fertile ground for transitioning from a "hobbyist" product developed in a controlled environment toward a scalable platform capable of integration into a structured university ecosystem.

The analysis presented here goes beyond a mere description of the chatbot's functionality; it aims to interpret its architectural and governance choices in light of the best practices identified in the literature. In particular, this study examines how the use of a closed and controlled corpus, the implementation of strict response protocols, and the emphasis on source traceability align with established quality criteria. This approach allows us to determine whether the AMPS chatbot represents a mature application—one that moves beyond basic informational support to serve as a cognitive mediation tool aligned with the course structure.

Finally, this chapter lays the groundwork for critical reflection and continuous improvement. The analysis is not intended as a final verdict, but as an essential step in identifying opportunities for functional and institutional growth. From this perspective, the AMPS case serves as a testing ground for translating theoretical knowledge into concrete redesign interventions, ultimately turning an academic prototype into a reliable, reproducible, and sustainable educational tool.

This subparagraph introduces the main choices that guided the construction of the chatbot for the Analysis and Management of Production Systems course, focusing on the role of the

adopted platform and, above all, on the rules governing the conversational agent's behavior. The chatbot was developed on Copilot Studio, a platform that allows for the creation of conversational agents without requiring traditional programming activities. Access to the agent is provided via the publishing link generated by Copilot Studio. The owner of the account associated with the tool has the possibility, through the share screen, to make the chatbot usable by anyone, grant access to all accounts within their own organization, for example all accounts created with a Politecnico email, or manually select which accounts to provide access to.

4,1,1 Prompt

The assistant's behavior is defined through instructions expressed in natural language, collected in a structured prompt that takes on the role of a true conversational contract. This conversational contract explicitly establishes how the chatbot must interpret student requests and how responses must be constructed, reducing the ambiguity typical of many applications based on generative language models.

The prompt is designed to pursue four fundamental objectives. First, the traceability of responses is guaranteed, requiring that every relevant statement be traceable to the official course slides through a final block of citations. Second, the prompt imposes high educational clarity, requiring responses structured according to a fixed scheme and with a concise register, typically between 250 and 300 words, without introducing content or theories not present in the teaching material. A third objective concerns linguistic and symbolic consistency, as the chatbot is required to faithfully preserve symbols, notation, and terminology used in the slides, avoiding arbitrary translations or renaming that could generate confusion in the study of a quantitative subject. Finally, a strong security mechanism is introduced, which requires the assistant to explicitly refuse any question that falls outside the theoretical scope of the course, returning a standardized response signaling the absence of information in the provided materials, without resorting to external sources or the web; this instruction is known as a guardrail.

Within the prompt, the section dedicated to the role defines the chatbot as a teaching assistant for the AMPS course, with a precise and deliberately limited mandate. The assistant must respond exclusively based on the provided corpus, consisting of slides in PDF format, a Glossary, and Bridge Notes in DOCX format, and must refuse any request for which there is no direct evidence in such materials. This design choice produces three relevant educational consequences. On one hand, it increases the perception of the tool's reliability, as the student knows that the answers are always anchored to the official course materials and not to

potentially inconsistent external sources. On the other hand, it reinforces verifiability, given that every definition, relationship, or formula can be traced back to the slides, allowing the student to easily return to the original source for further study. Finally, it guarantees strong terminological and notational consistency between what is explained by the chatbot and what is presented in the classroom or in official handouts.

A further relevant component of the conversational contract concerns the management of language and style. The chatbot is instructed to respond exclusively in the language used by the user in the current message, regardless of the interface language or previous turns in the conversation. In this way, if the question is asked in Italian, the answer will be in Italian, while a question in English will produce a response in English. This rule eliminates arbitrary fluctuations between different languages and contributes to making the interaction more natural and predictable. At the same time, the prompt dictates not to translate symbols, variable names, or technical terms present in the slides, maintaining even Anglicisms when these are part of the official material, so as to avoid misalignments between the chatbot's text and the cited documents.

The structure of the responses is also rigidly defined. Each output must be organized into an introductory overview section, which delimits the concept and its scope, followed by a part dedicated to key points, in which central information for learning or for solving theoretical exercises is highlighted. This is followed by a section on pitfalls and assumptions, designed to make explicit the underlying hypotheses and the main edge cases, which are often a source of conceptual errors. The response finally closes with the block of citations to the slides. This setup does not have a purely formal function but addresses a precise cognitive need, as it guides the student's attention, reduces cognitive load, and promotes a progressive understanding of the content.

4,1,2 Knowledge base

During the testing and initial experimentation phase, the chatbot was connected to the personal OneDrive folder of the supervising student and author of the thesis. This design choice allows for rapid file updates without the need for coordination with the course administration. However, should the chatbot be intended as a permanent teaching support or for the entire course, the thesis recommends migrating the materials to a SharePoint library dedicated to the course. In this scenario, roles are clearly defined: the teaching team holds read and write

permissions and is responsible for updating and validating the materials, while students have exclusive read-only access to the documents used as the knowledge base.

The files used by the chatbot are uploaded to OneDrive and linked to the agent through the Knowledge Base section available in the chatbot creation or editing menu, where the link pointing directly to the reference folder is inserted. This integration method allows the system to access materials dynamically, ensuring that any updates or modifications to the files are immediately available to the chatbot without requiring further configuration. The set of documents linked in this manner constitutes the knowledge corpus from which the chatbot draws to formulate its responses, representing the exclusive information base on which the agent's behavior is founded. The corpus includes all course slide packages in PDF format, used during lectures and containing the entirety of the theory covered in the course. In total, there are fourteen sets of slides that systematically and completely cover all topics in the course syllabus. Alongside these theoretical materials, there are also two files with a more operational and supportive function: the glossary and the bridge notes.

The glossary is a thematic glossary designed to ensure the coherent and uniform use of course terminology, without introducing additional content or interpretations beyond what is present in the teaching materials. Each entry is organized according to a fixed structure consisting of a definition, aliases, and citations: the definition clearly and faithfully reflects the language used in the slides; the aliases include common variants, abbreviations, and both Italian and English equivalents; while the citations refer directly to the slides where the concept is explained or applied. This structure allows the chatbot to correctly recognize questions even when students use alternative terms or terminology that is not perfectly aligned with the course, to maintain the same language and symbols as the slides in its responses, and to direct information retrieval toward the truly relevant parts of the material. The glossary is therefore a central element for ensuring consistency, clarity, and reliability in responses. It was constructed following criteria of essential completeness, elimination of duplications, and traceability of sources, keeping the document within the M365 repository to ensure continuous alignment with the chatbot.

Bridge notes are short conceptual connection texts designed to make explicit the cross-functional relationships between different course topics that, in the slides, might remain implicit due to their concise nature. Their primary purpose is to support the chatbot in constructing more coherent responses when student questions touch upon multiple modules simultaneously, for example, relating stochastic variability to queue waiting times, or linking push and pull concepts to the connection between WIP, cycle time, and throughput via Little’s Law. Bridge

notes therefore play a limited but strategic role; they do not introduce new content but rather clarify and summarize connections already present in the course materials, preparing the context for an accurate retrieval phase. The texts are intentionally brief and structured into a few paragraphs to guide the organization of the explanation without becoming an independent source of knowledge. The chatbot can use them as support to structure a fluid and logical explanation, while the final citations in the responses always and exclusively refer to the slide PDFs.

4,1,3 Functioning

From an information-retrieval perspective, the system adopts a retrieval plan explicitly designed to maximize coverage and coherence, while keeping strict traceability to the course slides. A clear distinction is made between the sources the chatbot can consult—slides, the Glossary, and the Bridge Notes—and the sources that can actually be cited in the final output, which are limited exclusively to the PDF slides. The goal of the retrieval plan is to identify within the PDFs the definitions, conceptual relationships, and formulas exactly as they are presented in the course, reducing noise and interpretative ambiguity. In this process, the Glossary is used to align the student's terminology with the official course language, while the Bridge Notes help make conceptual links explicit without introducing new or independent content.

The operational flow starts with a preliminary analysis of the query, aimed at normalizing the request by identifying key terms, the type of task required, the conceptual domain, and any explicit constraints, such as references to specific pages or comparisons between approaches. This is followed by a semantic mapping phase, in which the identified terms are linked to Glossary entries, using aliases to handle synonyms, language variants, and abbreviations. This respects the principle that the Glossary does not add new content, but only provides correct labels and pointers to the slides. The primary retrieval phase then focuses on the PDF slides, prioritizing pages that contain definitions, formulas and assumptions, or canonical diagrams, provided that the text and symbols match exactly those used in the final answer.

The Bridge Notes are used later as light support for structuring the reasoning, helping to organize concepts and clarify logical dependencies between topics, without acting as a content source or as justification for claims. Even when they guide the explanation, final citations always refer exclusively to the course PDFs. Source selection follows a principle of minimal sufficiency, choosing the smallest possible number of pages and slide decks needed to cover a

definition, a formula, and, where available, a canonical example, while avoiding redundant or marginal references. Before drafting the answer, the system also checks the citability of the selected sources, verifying that file names exactly match the authorized ones.

In cases of interpretative ambiguity, the system gives priority to the formal definition provided in the slides. If uncertainty remains, it is explicitly stated in a dedicated assumptions section, indicating the slide page used as a reference. For questions that combine multiple topics, information retrieval is carried out by breaking the query into sub-tasks and then recomposing the evidence into a single, coherent, structured answer. The final citation block lists all the slide decks involved on a single line, and the user's use of synonyms or different languages does not affect the anchoring of retrieval to the standardized terminology defined in the Glossary.

This entire setup is based on a clear source-of-truth policy, according to which the PDF slides are the only legitimate source for citations. The Glossary and Bridge Notes support vocabulary alignment and logical organization, but are never cited directly in the output unless explicitly requested by the user. Any information that cannot be verified in the slides is treated as out of scope and leads to a standardized refusal, avoiding any form of improvisation. Before generating the final response, the chatbot performs a series of quality checks, verifying coverage of all claims, symbolic consistency, citation efficiency, compliance with the file whitelist, and respect for page-level policies, in order to reduce recurring errors and maintain a high level of reliability.

4,2 Comparative testing

To assess, first of all, whether the implementation of this chatbot can actually represent an advantage for the students who use it, it is necessary to understand whether the tool is able to answer questions on course topics more effectively than the main general-purpose artificial intelligence solutions. To this end, specific questions related to the Analysis and Management of Production Systems course were submitted to ChatGPT and Google Gemini, selected because they are the commercial AI platforms with the largest number of users. The responses obtained were then compared with those provided by the AMPS chatbot, with the aim of evaluating differences in terms of accuracy, adherence to course content, and ability to correctly use definitions, conceptual relationships, and formalisms as presented in the official teaching material. This comparison makes it possible to isolate the added value of a domain-specific chatbot compared with generalist models, highlighting whether and to what extent specialization in course content improves the quality of responses for learning purposes.

To ensure a fair comparison between the AMPS chatbot and the two commercial chatbots, the latter were provided with the official course slides, so that they could rely on the same information sources. In this way, an attempt was made to reduce the informational advantage linked to access to uncontrolled external knowledge, focusing the comparison exclusively on the ability of the different systems to interpret and re-elaborate the course teaching material. Moreover, before submitting the questions, an initial command was given explicitly requesting the use of the uploaded material to answer the questions and to indicate the slide where the theory used could be found. This setup made it possible to assess not only the conceptual correctness of the responses, but also the chatbots' ability to anchor their statements to precise and verifiable references within the slides, bringing the behavior of generalist models closer to what is required of a course-oriented teaching assistant.

A total of seven questions of different types were submitted to the chatbots, designed to cover heterogeneous interaction types typical of a course like AMPS. In particular, three questions were open theoretical questions and required an articulated explanation of concepts, with the aim of evaluating the chatbot's ability to structure a discursive response consistent with course material. In addition, three quizzes were included, two theoretical and one numerical, structured with four possible closed answers identified by options a, b, c, and d, in order to verify the models' ability to correctly identify the solution among predefined alternatives. Finally, a calculation exercise was included, intended to evaluate not only understanding of the underlying theory but also the correct application of formulas and quantitative relationships presented in the slides.

The responses obtained were evaluated on the basis of three distinct factors, defined to measure in a structured way the quality of the output provided by the different chatbots. The first factor is completeness, which assesses the degree of completeness of the response. A value of zero was assigned to this criterion when the chatbot was unable to provide a meaningful answer to the question; a value of one when the response was correct but extremely essential and lacking adequate explanatory context; and a value of two when the response was exhaustive and fully covered the requested topic.

The second factor considered is coherence, which measures the level of alignment of the response with the language, formulas, and concepts used in the course slides. In this case, a score of zero indicates the use of terms, notation, or conceptual formulations completely different from those adopted in the teaching material. A value of one corresponds to an intermediate situation, in which the chatbot uses a mixed set of formulas and concepts derived

partly from the slides and partly from external knowledge. A value of two, on the other hand, indicates a fully coherent response, using exclusively terminology, relationships, and formalisms as presented in the course.

The third evaluation factor is citation, introduced to verify the chatbots' ability to anchor responses to explicit and verifiable references. A value of zero was assigned to this criterion when the chatbot was unable to cite any source. A value of one was used in the case of partial, generic, or imprecise citations, for example limited to reference to the document without specific indications. A value of two indicates the presence of correct and complete citations, with explicit reference to the slides and pages where the theory used is actually discussed.

#	**Type**	**Question**
1	Theory	Describe the main concepts of the Group Technology and indicate some advantages with respect to the functional layout.
2	Theory	What is the bottleneck rate of a line?
3	Theory	Which are the characteristic of Simulation in the context of Industry 4.0?
4	Quiz	Which is the right Kendall notation to indicate a workstation in which the inter-arrival time follows a normal distribution, the processing time follows an exponential distribution, there are three machines in parallel and the queue is unlimited?
5	Quiz	At which level of the hierarchy are the data concerning the realt time monitoring of the production machines used to take decisions?
6	Quiz	A workstation has a processing rate of 6 j/h. The inter arrival time of jobs to the workstation is 20 min. Which is the utilization of the workstation?
7	Exercise	In a system the mean arrival rate is $\lambda = 6$ jobs/hours, and it is exponentially distributed. The mean service times is $E[Ts] = 0.5$ hours, with a squared coefficients of variation $Cs2 = 0.8$. Which is the number of parallel machines needed to get a utilization lower or equal to 0.9? Considering the obtained number of parallel machines, compute the CT, the WIP and the TH of the system.

Table 3 List of questions used in the chatbot comparative test.

Drawing conclusions from the collected data, the scores obtained by each chatbot in the three evaluation categories: completeness, coherence and citation, were summed. The aggregated analysis shows that the system with the best overall performance is ChatGPT, with a total of 32

points. In second place is the AMPS chatbot, with an overall score of 29, thus showing performance very close to the generalist model but slightly lower in the final result. In last position is Google Gemini, which obtained a total of 21 points, highlighting a more marked gap compared with the other two evaluated systems.

	AMPS			**Chat GPT (5.2 thinking)**			**Gemini (3 reasoning)**		
#	**Completness**	**Coherence**	**Citations**	**Completness**	**Coherence**	**Citations**	**Completness**	**Coherence**	**Citations**
1	0	0	0	1	2	2	2	2	2
2	2	2	1	1	1	2	2	0	0
3	2	2	1	2	2	2	2	1	0
4	2	2	2	1	1	0	0	0	0
5	2	1	2	1	2	2	1	1	0
6	2	1	2	1	1	2	2	2	0
7	0	1	2	2	2	2	2	2	0
Sum	10	9	10	9	11	12	11	8	2
Total	29			32			21		

Table 4 Summary of the evaluation scores assigned to each chatbot.

The AMPS chatbot's overall score was strongly influenced by the outcome of the first question, to which the system provided no answer because the guardrail mechanism classified the query as external to the available theoretical corpus. As a consequence, the response was blocked, with a direct impact on the score assigned in all three evaluation categories. This episode highlights a fundamental design aspect. While, on the one hand, the guardrail is an essential tool to ensure that the chatbot remains aligned with the course domain and avoids hallucinations or content that is not coherent with the official material, on the other hand, excessive rigidity in filtering can compromise the system's overall performance. An overly restrictive mechanism

can in fact classify as out-of-domain even questions that, although phrased slightly differently from the slides, are fully relevant to the course.

A further element to consider concerns the immediacy of use of the different systems. The AMPS chatbot is immediately operational, as it is already configured to work exclusively on course materials and to provide responses consistent with the required structure. By contrast, in the case of ChatGPT and Google Gemini, it was necessary to manually upload the course slides before starting the comparison. This operation would need to be repeated at each new use, together with the insertion of the initial prompt explicitly requesting reliance on the provided material and source citation.

During use of both generalist systems, small adjustment commands were also needed to correct some drift in model behavior, which progressively tended to move away from the initial instructions, for example, by omitting citations or introducing formulations not perfectly aligned with the slides. This aspect was particularly evident in Gemini, which after the first question was no longer able to provide precise citations to the slides, despite repeated explicit requests to comply with that constraint. Precisely this systematic absence of verifiable citations had a decisive impact on Gemini's final score, penalizing it above all in the citation category.

Another element that emerged from the analysis concerns variability in the length, structure, and depth of responses provided by the two commercial chatbots. Across questions, very marked differences were observed, with responses sometimes articulated and at other times extremely concise. Emblematic examples are the second question and the fifth quiz in ChatGPT's case, where the response was particularly brief and essential, lacking contextualization and without any further point for deeper understanding. In general, ChatGPT seems to show a tendency to reduce response length to a minimum when the question structure allows it, favoring synthesis over argumentative development. The AMPS chatbot, by contrast, maintains a more stable and uniform expository structure. The form of the response does not vary significantly depending on whether the question is theoretical, multiple-choice, or related to a numerical exercise. This behavior suggests greater consistency in style and output organization, probably linked to system configuration and to the presence of more binding guidelines in response generation. From a teaching perspective, structural stability can represent an advantage in terms of predictability and clarity for the student, whereas excessive conciseness, even if content-wise correct, risks limiting the explanatory value of the response.

A significant critical issue that emerged from the analysis is that the AMPS chatbot was the only one among the three evaluated systems to provide incorrect results in solving the numerical exercise. In particular, Cycle Time and Work In Process were calculated incorrectly, with a direct impact on the overall correctness of the solution. The full transcription of the answers provided by the three chatbots to the submitted questions is reported at the end of the thesis, in Appendix A. This section systematically compiles the original outputs generated by the systems, allowing for direct verification of the content of the responses and comparison with the evaluations assigned across the different analysis categories.

4,3Replication of the Chatbot System

The final objective of the project is to create a chatbot that can be effectively used by students of the course as a study support tool, integrating it in a stable way among the available teaching resources. For this to happen, however, it is not sufficient to develop a functioning prototype, but it is also necessary to ensure the long term sustainability and manageability of the system.

During the development phase of the project, the chatbot was created using the personal accounts of the students who participated in its design and implementation. This solution made it possible to simplify the initial phases of configuration, testing, and modification of the system, allowing direct access to the tools required to build the knowledge base, define the prompt, and integrate the teaching materials. However, this choice presents several critical issues from the perspective of the chatbot's future management. The accounts used are associated with the university email addresses of the student developers, which are typically deactivated once their academic program is completed. As a result, when these accounts are deleted or disabled, the chatbot could become inaccessible or no longer editable, compromising the continuity of the service.

To avoid this situation and ensure that the system can be effectively used over time, it is therefore necessary to recreate the chatbot using the account of a stable figure within the course, such as the instructor responsible for the teaching or a member of the academic staff. In this way, the management of the platform, access to the knowledge base, and the possibility of updating or improving the system would remain linked to an individual who will continue to be involved in the course in the following years.In order to facilitate this transfer and replication process, a detailed operational guide has been prepared describing step by step all the operations required to recreate the chatbot in a new account while preserving the structure and functionalities developed during the project. The guide includes instructions regarding platform

configuration, knowledge base upload, and the setup of the supporting files. To avoid overloading the main body of the thesis with technical and procedural instructions, this documentation has been included at the end of the thesis as Appendix C.

The guide briefly outlines the steps necessary to recreate the chatbot. First, it explains how to prepare the knowledge base by organizing all the required files in a single folder, including the slides in Markdown format, the video lecture transcripts, the bridge notes, and the glossary. It then describes the actions to be performed in Microsoft Copilot Studio to create the new agent, insert the configuration prompt, and connect the folder containing the course materials. Finally, the guide explains how to complete the creation of the chatbot and verify its functionality through the testing interface of the platform.

5 Technical and Pedagogical Improvement

This chapter describes the technical and pedagogical improvement pathway of the chatbot examined in this thesis, building on the evidence collected in the preceding phases of the work. The analysis of related projects identified through the literature review, together with the observations obtained by empirically testing the chatbot through the comparative evaluation conducted earlier, made it possible to delineate more precisely the system's main strengths and shortcomings. In doing so, it highlighted both the elements that enhance its usefulness as a study support tool and those that constrain its effectiveness in real-world usage scenarios.

On the basis of these findings, the focus therefore shifts from evaluation to the design of targeted interventions. To date, the chatbot has been employed and examined in its initial version, without modifications following its original implementation. Drawing on the insights gained, this chapter introduces the changes that will be implemented to improve the user experience, the quality of the generated responses, and overall reliability, with the aim of aligning the system more closely with the course's instructional needs and increasing its robustness with respect to the limitations observed during testing. The implemented modifications were aimed at strengthening these dimensions through both technical and pedagogical measures. On the one hand, the work addressed the existing materials by improving their organization, quality, and machine-readability; on the other hand, a substantial part of the effort focused on expanding the corpus to provide broader coverage of the course content.

5,1 Replacement and Update of Knowledge Base Files

Before the revision intervention, the chatbot relied on earlier versions of the course files, characterized by content that partially differed from the material currently in use, as well as by a different internal structure and titles that were no longer consistent with the most recent organization of the teaching materials. Over time, the course documents have in fact been updated by the instructor both in structure and in the naming of sections. While these updates preserved the overall coherence of the topics covered, they nonetheless affected the organization of the content and its hierarchical structure. Keeping outdated versions within the knowledge base could have led to misalignment between the material actually used in the course and the content accessed by the chatbot during the retrieval process. For this reason, it became necessary to systematically replace the previous files with their updated versions, thereby

ensuring consistency between the official course documentation, the repository structure, and the responses generated by the system.

Below are the titles of the new slide decks that replaced the previous versions within the OneDrive folder containing the knowledge base files:

- 01_Flow_process_chart_VSM_IDEF0_with_solutions.md
- 02_Introduction_Factory_Models_with_solutions.md
- 03_Single_Workstation_Analysis_with_solutions.md
- 04_Process_variability_with_solutions.md
- 05_Multiple-Stage Factory Models_with_solutions.md
- 06_Multiple_Product_Factory_Models_with_solutions.md
- 07_Batch_models_with_solutions.md
- 08_Information_systems.md
- 09_Industry_4.0.md

5,2Document Parsing and Extraction

A crucial step in the chatbot's technical improvement concerns the preparation of the documents that populate the knowledge base. The course teaching materials are available in PDF format, which is primarily designed to ensure the reproducibility of layout and graphical rendering. In other words, PDF prioritizes the page-based representation of content, ensuring that text, images, and other elements are displayed consistently and independently of the software or operating system used for consultation. This design choice makes PDF particularly suitable for the distribution and long-term archiving of documents, but it entails an important computational implication: the logical structure of the text, understood as the sequence of paragraphs, the hierarchy of sections, and the intended reading order, is not necessarily encoded explicitly or reliably.

When a chatbot or, more precisely, the pipeline that prepares the knowledge base, extracts text from a PDF, it is often reconstructing the reading order ex post by relying on visual and geometric cues. This process becomes fragile as soon as the document is even moderately complex, because textual blocks may be interleaved or mixed. Even when the inferred reading order is acceptable, extraction tends to discard the semantic structure that is crucial for a chatbot: titles, subsections, paragraphs, notes, references, figures, and tables are frequently flattened into a single linear stream. This loss of hierarchy is particularly detrimental because

indexing and chunking rely precisely on local coherence: if a heading becomes detached from its associated content, or if a table is fragmented and intermingled with surrounding text, the resulting chunks become ambiguous and retrieval is more likely to return irrelevant information (Merola & Singh, 2025).

A third critical issue concerns nontextual or only partially textual content. Tables, formulas, and charts are common in university materials. In these cases, even when the PDF includes selectable text, the information is often represented as a collection of objects positioned on the page, with textual fragments distributed in ways that do not correspond to logical units that can be readily reconstructed, such as tabular rows and cells or mathematical symbols. These portions are difficult to convert into machine readable formats precisely because of the heterogeneity of styles and the need to recover structure and content simultaneously (Horn & Keuper, 2025).

The chatbot improvement work therefore introduces a parsing process for the teaching materials, aimed at transforming the PDFs in the knowledge base into a representation that is more stable and better suited to subsequent indexing and retrieval stages. Parsing is understood here as the set of operations through which a document is analyzed, interpreted, and converted into an explicit structure, in which the content is no longer described merely as elements laid out on a page but is instead mapped onto logically organized and semantically coherent components. In other words, parsing enables the transition from a format optimized for visual presentation to one optimized for automated processing, making both the text and its structure more usable for the retrieval system and, consequently, for the language model during generation (Zhang et al., 2024).

To express this output, the Markdown format is adopted, as it represents an effective compromise between human readability and machine processability. Markdown is a lightweight markup language based on plain text that makes the document structure explicit through minimal syntactic conventions, for instance, headings to define section hierarchies, breaks to separate paragraphs, and dedicated constructs for lists, formulas, or tables.

From an operational standpoint, to implement the conversion of materials within a timeframe compatible with the project's constraints, LightPDF was selected, a web-based service that provides a dedicated functionality for converting PDF documents into Markdown files. This choice was primarily motivated by practical considerations, as the tool allows conversion directly in the browser without requiring local installations or library configuration, thereby

significantly reducing initial friction and setup time. The platform claims a specific focus on preserving structural elements that are most relevant for producing a usable Markdown output [12]. The resulting Markdown documents were retained under the same naming convention as the corresponding original PDFs. This decision serves a purely functional purpose, as it enables immediate replacement within the folder hosting the knowledge base, avoiding the need to manually rename files or update references.

5,3Integration of Course Video Lectures

The material available to students is not limited to the slides used during teaching, but also includes a complete set of video lectures, which represent a particularly valuable resource because they allow students to recover not only the content, but also the reasoning and explanations provided by the instructor during class. In general, these video lectures derive from recordings of real lectures delivered in previous years and therefore constitute a natural extension of "traditional" materials, offering clarifications, examples, and intermediate steps that may appear more condensed in slide decks. In the context of this project, however, for practical reasons related to file sharing and management, the original format of full-length recordings was not adopted. Instead, selected excerpts of the recordings were used, divided into shorter sections and organized coherently by topic. This choice made the content easier to consult and reuse within the chatbot, facilitating the association between specific student requests and targeted portions of explanation, without having to manage excessively long or poorly indexed files.

Before integrating the video lectures into the chatbot, it was necessary to define an implementation approach consistent with the way language models can actually use audiovisual content. LLMs do not "consume" video as a student would, by continuously following images and audio, but operate on representations compatible with their processing mechanisms, which in practice require transforming video into textual, indexable information. Consequently, the integration did not concern the video files themselves, but rather the extraction and structuring of the informational content conveyed by the lectures, so that it could be retrieved and reused by the system during response generation. In this context, the literature review played a decisive role in guiding the choice of the most efficient strategy, identified in the work of Caccavale, which provided a methodological reference for converting video lectures into a format suitable for the chatbot corpus. The process begins with collecting and selecting relevant recordings, continues with extracting the audio track from the video files, and includes a diarization phase,

which is useful for distinguishing different speakers and separating student interventions from those of the instructor. Once the audio has been segmented, a speech-to-text model is applied to obtain the transcript, ideally preserving timestamps and segmentation into coherent blocks (Caccavale et al., 2025).

The files used in this work consisted of 35 videos in .mp4 format, organized into seven groups based on the macro-topic addressed. The first step involved standardizing the naming convention in order to make the dataset more coherent and easier to manage during subsequent indexing and integration phases. A unique format of the type *1.1.1_topic* was therefore adopted, where the numerical component hierarchically identifies the topic at progressively finer levels of granularity, while the final part of the title specifies the file content in natural language. The presence of a third numerical level was not imposed rigidly, since not all macro-topics required further internal subdivisions; consequently, in some cases the structure stopped at the second level while preserving overall classification consistency. Using VLC Media Player, the files were converted from .mp4 video format to audio-only .mp3, thereby simplifying the subsequent transcription step. Then, through the Hugging Face Spaces platform, OpenAI's Whisper Large V3 model was used to perform speech-to-text transcription of the audio tracks. Whisper is an automatic speech recognition model trained on large amounts of multilingual audio data and designed to convert recordings into text even in the presence of noise, different accents, or suboptimal audio quality. In addition to producing complete transcripts, the model can robustly handle the continuous speech typical of lectures, maintaining good accuracy even when the discourse includes technical terminology or explanations delivered at a fast pace. The Large V3 variant represents one of the most performant configurations in the family and was selected to maximize transcription quality and reduce the need for manual corrections, making the generated text sufficiently reliable to be subsequently reorganized and integrated into the chatbot's informational corpus [N].

The final output of the transcription process consisted of 35 .txt files. Given the predominantly linear structure of the video lectures, delivered almost entirely by a single instructor, with limited and sporadic interventions, it was considered appropriate to adopt a simple format such as plain text, which is sufficient to represent the informational content clearly and continuously for integration into the corpus. In contexts characterized by more complex conversational dynamics, for instance, lectures with multiple participants who intervene frequently, or with rapid exchanges among different speakers, it is preferable to use more structured transcription formats capable of preserving metadata such as speaker identification, turn segmentation, and

timestamps, thereby maintaining a representation that is more faithful and more reusable during subsequent indexing and information-retrieval stages. In any case, in order to maintain consistency across the formats used in the knowledge base and to ensure that the information extracted from video lectures was leveraged by the chatbot as effectively as possible, the files were converted from .txt to .md, consistent with the approach already applied to materials originally available in PDF format. In this case as well, adopting Markdown provided concrete advantages, as it allows the introduction of explicit structure through headings, sub-sections, and emphasis, making content hierarchy clearer and facilitating indexing and targeted retrieval. Moreover, compared to plain text, Markdown enables cleaner separation of conceptual blocks, improving chunking quality and reducing the risk that distinct explanations are merged into incoherent segments. The resulting files were stored in a dedicated folder created within the OneDrive repository hosting the chatbot knowledge base. This folder, titled *Videolectures*, was used to clearly separate video-lecture transcripts from slide-related files, thereby improving the overall organization of the repository and simplifying both content management and subsequent corpus updates and maintenance activities.

5,4Update of Prompt, Glossary and Bridge notes

To make the described modifications effective, it was necessary to intervene on the components that constitute the chatbot's configuration baseline. In particular, the system prompt and the two supporting files, namely the glossary and the bridge notes, were updated, as together they define the conceptual boundaries within which the model operates and guide how the information contained in the knowledge base is interpreted and reprocessed. The integration of new content, such as video lecture transcripts, required explicit alignment of these components in order to ensure terminological consistency, thematic continuity, and effective use of the expanded material during response generation.

Starting from the modifications made to the prompt, which represents the most fundamental component in the chatbot configuration, it was necessary to intervene on the references to the course slides contained within it. Since the slides were replaced with updated versions, all titles mentioned in the prompt were systematically revised in order to ensure consistency between the instructions provided to the model and the documents actually present in the knowledge base. Particular attention was paid to file extensions. In addition to the slide files and video lecture transcripts, it was decided to parse the bridge notes and the glossary as well, standardizing the entire corpus in Markdown format. Consequently, in all instances where

references with the `.docx` or `.pdf` extension appeared, these were manually replaced with the `.md` extension, thereby ensuring full alignment between the repository structure, the prompt configuration, and the materials effectively used by the system.

With regard to the glossary, since the previous version referred to slides that were no longer present in the repository and exhibited a particularly complex structure, it was decided to rewrite it entirely. The objective was to simplify the organization of the document and make it more functional for the retrieval process, avoiding redundancies and overly articulated hierarchical structures that could hinder the association between a user's question and the correct file. The new version adopts a linear and orderly structure. For each block, the name of the corresponding `.md` file that transcribes a specific set of slides is first indicated. This is followed by a *Scope* section, which provides a concise summary of the file's content and serves to delineate its thematic boundaries. A bullet-point list then presents a brief description of the topics covered, making the main concepts addressed in the document explicit and easily identifiable.

Within these descriptions, several recurring categories may appear. The *Metrics and formulas* entry signals the presence of quantitative indicators and mathematical relationships relevant to the analysis of production systems, such as performance measures or fundamental laws. The *Models* entry indicates the introduction or discussion of theoretical or analytical models used to represent and study production processes. The *Formulas* entry highlights the presence of specific mathematical expressions that are derived or applied in the file, distinguishing them from purely descriptive metrics.

Finally, for each file a section entitled *Typical questions routed here* was added, reporting examples of representative questions that should be directed to that document. This section does not serve a merely descriptive purpose, but is intended to guide the system in the semantic routing process, improving the likelihood that a given question will be associated with the most relevant file within the knowledge base.

5,5Post Improvement Testing of the Chatbot

Once all the described modifications had been completed, it became necessary to assess their impact on the overall functioning of the system. The update of the knowledge base, the rewriting of the glossary, the modification of the prompt, and the integration of video lectures constitute

structural interventions that directly affect the chatbot's behavior, making an empirical evaluation of performance in the new configuration appropriate.

To this end, a new operational test of the chatbot was conducted, with the aim of analyzing its performance after the implemented improvements. In order to ensure a fair and methodologically coherent comparison, the system was presented with the same theoretical questions, the same quizzes, and the same numerical exercise that had been used in the previous comparative test. This choice makes it possible to construct a direct comparison between the initial and the updated versions of the chatbot, providing a clear picture of the differences in terms of accuracy, completeness, response coherence, and ability to handle applied content. In this way, the effect of the introduced modifications can be isolated, avoiding the risk that variations in performance might be attributable to changes in the question set or in the level of difficulty of the tasks. The same evaluation criteria adopted in the previous comparative test were therefore maintained, allowing the differences between the initial and updated versions of the chatbot to be quantified numerically. Methodological consistency in the evaluation process is essential to ensure the comparability of results and to attribute any observed performance variations exclusively to the implemented modifications.

The scores assigned to the chatbot's responses after the improvement interventions are reported below, organized according to the same three categories used previously, namely *completeness*, *coherence*, and *citations*. This structure allows for a separate analysis of content completeness, logical and structural coherence of responses, and correctness in the use of knowledge base sources. All responses generated during this second round of testing are reported in full at the end of the document, in Appendix B, in order to ensure transparency and traceability in the analysis of the results.

Question	Completness	Coherence	Citations
Describe the main concepts of the Group Technology and indicate some advantages with respect to the functional layout.	2	2	2
What is the bottleneck rate of a line?	2	2	2
Which are the characteristic of Simulation in the context of Industry 4.0?	2	2	2
Which is the right Kendall notation to indicate a workstation in which the inter-arrival time follows a normal distribution, the processing time follows an exponential distribution, there are three machines in parallel and the queue is unlimited?	2	2	2

At which level of the hierarchy are the data concerning the realt time monitoring of the production machines used to take decisions?	2	2	2
A workstation has a processing rate of 6 j/h. The inter arrival time of jobs to the workstation is 20 min. Which is the utilization of the workstation?	2	2	2
In a system the mean arrival rate is λ = 6 jobs/hours, and it is exponentially distributed. The mean service times is E[Ts] = 0.5 hours, with a squared coefficients of variation Cs2 = 0.8. Which is the number of parallel machines needed to get a utilization lower or equal to 0.9? Considering the obtained number of parallel machines, compute the CT, the WIP and the TH of the system.	1	1	2
Sum	13	13	14
Total	40		

Table 5: Performance Grid by Question and Characteristic

A clear and substantial difference in the response to Question 1 is immediately evident. In the previous version, the chatbot's guardrail mechanisms had prevented the generation of an answer, classifying the topic of the question as external to the course material. This behavior was attributable to a difficulty in correctly recognizing the semantic relevance of the request with respect to the contents available in the knowledge base.

In the updated version, thanks to the improved ease of information retrieval enabled by parsing materials in Markdown format and to the presence of a more structured and coherent glossary aligned with the new files, the chatbot is instead able to correctly identify the topic as belonging to the theoretical corpus of the course. As a result, the system passes the guardrail filter and is able to provide a relevant response. The answer produced is correct according to all the adopted evaluation criteria; consequently, the chatbot's overall score shows a significant increase compared to the previous version.

With regard to the scores assigned to the *citations* category, the chatbot achieves the maximum score in all responses of the second testing round. In the updated version, the system is able to cite accurately and without errors one or more relevant slide decks and one or more pertinent video lectures for each question, demonstrating greater precision in linking the generated response to sources actually present in the knowledge base. In the previous version, by contrast, particularly in the cases of Questions 2 and 3, the chatbot did reference slide decks in which the topic appeared, but not as the main focus. Moreover, inaccuracies were present in the numbering of the cited slides, indicating less accurate retrieval and weaker coherence between the referenced content and the actual structure of the files.

In the questions for which the chatbot already exhibited correct behavior in the first version, the system continues to respond appropriately after the introduced modifications. In particular, for the other two theoretical questions and for the quizzes, it again achieves the maximum score, confirming that the implemented interventions did not compromise areas in which performance was already satisfactory. However, a further improvement can be observed at the formal level. The structure of the responses now appears even more homogeneous than in the previous version, which was already of a high standard. The internal organization of content, the logical sequencing of steps, and the integration of citations display greater uniformity, indicating improved coherence between the prompt, the glossary, and the corpus, with a positive impact on the overall quality of the generated output.

In the final exercise, prior to the modifications, the chatbot had provided incorrect results that were also significantly distant from the expected theoretical values. The error did not concern minor numerical deviations, but rather revealed a more structural difficulty in setting up the calculation model and consistently applying the correct formulas. The first solution adopted an excessively simplified approximation, treating the multi-parallel-machine system as a single aggregated server and directly applying Kingman's VUT formula for a G/G/1 system: an approach that is not consistent with the nature of the problem, which instead concerns a G/G/c system. This led to a substantial underestimation of cycle time and, consequently, of work-in-process (WIP). The second response, by contrast, explicitly recognizes the multi-server structure of the system, starts from the M/M/c model with Erlang C, and subsequently introduces the variability factor to obtain a G/G/c approximation, more faithfully following the methodological logic presented during the course.

After the introduced modifications, the chatbot is still unable to solve the exercise by producing values fully coincident with the theoretical solution; however, the obtained results are now significantly closer to the expected ones. The remaining deviation appears limited and compatible with differences arising from the approximation method adopted rather than from a conceptually incorrect problem formulation. It is therefore reasonable to hypothesize that the residual discrepancy is mainly attributable to an approximation factor intrinsic to the underlying language model of the chatbot, which, while correctly applying the theoretical structure, operates with simplified numerical procedures, as it is not designed as a computational tool but rather as a text-generative system with approximate mathematical capabilities.

	Completness	Coherence	Citations	Sum
Pre	10	9	10	29
post	13	13	14	40
Delta	3	4	4	11
Delta %	30%	44%	40%	38%

Table 6: Pre and Post modification Performances

The table provides an exceptionally clear summary of the quantitative impact of the introduced modifications. The overall score increases from 29 in the pre-intervention phase to 40 in the post-intervention phase, corresponding to an absolute gain of 11 points, or a 38 percent improvement. This represents a substantial variation and confirms numerically what had already emerged qualitatively from the analysis of individual responses.

Examining the results by category, improvements can be observed across all evaluated dimensions. *Completeness* increases from 10 to 13, corresponding to a 30 percent improvement. This indicates that, following the modifications, the chatbot is able to cover the elements required by the questions in a more extensive and structured manner, likely due to improved information retrieval and the expansion of the corpus through the integration of video lectures.

The *coherence* dimension shows an even more pronounced improvement, rising from 9 to 13, with a 44 percent increase: the largest among the three categories. This result is particularly noteworthy, as it suggests that the changes made to the prompt, the glossary, and the file structure did not merely increase the amount of retrieved content, but also enhanced the logical organization and argumentative robustness of the responses.

The *citations* dimension also records a significant increase, from 10 to 14, corresponding to a 40 percent improvement. This outcome directly reflects the effectiveness of format standardization, extension updates, and the explicit inclusion of video lectures within the citation section of the prompt. The chatbot not only cites more frequently, but does so more accurately, in a more relevant manner, and without formal errors.

Overall, the distribution of score increases indicates that the implemented improvements had a systemic effect rather than benefiting a single dimension in isolation. The homogeneous growth across all categories confirms that the technical and structural interventions strengthened the overall quality of the system in an integrated way, making the chatbot more complete, more coherent, and more reliable in terms of source traceability.

6 Experimental Design for Evaluating the Educational Impact of the AMPS Chatbot

The objective of this thesis, and more broadly the idea behind the project of developing a chatbot dedicated to the course *Analysis and Management of Production Systems*, is not limited to a purely theoretical exercise or to a technical demonstration of the potential of generative AI. From the beginning, the project has been conceived with a clear practical purpose, namely to explore how the new possibilities offered by artificial intelligence can be used in a structured and conscious way to support university teaching. The development of the chatbot should therefore not be considered an end in itself, but rather a tool through which to explore how a domain specific AI system, trained exclusively on the official course materials, can assist students during individual study, clarify theoretical concepts, and support the resolution of complex exercises. The ambition is to create a digital assistant capable of providing responses that are coherent, traceable, and aligned with the instructor's content, thereby reducing interpretative ambiguities and strengthening students' understanding of production system analysis models. From this perspective, the project aims to make a concrete contribution to the preparation of students approaching this subject, offering a tool that complements traditional lectures and course materials. The final phase of the thesis, dedicated to the design of an experiment to evaluate the chatbot's effectiveness, naturally follows from this objective, since it makes it possible to verify whether, and to what extent, the use of the chatbot can translate into a measurable improvement in learning outcomes.

It is important to note that, due to the time constraints of the project, this thesis does not aim to fully conduct and complete the empirical phase required to evaluate the chatbot's impact on learning. Instead, the analysis focuses on the rigorous design of the experimental framework necessary to measure such an impact in a methodologically sound and replicable way. The purpose of this chapter is therefore not to present quantitative results or to draw definitive conclusions regarding the effectiveness of the tool, but rather to define in a structured manner the experimental framework within which such effectiveness can be assessed. The chapter will describe the application context, the relevant variables, the procedures for assigning participants to experimental conditions, the instruments used to measure learning outcomes, and the evaluation criteria. Data collection, statistical analysis of results, and the final interpretation of the chatbot's impact are activities that are left to future developments.

6,1 Literature Review on the Evaluation of AI Supported Learning

As already done in the previous sections of this thesis, in order to ensure the methodological rigor of the proposed approach, a systematic literature search was conducted before proceeding with the detailed definition of the experimental design. The objective of this preliminary analysis was to identify the methodologies most commonly adopted to evaluate the contribution of technological tools, and in particular of artificial intelligence based systems, to learning processes. Through the comparative analysis of the selected papers, the most recurring research designs were examined, including the methods used to assign participants to experimental conditions, the types of outcomes measured, the statistical techniques employed, and the main methodological limitations highlighted by the authors themselves. A systematic comparison between different approaches makes it possible to identify methodological convergences and shared best practices within the scientific community, reducing the risk of arbitrary choices or decisions that are not aligned with academic standards. Basing the experimental design on frameworks that have already been tested, discussed, and replicated in the literature therefore increases the credibility of the proposed approach, ensuring that methodological decisions are consistent with established research practices.

6,1,1 Search Strategy and Keyword Selection

Also in this phase of the literature review, in line with the methodological approach adopted in the previous chapters, the selection of articles was guided by the construction of structured queries to be used within online academic databases, specifically Scopus and Google Scholar. The definition of these queries was not random, but derived from the systematic combination of three distinct conceptual groups, each aimed at clearly delimiting the object of investigation.

The first group concerns technology and includes the terms *ChatGPT*, *LLM*, and *Generative AI*. The inclusion of these keywords reflects the need to identify studies focusing on tools based on large language models or, more broadly, on generative artificial intelligence systems. The inclusion of *ChatGPT* makes it possible to capture the most recent contributions specifically related to large language models applied to education, while *LLM* and *Generative AI* broaden the scope of the search by including studies that analyze similar tools even without direct reference to a specific commercial platform.

The second group focuses on the educational context and includes the terms *learning outcomes*, *student performance*, and *education*. The aim is to restrict the search to studies that measure concrete effects on learning, avoiding works that are purely theoretical or descriptive. The attention is therefore directed toward contributions that analyze measurable results, such as improvements in scores, academic performance, or skill development.

The third group concerns the methodological approach and includes the terms *RCT*, *experimental*, and *meta-analysis*. This choice is motivated by the intention to identify studies characterized by a robust methodological framework capable of credibly isolating the causal effect of technological tools on learning. The reference to RCTs and experimental studies allows the selection of contributions involving control groups and structured assignment to experimental conditions, while the term *meta-analysis* enables the inclusion of quantitative synthesis studies that provide an aggregated view of the available empirical evidence.

The combination of these three groups, through the use of logical operators within the search queries, made it possible to identify articles located at the intersection between generative AI technologies, the university educational context, and an experimental methodological approach, thereby ensuring coherence between the analyzed literature and the objective of this chapter.

6,1,2 Overview of Selected Studies

The corpus selected for this literature review consists of a total of 29 contributions and covers a broad time span, from 2007 to 2026. The yearly distribution shows a strong concentration of recent studies, with 17 papers published between 2024 and 2026, corresponding to approximately 59 percent of the total. The main peak occurs in 2025, with 12 publications, reflecting the rapid growth of academic interest in the use of generative artificial intelligence tools and large language models within the educational domain.

The following table presents the complete list of papers analyzed for this literature review:

Authors	**Title**	**Year**	**Summary**
Bastani et al.	Generative AI Without Guardrails Can Harm Learning	2025	Examines how ChatGPT access without pedagogical constraints leads to "shortcut" learning and lower exam scores in math.

Kestin et al.	AI tutoring outperforms in-class active learning	2025	Shows that a specifically designed AI tutor can be more effective than traditional active learning methods in physics.
Wu & Yu	Do AI chatbots improve students learning outcomes?	2024	Comprehensive meta-analysis assessing the overall effectiveness of chatbots in various educational contexts.
Wang & Fan	The effect of ChatGPT on students' learning performance...	2025	Investigates how ChatGPT impacts not just grades, but perception and critical thinking skills.
Barcaui	ChatGPT as a cognitive crutch	2025	Explores the "crutch" effect where students rely on AI to solve tasks but fail to retain the knowledge long-term.
Kalam et al.	ChatGPT as a Learning Tool for Medical Students	2025	Evaluates the integration of GenAI in medical education for clinical reasoning and knowledge acquisition.
Saloojee et al.	AI at the bedside	2026	Analyzes the impact of using ChatGPT for real-time clinical decision support and medical learning.
Cheng et al.	Asking generative artificial intelligence the right questions	2025	Highlights the importance of "prompt engineering" and question quality in improving student writing.
Hanshaw et al.	An Experimental Study of the Effect of AI Course Assistants	2025	Tests whether automated course assistants improve final grades in higher education.
Henkel et al.	Effective and Scalable Math Support: AI-Tutor in Ghana	2024	Evaluates a low-cost AI tutor's ability to improve math achievement in developing regions.
Vanzo et al.	GPT-4 as a Homework Tutor Can Improve Student Engagement	2025	Explores how advanced LLMs (GPT-4) act as effective tutors to increase engagement and learning.
Wecks et al.	Generative AI Usage and Exam Performance	2024	Investigates the correlation (and potential negative impact) between high AI usage and final exam scores.
Benedek & Sziklai	Impact of AI Tools on Learning Outcomes	2025	Warns about over-reliance on AI leading to a decrease in independent problem-solving skills.
Fischer et al.	AI Tutoring Enhances Student Learning Without Crowding Out Reading	2025	Finds that AI tutoring is an additive benefit that doesn't reduce student effort in traditional reading.

Junpeng et al.	Effect of intelligent tutoring system-delivered scaffolding	2026	Studies how step-by-step "scaffolding" in AI systems helps students master complex concepts.
Razzaq et al.	Effect of Immediate Feedback on Math Achievement	2020	Classic study on how instant feedback (a core AI feature) significantly boosts math mastery.
Roschelle et al.	Online Mathematics Homework Increases Student Achievement	2016	Longitudinal evidence that online tutoring platforms (ASSISTments) improve math scores.
Pane et al.	Effectiveness of Cognitive Tutor Algebra I at Scale	2014	Evaluates the "Cognitive Tutor" software's impact on Algebra performance across diverse districts.
Ritter et al.	What Evidence Matters? A randomized field trial	2007	Early evidence of how intelligent tutoring systems (ITS) outperform traditional instruction in Algebra.
Wijekumar et al.	Large-scale RCT... structure strategy to improve reading	2012	Focuses on using an ITS to improve nonfiction reading comprehension in 4th graders.
Wijekumar et al.	High-fidelity implementation of web-based ITS	2013	Examines how the quality of implementation affects the success of AI-based reading tutors.
Wijekumar et al.	Multisite Randomized Controlled Trial...	2014	Further validation of the structure-strategy ITS across multiple geographical sites.
McGuire et al.	Results from a randomized controlled trial in ASSISTments	2017	Evaluates the efficacy of the ASSISTments platform for middle school math homework.
Feng et al.	Implementing and Evaluating ASSISTments Online Math	2023	Modern evaluation of a long-standing ITS platform's impact on remote and hybrid learning.
Craig et al.	The impact of a technology-based math after-school program	2013	Tests the use of ALEKS and other platforms in extracurricular settings.
Various	The Effects of ALEKS on Mathematics Learning in K-12	2021	Aggregated findings on the ALEKS platform, showing positive gains in K-12 math.
Roscoe et al.	The Writing Pal Intelligent Tutoring System	2014	Describes the development and testing of an ITS designed specifically for writing strategies.
Létourneau et al.	A systematic review of AI-driven ITS	2025	Summarizes the current state of AI tutors, highlighting shifts from rule-based to generative models.

Slijepcevic & Yaylali	Leveraging "Khanmigo" Generative AI-Powered Tool	2025	Investigates the use of Khan Academy's AI (Khanmigo) for personalized science learning.

Table 7: Literature review corpus

6,1,1 Synthesis of the Literature

The increasing diffusion of generative artificial intelligence tools, and in particular chatbots based on large language models, has made it necessary to develop rigorous methodologies to evaluate their impact in educational contexts. In recent years, the academic literature has shown a strong increase in studies aimed at understanding whether and how these tools can improve students' learning. However, one of the central elements that emerges from the analysis of existing contributions is that the concept of "performance improvement" is not univocal. Many studies highlight the need to distinguish between two different dimensions, on the one hand assisted performance, that is the ability of students to obtain better results when the tool is available during the completion of tasks, and on the other hand real and transferable learning, that is the ability to solve problems even when technological support is not available. This distinction is fundamental because several works show that the use of generative AI tools may improve results during practice but does not always translate into a real acquisition of skills. In some cases, the improper use of the tool can instead lead to a form of cognitive dependency, in which students rely on AI to quickly obtain a solution without developing a deep understanding of the underlying concepts.

In light of these considerations, the literature has progressively developed a series of methodological approaches to measure more precisely the effect of AI technologies on learning. Among these, the most robust method from the perspective of causal inference is represented by randomized controlled trials (RCTs). In this type of experimental design, access to the tool is randomly assigned to one group of students, while another group acts as a control. This structure makes it possible to isolate the effect of the intervention and reduce the risk that observed differences in results are due to pre existing differences among students. Several recent studies apply this approach both in school and university contexts. Some works directly compare students who use a chatbot with students who follow traditional teaching methods, while others introduce multiple experimental conditions to analyze different ways of integrating AI into the educational process.

Alongside individual RCTs, the literature often adopts experimental designs based on cluster randomization. In this case, the assignment to the treatment or control group takes place at the

level of a class, school, or student group rather than at the individual level. This solution is particularly useful when there is a risk that students share tools or solutions among themselves, generating contamination between groups. In real educational contexts, where interaction among students is frequent and difficult to control, cluster randomization therefore represents an effective compromise between methodological rigor and operational feasibility. Studies conducted in different educational settings show that this approach makes it possible to implement large scale experiments while maintaining a good level of internal validity.

Another stream of literature concerns observational studies that analyze the relationship between the spontaneous use of generative AI tools and students' academic performance. In these studies, the objective is not so much to evaluate the effect of an intervention designed by researchers, but rather to understand how the spontaneous adoption of AI is modifying learning dynamics. For example, authors analyze the correlation between the frequency of ChatGPT use and exam results, or study changes in the way students produce written assignments. However, this type of study presents important methodological limitations, mainly related to the difficulty of reliably measuring the actual use of the tool and isolating the effect of AI from other factors that may influence academic outcomes.

To integrate and systematize the results of these different research streams, several meta analyses and systematic reviews have also been published in recent years. These studies synthesize the results of numerous experiments and empirical analyses, seeking to identify common patterns and evaluate the average magnitude of the observed effects. Overall, literature reviews suggest that AI tutoring tools can have positive effects on learning, particularly when they are designed to provide immediate feedback, personalized support, and guided practice activities. However, the same studies emphasize the high heterogeneity of results and the importance of carefully considering the context of use, the pedagogical design of the tool, and the way it is integrated into the curriculum.

A central aspect that emerges from the analysis of the studies concerns the definition of learning outcomes. Many works highlight the need to distinguish between performance during practice and actual learning measured through tests conducted without technological support. The most rigorous studies therefore introduce a clear separation between activities carried out with AI assistance and evaluation moments in which the tool is not available. This approach makes it possible to verify whether improvements observed during practice translate into a real development of competencies. Some experiments also include a delayed post test, that is a

second evaluation conducted after a certain period of time, with the aim of measuring students' ability to retain and transfer the knowledge acquired.

In addition to outcomes directly related to learning, the literature also considers a series of secondary indicators related to the educational process. These include, for example, time devoted to study, levels of engagement, perceived motivation, and student satisfaction. In some cases, AI tutoring is also evaluated in terms of efficiency, that is the ability to obtain similar or better results than traditional methods with a lower investment of time or resources. This type of measure is particularly relevant for educational institutions, which must evaluate not only the pedagogical effectiveness of these tools but also their sustainability and scalability.

Another important methodological dimension concerns process measures, that is indicators describing how and how much the tool is actually used by students. Several studies highlight that the impact of a chatbot may depend significantly on the intensity and quality of the interaction. For this reason, many experiments integrate logging systems that record the number of interactions, the duration of sessions, the type of requests submitted, and the time spent on the platform. These data make it possible to analyze the so called dosage of the intervention and better understand the mechanisms through which AI influences the learning process.

Based on the evidence collected, the literature progressively converges toward a set of methodological recommendations for the evaluation of AI tools in educational contexts. First, it is generally preferable to adopt a randomized experimental design, possibly at the cluster level, in order to ensure a credible estimate of the causal effect. Second, it is essential to include at least one measure of learning without technological support, so as to distinguish between assisted performance and the real acquisition of competencies. Third, the definition of the intervention must be sufficiently precise to allow the study to be replicated, clearly specifying how the chatbot is used, the type of feedback provided, and the educational context in which it is integrated.

Finally, the analysis of the studies highlights the importance of considering aspects related to the acceptability and fairness of the intervention. Some works show that perceptions of unfairness or competitive advantage may influence students' behavior and compromise the validity of the experiment. For this reason, several researchers suggest adopting mechanisms such as delayed access, which allow the control group to use the tool once the data collection phase has been completed.

Overall, the literature therefore provides a relatively clear methodological framework for designing experiments aimed at evaluating the impact of chatbots on learning. The body of evidence analyzed suggests that a robust evaluation should combine a controlled experimental design, a clear distinction between assisted performance and real learning, a data collection system that includes both cognitive outcomes and process indicators, and an analysis plan capable of accounting for variability in the use of the tool. These principles constitute the methodological foundation upon which replicable and comparable experimental protocols can be built, capable of providing credible evidence on the educational value of artificial intelligence systems applied to teaching.

6,2Design of the experiment

6,2,1 Objectives of the Experiment

The main objective of the experiment is to evaluate whether the use of a domain specific chatbot can produce measurable benefits in the learning process of students attending the course *Analysis and Management of Production Systems*. Although the development of the chatbot itself represents a relevant technological achievement, the broader aim of the project is to understand whether a tool of this kind can provide a concrete contribution to the teaching of the course and support students in understanding complex concepts related to the analysis and management of production systems.

As highlighted in the literature review, evaluating the educational impact of artificial intelligence based tools requires particular attention to the definition of what is meant by "performance improvement".

In light of these considerations, the experiment proposed in this chapter is designed not only to observe how students use the chatbot during the resolution of a complex problem, but also to evaluate whether such use translates into an actual improvement in their understanding of the subject. The focus of the analysis is therefore placed on students' ability to apply the models and analytical tools presented in the course to address problems related to the management of production systems, rather than on the simple ability to obtain correct answers with the assistance of artificial intelligence.

More specifically, the experiment aims to address three main research questions. The first concerns the effect of chatbot use on students' learning outcomes. The objective is to verify whether students who have access to the chatbot demonstrate a higher level of understanding

of the course content compared with those who rely exclusively on traditional learning resources. The second research question concerns the role of the chatbot in the problem solving process. In this case, the experiment aims to analyze whether the availability of an AI based support tool can improve both the efficiency and the quality of the solutions developed by students when performing complex tasks. The third research question focuses on the broader learning experience, with the objective of understanding whether the use of the chatbot influences students' level of engagement, their confidence in solving problems, and their perception of the usefulness of the tool.

In light of the objectives described above, it is appropriate to formalize a set of research hypotheses that will guide the design of the experiment. The explicit definition of hypotheses helps clarify the relationships between variables that the study intends to test and allows the development of an experimental protocol consistent with these objectives.

H1 – Effect on learning.
Students who have access to the chatbot during study and problem solving activities achieve better results in individual evaluation tests than students who do not use the chatbot.

H2 – Effect on performance during problem solving.
Groups of students who use the chatbot produce higher quality solutions during problem solving activities compared with groups that rely exclusively on traditional learning resources.

H3 – Effect on the learning experience.
The use of the chatbot increases students' level of engagement and their perceived usefulness of the learning activity.

6,2,2 Experimental Context and Educational Setting

The proposed experiment is embedded within two group work sessions already included in the course *Analysis and Management of Production Systems*. These activities represent a particularly suitable context for experimenting with study support tools, as they require students to apply in an integrated manner the theoretical knowledge acquired during the course in order to solve complex problems related to the analysis of production systems.

Students participate in the activity organized in small groups composed of two or three members. This mode of work is consistent with the teaching approach adopted in the course, which includes collaborative problem solving sessions in which students are encouraged to discuss and compare possible solution strategies. The small size of the groups also helps

maintain a good level of individual participation, reducing the risk that some students assume a passive role during the activity.

The experiment takes place over two work sessions, each lasting between one and a half and two hours. At the beginning of each session, each group receives a document containing a set of exercises to be solved. In particular, the assigned task consists of solving three or four exercises related to the topics covered in the course, designed to require not only the direct application of formulas or standard procedures, but also the ability to interpret the problem, identify the appropriate analytical model, and justify the steps of the solution.

Once the work has been completed, the groups are required to upload their solutions through the course portal. This submission method allows all the assignments produced by the students to be collected in a structured way and ensures that the material can later be analyzed and evaluated in a consistent manner.

The evaluation of the solutions will be carried out by the teaching assistants who collaborate in the course activities. For this purpose, the evaluators will be provided with an official solution to the exercises, prepared by the instructor or by the course coordinators, which will serve as the reference for assigning scores. This approach helps ensure greater consistency in the grading process and reduces the risk of discretion in the evaluation of the assignments.

With regard to the learning resources available during the activity, all students have access to the course slides, which represent the main reference material for studying the topics covered. The slides therefore constitute the common informational base on which students can rely while solving the exercises. In this way it is possible to ensure that any differences in results between groups do not depend on the availability of different teaching materials, but only on the experimental conditions introduced in the design of the experiment.

It is important to note that, at the time of writing this thesis, the exact content of the exercises that will be used in the work sessions has not yet been finalized. The detailed design of the problems will be developed at a later stage, taking into account both the learning objectives of the course and the need to include exercises that require the application of the main analytical tools presented during the lectures. In particular, the problems should be sufficiently complex to allow the observation of meaningful differences in the solution strategies adopted by the students and in the quality of the solutions produced.

Overall, the structure of the experimental activity is designed to replicate as closely as possible an authentic learning situation, in which students work in groups to address complex problems

using the resources available to them. This choice makes it possible to evaluate the effectiveness of the chatbot not in an artificial or highly controlled environment, but within a realistic educational context that reflects the learning practices normally adopted in the course.

6,2,3 Definition of the Intervention

This subsection aims to clearly and operationally define what is meant by the **intervention** within the experiment. In experimental designs applied to the educational domain, it is essential to specify precisely what treatment is introduced and how it differs from the control conditions. An explicit definition of the intervention makes the experimental protocol replicable and allows the results of the experiment to be interpreted correctly.

In the context of this research, the intervention consists of allowing some groups of students to use the chatbot developed for the course *Analysis and Management of Production Systems* as a support tool during the completion of the assigned activities.

In this way, the tool functions as a digital assistant capable of helping students clarify theoretical concepts, better understand the analytical models discussed in the course, and receive useful guidance when working on the exercises.

Within the experiment, students assigned to the treatment groups will be allowed to use the chatbot freely during the work sessions. No specific limitations will be imposed regarding the number of interactions, the type of questions that can be asked, or the moment in which the tool can be consulted. Students will therefore be able to decide independently whether and how to use the chatbot to support their problem solving process.

This methodological choice reflects the objective of observing students' behavior in a learning situation that is as natural as possible. Allowing free use of the tool makes it possible to analyze how students spontaneously integrate the chatbot into their study strategies and problem solving processes. In this way, the intervention does not simply test the technical capabilities of the system, but also allows an evaluation of its potential value as a study support tool within a real educational context.

At the same time, it is important to emphasize that all students, regardless of their group assignment, will continue to have access to the standard learning materials of the course, primarily the lecture slides. This ensures that the chatbot represents the only substantial difference between the experimental conditions. Such a design makes it possible to isolate the

effect of the intervention more clearly and to attribute any differences in outcomes to the use of the artificial intelligence tool rather than to the availability of different informational resources.

Precisely defining the intervention therefore represents a fundamental step in the design of the experiment. This section establishes the conditions under which the chatbot can be used and clarifies the role that the tool plays within the learning activity, providing the basis for the definition of the experimental design and the comparison between the different groups of students.

6,2,4 Experimental Design and Group Assignment

The present experiment is designed with the objective of systematically evaluating the effect of chatbot use on the learning process of students. To achieve this objective, it is necessary to define an experimental design that allows a comparison between the results obtained by students who have access to the tool and those obtained by students who complete the same activity without such support. The principle underlying the experimental design therefore consists of introducing a controlled difference between groups of students while keeping all other conditions as similar as possible.

Within the context of the experiment, students participate in the activities organized in groups of two or three members, as established by the teaching structure of the course. However, the main unit of observation remains the working group, since the assigned exercises are solved collectively and the final solution is produced through a collaborative process. For this reason, the assignment to the different experimental conditions takes place at the group level rather than at the individual level.

Once the groups have been formed, they are divided into two main categories. The first set of groups represents the treatment condition and has access to the chatbot during the work sessions. Students belonging to these groups can freely use the tool as support while solving the exercises, asking questions, requesting theoretical clarifications, or seeking guidance to correctly structure the problems. The second set of groups represents the control condition and performs the same activity without the possibility of using the chatbot. Students in these groups can rely exclusively on the traditional teaching resources of the course, in particular the lecture slides.

To ensure a methodologically sound comparison between the two conditions, groups are assigned to the categories randomly. Randomization reduces the risk that pre existing characteristics of the students, such as their level of preparation, familiarity with digital tools,

or individual abilities, systematically influence the results of the experiment. When a sufficient number of groups is available, this process tends to distribute these characteristics in a balanced way across the experimental conditions. In operational terms, the groups are divided approximately equally between the two conditions, assigning about 50 percent of the groups to the treatment condition and the remaining 50 percent to the control condition. This choice is consistent with common practices in experimental design, since an equal distribution between groups maximizes comparability between the two conditions and facilitates the interpretation of potential differences in results. In this way, any variation in performance observed between the groups can be attributed more reliably to the presence or absence of the chatbot.

Another relevant aspect of the experimental design concerns the role of the two work sessions scheduled during the activity. The groups assigned to the treatment condition will maintain access to the chatbot in both sessions, while the groups belonging to the control condition will not have access to the tool in either session. In other words, the distinction between groups with chatbot access and groups without chatbot access remains unchanged throughout the entire duration of the experiment. This choice allows the experimental design to remain clear and consistent, avoiding changes in conditions between sessions that could introduce additional sources of variability that are difficult to interpret. In this way, the groups work under the same conditions during both sessions, and the only systematic difference between the two categories remains the availability of the chatbot as a support tool.

The presence of two sessions scheduled at different times also makes it possible to observe the behavior of the groups over a longer time horizon. On the one hand, it is possible to evaluate the quality of the solutions produced during the first problem solving activity. On the other hand, it becomes possible to analyze whether any differences between groups persist in the second session. This approach allows for a more comprehensive understanding of the effect of the chatbot on the problem solving activity while maintaining a simple and easily interpretable experimental structure.

6,2,5 Measurement of Learning Outcomes

Evaluating the effects of the intervention requires the definition of a measurement system that allows a clear and comparable assessment of the performance of students belonging to the different experimental conditions. In the context of the proposed experiment, learning outcomes are primarily measured through the quality of the solutions produced by students during the two

work sessions, complemented by information collected through a questionnaire administered at the end of the activity.

With regard to the evaluation of the solutions, each group will receive a score based on the correctness and completeness of the exercises completed. To ensure greater clarity in the presentation of results and easier comparability between the two sessions, it is appropriate to adopt a standardized evaluation scale. A solution consistent with the Italian university grading system is the use of a scale expressed in thirtieths. In this way, each group will receive an overall score between 0 and 30 for each of the two work sessions. The score assigned to each group will be determined by the teaching assistants responsible for grading the assignments, using the official solution to the exercises as a reference. If the assigned task includes multiple exercises, the final score may be calculated as the sum or the average of the scores assigned to the individual exercises, appropriately normalized to the final scale in thirtieths. This approach allows a consistent evaluation criterion to be maintained across different sessions and across different groups. The use of a scale expressed in thirtieths also provides an additional methodological advantage. Since the official exams of the course are evaluated using the same grading scale, the results obtained during the experiment can be easily compared with the performance of students in the official exam sessions. This comparison can also be conducted at a later stage, after the completion of the experiment. In particular, observing whether students who had access to the chatbot during the experimental activity achieve better results in exams conducted without the aid of the tool would make it possible to evaluate whether the benefits observed during the experiment translate into a lasting advantage over time. The possibility of comparing the results of the experiment with those of the official course exams is particularly relevant from a pedagogical perspective. If students who used the chatbot demonstrate better performance in an exam conducted without the tool, this would indicate that the chatbot contributed to a deeper and more durable understanding of the course content, rather than simply improving performance during assisted activities.

The presence of two work sessions held at different points in time also allows the evolution of group performance to be analyzed throughout the experiment. In particular, for each group it will be possible to observe two distinct values, corresponding to the score obtained in the first session and the score obtained in the second session. Comparing these two values makes it possible to analyze potential changes in the quality of the solutions produced over time. From the perspective of data analysis, a first summary measure can be obtained by calculating the average score of the groups belonging to the same experimental condition. In this way it

becomes possible to compare the average score of the groups that had access to the chatbot with the average score of the groups that completed the activity without such support. This comparison can be carried out separately for each of the two sessions or by considering an overall average of the two tests. A further analysis can be conducted by observing the variation in scores between the first and the second session. For each group it is possible to calculate the difference between the score obtained in the second session and the score obtained in the first. This value provides a measure of any improvement or decline in performance during the experiment. By subsequently comparing the average variation in scores between groups with chatbot access and groups without chatbot access, it becomes possible to obtain a first indication of the potential effect of the tool on the evolution of students' problem solving abilities.

In addition to the evaluation of the solutions produced by the groups, the experiment also includes the collection of information through a questionnaire administered to students at the end of the activity. As previously mentioned, the design of the questionnaire is the subject of a separate research project developed within another student's thesis. Nevertheless, the results of the questionnaire represent a valuable source of information for understanding certain aspects of the learning experience that cannot be directly observed through the evaluation of the assignments. For each question it will be possible to calculate the average value of the responses provided by students belonging to the different experimental groups. Comparing the average responses of the groups with chatbot access and those without chatbot access may provide useful insights into differences in the perception of the activity and into the potential role of the tool in supporting students' work.

Overall, the adopted measurement strategy therefore combines objective performance indicators derived from the evaluation of the solutions produced by the groups with subjective indicators related to the students' learning experience. The integration of these two dimensions allows a more comprehensive evaluation of the impact of the chatbot within the educational context of the course.

7 Conclusions

This thesis began with the development of an already existing chatbot, initially created for the *Analysis and Management of Production Systems* (AMPS) course at Politecnico di Torino. The first step in this research was to evaluate the existing version of the chatbot, considering both its technical capabilities and its educational effectiveness. A critical part of this process was exploring alternative technical solutions and identifying best practices used in the development of AI-driven educational tools. This evaluation aimed to ensure that the chatbot could be further optimized to better support students in their learning journey.

A comprehensive comparison was made between the initial version of the chatbot and commercial alternatives available on the market. By examining how these tools were implemented in similar educational settings, it was possible to identify both strengths and weaknesses in the chatbot's design. This comparative analysis also highlighted the potential for improvement in terms of user interaction, response accuracy, and alignment with course-specific content. The evaluation of commercial chatbots provided valuable insights into what features were effective in promoting student engagement and enhancing learning, as well as what areas could benefit from further development in the AMPS chatbot.

Following this evaluation, a series of improvements were made to the chatbot. Key actions involved updating the knowledge base by converting course materials into a structured format suitable for AI interaction, modifying the chatbot's prompt to ensure clearer and more contextually accurate responses, and adding text from the video lectures to further enrich the content provided by the chatbot. These improvements aimed to enhance the chatbot's ability to provide more accurate, relevant, and interactive support to students, creating a more comprehensive learning tool.

Once these changes were implemented, a **post-modification test** was conducted to assess the impact of the updates on the chatbot's performance. This test provided an opportunity to evaluate whether the modifications resulted in improvements in terms of response accuracy, user engagement, and overall usefulness as an educational support tool. The results of this test demonstrated that the modifications led to a more responsive and effective chatbot, aligning more closely with the learning objectives of the course.

In addition to these development and testing activities, this thesis also focused on designing an experiment to assess the impact of the enhanced chatbot on student learning outcomes. The

experimental design implies an analysis of student performance during group problem-solving tasks with and without access to the chatbot. This experimental framework aims to provide valuable data on whether the chatbot leads to better understanding and retention of course material, improving not only immediate performance but also long-term learning outcomes.

In conclusion, the work presented in this thesis contributes to the development of AI-driven educational tools, specifically chatbots, by outlining a clear methodology for improving and testing their effectiveness in a university course setting. While the chatbot's technical improvements are promising, future empirical research will be required to evaluate its real impact on students' learning and its ability to foster deeper, long-lasting understanding. The experiment designed in this thesis provides a solid foundation for future testing and offers a replicable approach for assessing the educational value of chatbots in higher education.

Bibliography

Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. *arXiv Preprint arXiv:1712.05181*.

Brock, D. C. (2018). Learning from artificial intelligence's previous awakenings: The history of expert systems. *AI Magazine*, *39*(3), 3–15.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., & Askell, A. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper/7017-deep-reinforcement-learning-from-human-preferences

Copeland, B. J. (2000). The turing test. *Minds and Machines*, *10*(4), 519–539.

Dinesh, T., R, A. M., Newton, T. T., & R, S. G. (2021). AI Bot for Academic Schedules using Rasa. *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 1–6. https://doi.org/10.1109/ICSES52305.2021.9633799

Dyson, S. R. (2025). *An Exploration into The Capabilities of Microsoft Copilot and Its AI Agent Creation*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*.

Haugeland, I. K. F., Følstad, A., Taylor, C., & Bjørkli, C. A. (2022). Understanding the user experience of customer service chatbots: An experimental study of chatbot interaction design. *International Journal of Human-Computer Studies*, *161*, 102788.

Horn, P., & Keuper, J. (2025). Benchmarking Document Parsers on Mathematical Formula Extraction from PDFs. *arXiv Preprint arXiv:2512.09874*.

Kreynin, I., Manek, M., & Variawa, C. (2019). CREATING A VIRTUAL CHATBOT TO SCAFFOLD SKILLS DEVELOPMENT IN FIRST-YEAR ENGINEERING EDUCATION. *Proceedings of the Canadian Engineering Education Association (CEEA)*. https://doi.org/10.24908/pceea.vi0.13784

Kuipers, B., Feigenbaum, E. A., Hart, P. E., & Nilsson, N. J. (2017). Shakey: From conception to history. *Ai Magazine*, *38*(1), 88–103.

Lederberg, J. (1990). How DENDRAL was conceived and born. In *A history of medical informatics* (pp. 14–44).

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., & Rocktäschel, T. (2020). Retrieval-augmented generation for knowledge-

intensive nlp tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474.

Magnolfi, A., Vicario, E., & Boncinelli, L. (n.d.). *AI-Powered Chatbot Assistant for Course Organization*.

Merola, C., & Singh, J. (2025). Reconstructing context: Evaluating advanced chunking strategies for retrieval-augmented generation. *International Workshop on Knowledge-Enhanced Information Retrieval*, 3–18.

Mikael, K., Öz, C., Rashid, T. A., & Nariman, G. S. (2025). A Hybrid Chatbot Model for Enhancing Administrative Support in Education: Comparative Analysis, Integration, and Optimization. *IEEE Access*, *13*, 50741–50760. https://doi.org/10.1109/ACCESS.2025.3552501

Priadko, A. O., Osadcha, K. P., Kruhlyk, V. S., & Rakovych, V. A. (2020). *Development of a chatbot for informing students of the schedule*. [б. в.]. https://doi.org/10.31812/123456789/3744

Ramandanis, D., & Xinogalos, S. (2023). Investigating the support provided by chatbots to educational institutions and their students: A systematic literature review. *Multimodal Technologies and Interaction*, *7*(11), 103.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, *36*, 68539–68551.

Seirawan, Y., Simon, H. A., & Munakata, T. (1997). The implications of kasparov vs. Deep blue. *Communications of the ACM*, *40*(8), 21–25.

Smith, R., Castles, R., & Sondergard, E. (2025). The Engineering Ethics Coach: Integrating AI Chatbots to Strengthen Ethical Decision-Making in Engineering Economy. *2025 ASEE Annual Conference & Exposition Proceedings*, 57218. https://doi.org/10.18260/1-2--57218

Toosi, A., Bottino, A. G., Saboury, B., Siegel, E., & Rahmim, A. (2021). A brief history of AI: how to prevent another winter (a critical review). *PET Clinics*, *16*(4), 449–469.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., & Bhosale, S. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv Preprint arXiv:2307.09288*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is All you Need*.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., & Zhou, E. (2025). The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, *68*(2), 121101.

Zhang, Q., Wang, B., Huang, V. S.-J., Zhang, J., Wang, Z., Liang, H., He, C., & Zhang, W. (2024). Document parsing unveiled: Techniques, challenges, and prospects for structured information extraction. *arXiv Preprint arXiv:2410.21169*.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., & Dong, Z. (2023). A survey of large language models. *arXiv Preprint arXiv:2303.18223*, *1*(2), 1–124.

Web Resources

[1] OpenAI, "What are tokens and how to count them?," *OpenAI Help Center*. [Online]. Available: https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them.

[2] R. Lunden, "Hugging Face wants to become your artificial BFF," TechCrunch. [Online]. Available: https://techcrunch.com/2017/03/09/hugging-face-wants-to-become-your-artificial-bff/

[3] Hugging Face, "Hugging Face Hub Documentation," Hugging Face Docs. [Online]. Available: https://huggingface.co/docs/hub/en/index

[4] IBM, "What is LangChain?," IBM Think Topics. [Online]. Available: https://www.ibm.com/think/topics/langchain

[5] Contrary Research, "LangChain," Contrary Research Company Index. [Online]. Available: https://research.contrary.com/company/langchain

[6] Rasa, "Improve your AI Assistant with Rasa X," Rasa Resources. [Online]. Available: https://info.rasa.com/improve-ai-assistant-rasa-x

[7] Rasa, "Meet Hello Rasa – A faster way to get started with Rasa," Rasa Forum. [Online]. Available: https://forum.rasa.com/t/meet-hello-rasa-a-faster-way-to-get-started-with-rasa/67994

[8] IBM Watson, "Introducing Watson Assistant," Medium. [Online]. Available: https://medium.com/ibm-watson/introducing-watson-assistant-7e7dcf2d5042

[9] IBM, "IBM Unveils the Watsonx Platform to Power Next-Generation Foundation Models for Business," IBM Newsroom. [Online]. Available: https://newsroom.ibm.com/2023-05-09-IBM-Unveils-the-Watsonx-Platform-to-Power-Next-Generation-Foundation-Models-for-Business

[10] Global Venturing, "Google programs API.ai acquisition," Global Venturing Blog. [Online]. Available: https://globalventuring.com/blog/2016/09/21/google-programs-api-ai-acquisition/

[11] D. Rajkotia, "Difference between Dialogflow CX vs Dialogflow ES," DEV Community. [Online]. Available: https://dev.to/dhruv_rajkotia/difference-between-dialogflow-cx-vs-dialogflow-es-3n1k

[12] LightPDF, "PDF to Markdown," LightPDF Online Tools. [Online]. Available: https://lightpdf.com/it/pdf-to-markdown

[N] OpenAI, "Whisper," OpenAI Index. [Online]. Available: https://openai.com/it-IT/index/whisper/

Appendix

Appendix A

Q.1: Describe the main concepts of the Group Technology and indicate some advantages with respect to the functional layout.

AMPS:

Not present in the provided course materials.

ChatGPT:

Group Technology (GT): main concepts

Group Technology is a production philosophy where parts are analyzed and grouped into families based on similar processing requirements. The factory is then organized into sub-factories (cells), each dedicated to one part family and equipped with the machine types needed for that family. This organization is called cellular manufacturing. [06_Multiple_Product_Factory_Models, slide 27]

In practice, this corresponds to a cellular/group layout (as opposed to a process/functional layout). [06_Multiple_Product_Factory_Models, slide 30]

Compared with the traditional functional layout (machines grouped by function, parts moving across many departments), cellular layout organizes machines by processing cells, splits parts into families, and makes planning/scheduling simpler. [06_Multiple_Product_Factory_Models, slide 31]

Advantages vs functional layout

According to the slides, key advantages of GT/cellular layout are:

More efficient processing for a smaller, similar set of parts

→ lower setup times between part types and learning-curve gains from specialization.

[06_Multiple_Product_Factory_Models, slide 32]

Lower WIP in each sub-factory

→ parts mainly compete with parts of the same family; variability effects are reduced.

[06_Multiple_Product_Factory_Models, slide 32]

Reduced material handling and travel distances

→ routes inside a cell are typically shorter than in a functional plant.

[06_Multiple_Product_Factory_Models, slide 32]

Simplified planning and scheduling compared with chaotic inter-department flows in functional layouts.

[06_Multiple_Product_Factory_Models, slide 31]

Gemini

Group Technology (GT) is a manufacturing philosophy that involves analyzing processing operations to identify similarities between parts. Based on these common characteristics, parts are grouped into families for production purposes.

This analysis leads to the implementation of Cellular Manufacturing, where the factory is organized into "sub-factories" or cells. Each cell contains a group of machines specifically dedicated to producing a particular job type family.

Advantages over Functional Layout

A Functional Layout organizes the plant according to the function of the machines (e.g., all lathes in one area, all milling machines in another), which often leads to chaotic movement of parts. In contrast, a Cellular Layout offers several advantages:

More Efficient Processing: Specializing in a smaller set of similar parts reduces setup times and benefits from learning-curve effects.

Reduced Work-in-Process (WIP): Parts only encounter others from the same technology group, and there is often a reduction in the service time squared coefficient of variation (C_s^2), leading to lower WIP levels.

Lower Material Handling Requirements: Distances jobs must travel between machines within a cell are significantly shorter than the long routes required in a traditional functional setting.

Simplification of Planning: Organizing the plant by cells simplifies the overall planning and scheduling of production.

Performance Improvements through Specialization: The primary gains in performance measures (such as reduced cycle times) come from specialization-related improvements, including processing simplifications and improved quality.

Q2. What is the bottleneck rate of a line?

AMPS:

Bottleneck rate of a line

The bottleneck rate is the maximum sustainable long term throughput of a production line, determined by the slowest workstation. It corresponds to the station with the largest cycle time (or lowest processing rate).

Overview

In a multi stage production system, each workstation has a processing time. The stage with the longest cycle time limits the overall output rate of the line. This limiting stage is called the bottleneck, and its processing rate is the bottleneck rate.

Key points from the course materials

In flow lines, the production rate is determined by the workstation with the smallest capacity.

The bottleneck rate is defined as:

“the reciprocal of the largest effective processing time among all stages.”

More intuitively:

If a station has cycle time c, its processing rate is 1/c.

The bottleneck rate is the minimum among all processing rates (because it is the slowest).

This is shown in the multi stage systems slides where the bottleneck station governs the line’s performance.

Pitfalls / Assumptions

The bottleneck rate assumes steady state conditions.

Variability may change waiting times and WIP, but the rate is still determined by the slowest effective processing time.

The bottleneck may shift if processing times change or if variability alters effective station performance.

Where to go further

See the sections on multi stage flow lines and system performance indicators for examples of identifying bottlenecks in practice.

Citations

[Slides: 07_Multiple-Stage Factory Models_with_solutions.pdf]

ChatGPT:

The bottleneck rate of a line (r_b) is the throughput rate of the bottleneck workstation (i.e., the station with the lowest capacity).

So it is also the maximum throughput the whole line can achieve.

02_Introduction_Factory_Models

In formula terms:

r_b=(min⁡)┬k { 〖"station capacity" 〗 _k}

and this value caps the line throughput. (Example in slides: r_b=0.4pcs/h when WS2 is the bottleneck.)

02_Introduction_Factory_Models

Gemini:

The bottleneck rate (r_b) is a fundamental parameter of a production line, defined and calculated as follows according to the provided materials:

Definition: The bottleneck rate is the rate (jobs per unit of time) of the workstation having the

highest long-term utilization.

Capacity Limit: It represents the capacity of the bottleneck station and determines the maximum sustainable throughput for the entire production line.

Workload Calculation: To identify the bottleneck in a multi-product system, the workload (WL_k) for each workstation k must be computed by summing the arrival rate of each product type ($\lambda_{i,k}$) multiplied by its mean processing time ($E[T_s(i,k)]$): $WL_k = \sum_{i=1}^{m} \lambda_{i,k} E[T_s(i,k)]$.

Utilization Factor: The utilization (u_k) is then found by dividing the workload by the number of identical processors (c_k) at the station: $u_k = WL_k / c_k$.

Bottleneck Identification: The workstation with the highest utilization factor is identified as the bottleneck.

Steady-State Requirement: A factory can only achieve steady-state (and thus avoid infinite queues) if the utilization factors of all workstations, including the bottleneck, are strictly less than 1.0.

Q3. Which are the characteristic of Simulation in the context of Industry 4.0?

AMPS:

Characteristics of Simulation in Industry 4.0

Overview

In the context of Industry 4.0, Simulation is presented as a key enabling technology that supports decision making, optimization, and digitalization of production systems.

Key Characteristics (from the slides)

According to the Industry 4.0 lecture:

Digital representation of the physical world

Simulation is used to build a virtual model of real production systems.

It allows testing and comparing system behaviors before implementing changes in the real factory.

(Slides describe Simulation as part of the virtual–physical integration.)

Interconnection with data / Cyber Physical Systems

Simulation in Industry 4.0 can use real time data coming from sensors and smart devices.

This enables dynamic, data driven models that can mirror the physical system.

Support for decision making and optimization

The slides highlight Simulation as a tool for:

planning,

scheduling,

layout analysis,

forecasting system performance.

It reduces uncertainty and risk when evaluating alternatives.

Integration into the Digital Twin concept

Simulation is shown as a fundamental component of the Digital Twin, representing the virtual counterpart of the physical asset.

Experimentation without physical intervention

Simulation allows evaluating “what if” scenarios without interrupting production.

Connection with Discrete Event Simulation (DES)

The course includes a full module on DES, reinforcing its role in analyzing:

flow lines,

variability,

queues,

performance indicators.

Pitfalls / Assumptions in the slides

A simulation model is only as accurate as the data and assumptions behind it.

Building a good model requires clear system boundaries and validated inputs.

Simulation does not optimize by itself; experimentation and analysis are required.

Where to go further

Check the lecture on Discrete Event Simulations and the core Industry 4.0 slides for examples of simulation in smart factories.

Citations

[Slides: 11_Discrete_event_simulations.pdf; 12_Industry_4.0.pdf]

ChatGPT:

In Industry 4.0, simulation is characterized as:

Model-based experimentation: it is the process of creating a model of a real system and running experiments on it to understand behavior and evaluate alternative operating strategies.

[10_Industry_4.0_and_5.0, slide 4]

10_Industry_4.0_and_5.0

Decision-support tool for many tests (“what-if”): simulation is used to test multiple scenarios before acting on the real system.

[10_Industry_4.0_and_5.0, slide 4]

10_Industry_4.0_and_5.0

Strong link with Digital Twin: in I4.0, simulation evolves toward high-fidelity digital representations of physical systems.

[10_Industry_4.0_and_5.0, slide 5]

10_Industry_4.0_and_5.0

Dynamic and data-updated behavior (Digital Twin): the model is “living”, continuously updated with operational information, and can forecast future behavior of the physical counterpart.

[10_Industry_4.0_and_5.0, slide 5]

10_Industry_4.0_and_5.0

Maturity levels: the lesson distinguishes digital model → digital shadow → digital twin, with increasing integration between physical and virtual system.

[10_Industry_4.0_and_5.0, slide 5]

Gemini:

In the context of Industry 4.0, simulation is a key technology used to analyze and optimize production systems. According to the provided materials, its characteristics and roles are:

Decision Support Tool: Simulation is used as a usable tool for decision-making purposes by providing analytical results that closely match real-world parameters.

Performance Comparison: It allows for the comparison of critical parameters between analytical models and simulated environments to validate modeling approaches.

Digital Twin Foundation: Simulation is a fundamental component of the "Digital Twin" concept, which is a key enabling technology of Industry 4.0.

Virtual Modeling: It involves creating virtual models of physical systems to simulate their behavior before physical implementation.

Discrete Event Simulation (DES): It is specifically characterized as "Discrete Event Simulation," where the system changes only at discrete points in time when specific events occur.

Evaluation of Scenarios: It is used to evaluate "what-if" scenarios, such as comparing the performance of traditional layouts versus cellular layouts.

Handling Complexity: Simulation handles the complexity of multiple product types, re-entrant flows, and varying processing times that are typical in modern smart factories.

Q.4: Which of the following definitions best describes the concept of maintenance in production systems?

a) replacement of a component after a failure

b) application of predictive algorithms to identify the causes of failures

c) set of activities aimed at ensuring that each element of the system can perform its function correctly (X)

d) signing of a contract with a company specializing in the repair of systems

AMPS

Answer (English)

To choose the correct Kendall notation, we use the structure A/S/c/K, where:

A = inter arrival distribution

S = service time distribution

c = number of parallel machines

K = system capacity (∞ if unlimited)

From the problem:

Inter arrival time: Normal distribution → N

Processing time: Exponential distribution → M

Machines in parallel: 3

Queue: unlimited → ∞

Therefore, the correct notation must be:

N/M/3/∞

Among the options, this matches:

→ e) G/M/3/∞ ?

But we must check the slides.

Check in course materials

In the slides, the Kendall notation is defined explicitly as:

D: deterministic

M: exponential

G: generic (any distribution)

There is no symbol “N” for “Normal distribution” in the course slides.

According to the course:

If the arrival distribution is not D nor M, it must be labeled G (generic).

A Normal distribution falls under G.

This is shown in:

“Arrival process (D, M and G)” and “Processing time distributions (D, M and G)”

[Slides: 05_Single_Workstation_Analysis_with_solutions.pdf]

Thus:

Inter arrival = Normal → G

Service time = Exponential → M

c = 3

K = ∞

Correct notation in this course:

✅ G/M/3/∞

The correct answer is:

➡️ e) G/M/3/∞

Citations

[Slides: 05_Single_Workstation_Analysis_with_solutions.pdf]

ChatGPT:

The correct choice is e) G/M/3/∞.

Why:

Arrival process = normal distribution → not Markovian, so in Kendall notation we use G(general).

Processing time = exponential → M.

Three parallel machines → 3servers.

Unlimited queue capacity → ∞system capacity.

So the workstation is G/M/3/∞.

Gemini:

I am sorry, but I cannot assist with that. I am unable to see or process the quiz you are referring to as there is no image attached to your message.

Please upload the image or type out the questions, and I will be happy to provide the correct answers and explanations based on the provided course materials, including the specific slide and page citations.

Q5. At which level of the hierarchy are the data concerning the realt time monitoring of the production machines used to take decisions?

a) strategical

b) tactical

c) operational

AMPS:

In the course materials, the hierarchy of decision levels is presented clearly in Information Systems and Industry 4.0 slides.

Real time monitoring of production machines refers to short term, immediate decisions based on data from the shop floor (e.g., current status, alarms, machine conditions).

According to the slides, operational level deals with:

real time control of machines,

monitoring,

execution of production activities,

short term decisions based on current system status.

This matches exactly the use of real time machine monitoring data.

Correct answer: → c) operational

Citations

[Slides: 10_Information_systems.pdf]

ChatGPT:

The correct level is c) operational.

In Anthony's hierarchy, the operational level is linked to day-to-day execution, and it is the level where information is mainly exact, analytical, and real-time.

[09_Information_systems, slides 13 and 15]

Gemini:

Based on the provided materials regarding information systems and factory modeling, the correct answer is c) operational.

Here is the detailed explanation with citations:

Operational Level: Real-time monitoring data from production machines is used at the operational level to support day-to-day and minute-to-minute decisions.

Data Usage: At this level, information systems (such as MES - Manufacturing Execution Systems) collect real-time data to manage work orders, track progress, and monitor machine status.

Comparison with Other Levels:

Tactical Level: This level typically involves medium-term planning, such as monthly production scheduling and resource allocation, rather than real-time machine monitoring.

Strategic Level: This level focuses on long-term goals and high-level corporate decisions, using aggregated data rather than real-time machine signals.

Industry 4.0 Context: The integration of real-time monitoring through Cyber-Physical Systems (CPS) specifically targets the operational efficiency of the shop floor.

Would you like me to explain the specific role of the Manufacturing Execution System (MES) in processing this real-time data as described in the documents?

Q.6: A workstation has a processing rate of 6 j/h. The inter arrival time of jobs to the workstation is 20 min. Which is the utilization of the workstation?

AMPS

To compute the utilization ρ of a workstation, the slides define:

$\rho=\lambda/\mu$

where:

λ = arrival rate

μ = processing rate

[Slides: 05_Single_Workstation_Analysis_with_solutions.pdf]

Given data

Processing rate: 6 jobs/hour → $\mu = 6$ j/h

Inter arrival time: 20 minutes

Convert inter arrival time to arrival rate:

$20" min"=1/3 " h"$

$\lambda=1/"inter\text{-}arrival\ time" =1/(1/3)=3" jobs/hour"$

Utilization

$\rho=\lambda/\mu=3/6=0.5=50\%$

✅ Correct answer: c) 50%

Citation

[Slides: 05_Single_Workstation_Analysis_with_solutions.pdf]

ChatGPT

Utilization is:

$u=\lambda/\mu$

Inter-arrival time $=20min \Rightarrow \lambda=60/20=3jobs/h$

Processing rate $\mu=6jobs/h$

$u=3/6=0.5=50\%$

✅ Correct answer: c) 50%.

(Formula u=λ/μfrom M/M/1 performance summary.)

Gemini

The correct answer is c) 50%.

Step-by-Step Calculation

To find the utilization (u), we need to compare the arrival rate (λ) with the processing capacity (μ):

Find the Arrival Rate (λ):

The inter-arrival time ($E[T_a]$) is 20 min.

Since there are 60 minutes in an hour, the arrival rate is $60 / 20 = \mathbf{3~j/h}$.

Identify the Processing Rate (μ or $1/E[T_s]$):

The problem states the processing rate is 6 j/h.

Calculate Utilization (u):

The formula for utilization is $u = \lambda / \mu$ (or $u = \lambda \cdot E[T_s]$).

$u = 3 / 6 = \mathbf{0.5}$.

Converted to a percentage, $0.5 \times 100 = \mathbf{50\%}$.

Q7. In a system the mean arrival rate is λ = 6 jobs/hours, and it is exponentially distributed. The mean service times is E[Ts] = 0.5 hours, with a squared coefficients of variation Cs2 = 0.8. Which is the number of parallel machines needed to get a utilization lower or equal to 0.9? Considering the obtained number of parallel machines, compute the CT, the WIP and the TH of the system.

AMPS:

1. Number of parallel machines (m)

Data:

Arrival rate: λ=6" jobs/h" (exponential ⇒ C_a^2=1)

Mean service time (per machine): E[T_s]=0.5" h"

⇒ service rate per machine: μ=1/E[T_s]=2" jobs/h"

Service time SCV: C_s^2=0.8

For a workstation with mparallel machines, utilization is

u=λ/mμ

We want u≤0.9:

6/(m·2)≤0.9⇒6/2m≤0.9⇒3/m≤0.9⇒m≥3.33

So we need m=4machines.

[Slides: 05_Single_Workstation_Analysis_with_solutions.pdf]

---

2. CT (mean cycle time)

Approximate the workstation as a single “aggregated” server with total rate

$\mu_{tot} = m\mu = 4 \cdot 2 = 8$ jobs/h

So

Effective mean service time: $E[T_s^{eff}] = 1/\mu_{tot} = 0.125$ h

Utilization: $u = \lambda/\mu_{tot} = 6/8 = 0.75$

Use Kingman’s VUT approximation for a G/G/1station:

$CT_q \approx (C_a^2 + C_s^2)/2 \cdot u/(1-u) \cdot E[T_s^{eff}]$

$(C_a^2 + C_s^2)/2 = (1+0.8)/2 = 0.9$

$u/(1-u) = 0.75/0.25 = 3$

$CT_q \approx 0.9 \cdot 3 \cdot 0.125 = 0.3375$ h

Total cycle time:

$CT = CT_q + E[T_s^{eff}] = 0.3375 + 0.125 = 0.4625$ h ≈ 0.46 h

[Slides: 06_Process_variability_with_solutions.pdf]

---

3. TH and WIP (Little’s Law)

For a stable system, throughput equals the arrival rate:

$TH = \lambda = 6$ jobs/h

Using Little’s Law $WIP = TH \cdot CT$:

$WIP = 6 \cdot 0.4625 \approx 2.78$ jobs

Results

Number of machines: $m = 4$

Utilization: $u = 0.75$

CT ≈ 0.46 h (≈ 27.8 min)

TH ≈ 6 jobs/h

WIP ≈ 2.8 jobs

These calculations use the utilization definition, Kingman's VUT formula, and Little's Law as presented in the course.

[Slides: 05_Single_Workstation_Analysis_with_solutions.pdf; 06_Process_variability_with_solutions.pdf]

ChatGPT

Great exercise. Treat it as a G/G/c workstation with:

λ=6jobs/h

E[T_s]=0.5h

C_a^2=1(exponential arrivals)

C_s^2=0.8

The multiserver approximation used in class is the Kingman extension (Allen–Cunneen/Hall style). [03_Single_Workstation_Analysis, slides 83–84]

03_Single_Workstation_Analysis

---

1) Number of parallel machines for u≤0.9

For cidentical machines:

u=(λE[T_s])/c

[06_Multiple_Product_Factory_Models, slide 11]

06_Multiple_Product_Factory_Mod…

λE[T_s]=6·0.5=3⇒3/c≤0.9⇒c≥3.33

So minimum integer is:

□(c=4)

(check: u=3/4=0.75≤0.9).

---

2) CT (cycle time), WIP, TH with c=4

For G/G/c approximation:

CT_q≈(C_a^2+C_s^2)/2 " " u^(√(2(c+1))-1)/(c(1-u)) " " E[T_s]

CT=CT_q+E[T_s]

[03_Single_Workstation_Analysis, slides 84 and 77]

Numbers:

$(C_a^2+C_s^2)/2=(1+0.8)/2=0.9$

$u=0.75,"\ "\ c=4$

$\sqrt{(2(c+1))}-1=\sqrt{10}-1\approx2.162$

$u^{2.162}\approx0.537$

$c(1-u)=4\cdot0.25=1$

So:

$CT_q\approx0.9\cdot0.537\cdot0.5\approx0.242"\ h"$

$CT\approx0.242+0.5=0.742"\ h"$

Throughput in steady state (infinite-buffer, stable system) is:

$TH\approx\lambda=6"\ jobs/h"$

[03_Single_Workstation_Analysis, slide 36 summary: $TH=\lambda$]

03_Single_Workstation_Analysis

Then Little's Law:

$WIP=TH\cdot CT\approx6\cdot0.742=4.45"\ jobs"$

[05_Multiple-Stage Factory Models, examples using $WIP=\lambda\cdot CT$]

05_Multiple-Stage Factory Model…

Final results

Parallel machines needed: □4

Utilization (with 4 machines): □(u=0.75)

Cycle time: □($CT\approx0.742"\ h\ "$ ($\approx44.5"\ min"$))

Throughput: □($TH\approx6"\ jobs/h"$)

WIP: □($WIP\approx4.45"\ jobs"$)

Gemini

To solve this problem, we will first determine the number of parallel machines needed and then calculate the performance measures (CT, $WIP, and TH) using the formulas provided in the course materials.

1. Number of Parallel Machines (c)

The utilization (u) of a workstation with c parallel machines is calculated as:

$u = \frac{\lambda \cdot E[T_s]}{c}$

Given:

$\lambda = 6$ jobs/hour

$E[T_s] = 0.5$ hours

We want $u \leq 0.9$:

$$\frac{6 \cdot 0.5}{c} \leq 0.9 \implies \frac{3}{c} \leq 0.9 \implies c \geq \frac{3}{0.9} \approx 3.33$$

Since the number of machines must be an integer, we need $c = 4$ parallel machines to achieve a utilization lower than or equal to 0.9.

With $c = 4$, the actual utilization is:

$$u = \frac{3}{4} = 0.75$$

---

2. Throughput (TH)

In a stable system ($u < 1$), the throughput is equal to the arrival rate:

$$TH = \lambda = 6 \text{ jobs/hour}$$

---

3. Cycle Time (CT)

The total cycle time is the sum of the expected waiting time in the queue (CT_q) and the mean service time ($E[T_s]$):

$$CT = CT_q + E[T_s]$$

Since the inter-arrival times are exponentially distributed, the squared coefficient of variation for arrivals is $C_a^2 = 1$. We use the CT_q formula for multi-server workstations:

$$CT_q \approx \left(\frac{C_a^2 + C_s^2}{2} \right) \left(\frac{u^{\sqrt{2(c+1)}-1}}{c(1-u)} \right) E[T_s]$$

Plugging in the values ($C_a^2 = 1$, $C_s^2 = 0.8$, $u = 0.75$, $c = 4$, $E[T_s] = 0.5$):

$$V = \frac{1 + 0.8}{2} = 0.9$$

$$\text{Exponent} = \sqrt{2(4+1)} - 1 = \sqrt{10} - 1 \approx 2.162$$

$$CT_q \approx 0.9 \cdot \frac{0.75^{2.162}}{4(1 - 0.75)} \cdot 0.5 \approx 0.9 \cdot \frac{0.5368}{1} \cdot 0.5 \approx 0.2416 \text{ hours}$$

Total Cycle Time:

$$CT = 0.2416 + 0.5 = \mathbf{0.7416 \text{ hours}}$$

---

4. Work-In-Process (WIP)

According to Little's Law, the average number of jobs in the system is:

$$WIP = TH \cdot CT$$

$$WIP = 6 \cdot 0.7416 = \mathbf{4.4496 \text{ jobs}}$$

Appendix B

Q.1: Describe the main concepts of the Group Technology and indicate some advantages with respect to the functional layout.

Below is a concise, course materials–compliant explanation of Group Technology (GT) and the main advantages over a functional layout, using only content that appears in the Multiple Product / Cellular Manufacturing materials.

Overview

In the course, Group Technology appears through the concept of Cellular Manufacturing, where products with similar processing requirements are grouped and produced in dedicated cells rather than in a traditional functional layout.

This redesign aims to reduce flow complexity and improve performance in multiple product environments.

Key Concepts of Group Technology (GT)

From the Cellular Manufacturing video lecture:

Families of products: items are grouped when they share similar process plans or routing sequences.

Manufacturing cells: machines are physically arranged in “cells” dedicated to one product family.

Reduced routing variability: similar products follow similar, shorter internal flows.

Simplified material handling: the movement inside a cell replaces long cross department transfers typical of functional layouts.

These ideas appear explicitly in the multiple product / cellular manufacturing lecture where the structure of product families and cells is introduced.

Advantages vs. Functional Layout

Based on the same materials:

Reduced flow distances and transport times

Cells eliminate long inter department travel.

Lower WIP and shorter Throughput Time (TH)

More focused flows reduce queueing and variability effects typical of multi department functional layouts.

Simpler scheduling and control

Each cell manages a limited and homogeneous product family.

Improved visibility and coordination

Workers handle complete flows inside a cell rather than fragmented operations across departments.

Higher responsiveness to variability

Dedicated cells reduce routing complexity, enabling more stable performance.

Pitfalls / Assumptions

GT is effective only when clear product families exist.

Machine duplication may be needed to create fully independent cells.

Where to Go Next

Review the numerical examples in Multiple Product and Cellular Manufacturing materials to see how GT changes flow representation and performance.

Sources

[Slides: 06_Multiple_Product_Factory_Models_with_solutions.md]

[Videolectures: 6.3_Cellular_Manufacturing.md]

Q2. What is the bottleneck rate of a line?

Definition (from course materials)

In the course, the bottleneck rate is defined in the context of multi stage lines and appears explicitly in the “Critical Parameters” video and in the multiple stage models.

Overview

The bottleneck rate, often denoted rb, represents the maximum sustainable output rate of the entire production line, determined by the slowest (i.e., least productive) station.

Key Points from the Course

A production line’s output rate cannot exceed the rate of its bottleneck station.

This is stated in the “Critical Parameters” lecture when discussing how the system throughput is constrained by the station with the lowest capacity.

In a serial line (as discussed in Multiple-Stage Factory Models), the station with the longest processing time or lowest capacity defines the overall line capability.

The bottleneck rate rb is therefore:

The maximum throughput rate the line can achieve.

Equal to the minimum capacity among all stations.

Although the slides do not present an explicit closed form formula labelled “rb”, the concept is the same one applied when identifying the “slowest stage” limiting the line.

Pitfalls / Assumptions

The bottleneck rate is defined under steady state conditions.

Variability and blocking/starving effects may make the effective throughput lower than the bottleneck rate.

Where to find more in the course

Serial lines and throughput limits → Multiple-Stage Factory Models

Definition of critical constraints like rb and CT → Critical Parameters video lecture.

Citations

[Slides: 05_Multiple-Stage Factory Models_with_solutions.md]

[Videolectures: 2.3_Critical Parameters.md]

Q3. Which are the characteristic of Simulation in the context of Industry 4.0?

Below is the answer only using the official course materials, as required.

Overview

In Industry 4.0, the course presents Simulation as one of the enabling digital technologies that support advanced decision making, process analysis, and virtual validation of production systems.

The relevant content appears in the slide deck 09_Industry_4.0.md.

Characteristics of Simulation in the context of Industry 4.0

From the Industry 4.0 slides, Simulation is described as a key tool used to create digital models of production systems and analyze their behaviour before real implementation.

The main characteristics highlighted are:

1. Digital representation of the system

Simulation creates a virtual model of machines, flows, and processes to test configurations without interrupting real operations.

(Reference: “Digital Models”, “Virtual factory” concepts in 09_Industry_4.0.md)

2. Support for decision-making

Simulation allows the evaluation of alternatives, “what if” scenarios, layout changes, and system behaviors under different conditions.

This directly links to the course emphasis on testing system performance without physical experimentation.

3. Integration with data

In the slide deck, Simulation is described as part of the Industry 4.0 ecosystem interconnected with production data, enabling data driven models and more accurate analyses.

4. Risk reduction and validation

By experimenting virtually, firms reduce costs and risks associated with trial and error on the shop floor.

5. Enabler of the “Digital Twin”

Simulation is explicitly connected with the Digital Twin, a digital replica of the physical factory, updated with real time data.

Pitfalls / Assumptions

The slides emphasize Simulation as a supportive technology, not a standalone solution.

Accuracy depends on the quality and availability of data.

Where to go next

See the “Simulation”, “Digital Twin”, and “Virtual factory” sections embedded in the Industry 4.0 slide deck.

Citations

[Slides: 09_Industry_4.0.md]

[Videolectures: 3.1_Variability, Model Simulation.md]

Q.4: Which of the following definitions best describes the concept of maintenance in production systems?

a) replacement of a component after a failure

b) application of predictive algorithms to identify the causes of failures

c) set of activities aimed at ensuring that each element of the system can perform its function correctly (X)

d) signing of a contract with a company specializing in the repair of systems

The correct option is c).

✓ Best definition according to the course materials

The course treats maintenance in terms of machine breakdown, repair, and the resulting availability A = m / (m + d).

From these materials, maintenance corresponds to all activities that keep a machine able to perform its function and restore it after failures.

This aligns only with:

👉 c) set of activities aimed at ensuring that each element of the system can perform its function correctly

Why the others are incorrect (from course perspective)

a) refers only to corrective replacement, but maintenance includes both preventive and corrective actions.

b) addresses predictive methods, which are part of maintenance strategies but not the definition.

d) refers to an administrative or contractual aspect, not the technical meaning.

Citations

Machine breakdown, repair and availability:

[Slides: 04_Process_variability_with_solutions.md]

[Videolectures: 4.2_Machine Breakdown and examples.md]

Q5. At which level of the hierarchy are the data concerning the realt time monitoring of the production machines used to take decisions?

a) strategical

b) tactical

c) operational

The correct answer is c) operational.

Explanation (based strictly on course materials)

In the Information Systems slide deck, the hierarchy of decision levels is defined as:

Strategical level → long term, structural decisions

Tactical level → medium term planning and resource allocation

Operational level → short term decisions based on real time data from the shop floor

In 08_Information_systems.md, the role of real time shop-floor data (from sensors, PLCs, MES) is explicitly associated with operational control, where immediate decisions are taken to manage production execution.

Thus, real-time monitoring of machines → operational level.

Citations

[Slides: 08_Information_systems.md]

[Videolectures: 0-Course_Introduction.md]

Q.6: A workstation has a processing rate of 6 j/h. The inter arrival time of jobs to the workstation is 20 min. Which is the utilization of the workstation?

To compute the utilization ρ of the workstation, we use the definitions provided in the Single Workstation Analysis slides.

A workstation with:

Processing rate $\mu = 6$ jobs/hour

Interarrival time = 20 min = 1/3 hour

$\Rightarrow$ Arrival rate $\lambda = 1 / (1/3) = 3$ jobs/hour

Utilization formula

From the course:

$\rho = \lambda / \mu$

(see definition of utilization in the M/M/1 model)

[Slides: 03_Single_Workstation_Analysis_with_solutions.md]

[Videolectures: 3.3_Queuing System, Kendall Notion,MM1 formula.md]

Compute it

$\rho = 3 / 6 = 0.5$

✅ Utilization = 0.5 (50%)

Citations

[Slides: 03_Single_Workstation_Analysis_with_solutions.md]

[Videolectures: 3.3_Queuing System, Kendall Notion,MM1 formula.md]

Q7. In a system the mean arrival rate is $\lambda = 6$ jobs/hours, and it is exponentially distributed. The mean service times is E[Ts] = 0.5 hours, with a squared coefficients of variation Cs2 = 0.8. Which is the number of parallel machines needed to get a utilization lower or equal to 0.9? Considering the obtained number of parallel machines, compute the CT, the WIP and the TH of the system.

1. Find the number of parallel machines c

Given:

Arrival rate: λ=6jobs/h (Poisson ⇒ exponential interarrival)

Mean service time: E[T_s]=0.5h ⇒ service rate per machine μ=1/E[T_s]=2jobs/h

Squared coefficient of variation of service times: C_s^2=0.8

For a system with cidentical parallel machines:

$\rho=\lambda/c\mu$

Requirement: $\rho \leq 0.9$:

$6/(c \cdot 2) \leq 0.9 \Rightarrow 6/2c \leq 0.9 \Rightarrow 3/c \leq 0.9 \Rightarrow c \geq 3/0.9 = 3.3\bar{}$

⇒ Minimum integer c=4.

Then the actual utilization is:

ρ=6/(4·2)=6/8=0.75

---

2. Compute CT, WIP, TH

We use the G/G/c approximation from the slides:

First compute CT_q (M/M/c)with Erlang C (since arrivals are Poisson)

Traffic offered: a=λ/μ=6/2=3

c=4, ρ=a/c=0.75

P_0=[∑_(n=0)^3▒a^n/n!ⓜ+a^4/(4!(1-ρ))]^(-1)=[1ⓜ+3ⓜ+4.5ⓜ+4.5ⓜ+13.5]^(-1)=1/26.5≈0.0377

P_w=a^4/(4!(1-ρ)) P_0=13.5×0.0377≈0.51

CT_q (M/M/c)=P_w/(cμ-λ)=0.51/(8-6)≈0.255" h"

Adjust to G/G/c using the variability factor

For exponential arrivals: C_a^2=1

V=(C_a^2+C_s^2)/2=(1+0.8)/2=0.9

CT_q (G/G/c)≈V·CT_q (M/M/c)≈0.9×0.255≈0.229" h"

Add service time to get total cycle time:

CT=CT_q+E[T_s]≈0.229+0.5=0.729" h"

Throughput for a stable open system:

$TH \approx \lambda = 6$" jobs/h"

Use Little's Law to get WIP:

$WIP = TH \cdot CT \approx 6 \times 0.729 \approx 4.37$" jobs"

Final results

Number of machines: $c=4$

Utilization: $\rho = 0.75 \leq 0.9$

CT ≈ 0.73 h

TH ≈ 6 jobs/h

WIP ≈ 4.4 jobs

If you want, next we can repeat the calculations numerically in minutes to match how your exam usually expresses results.

[Slides: 03_Single_Workstation_Analysis_with_solutions.md; 04_Process_variability_with_solutions.md]

[Videolectures: 3.3_Queuing System, Kendall Notion,MM1 formula.md; 3.6_MG1, GG1, GGc.md]

Appendix C

Procedure for Recreating the AMPS Chatbot

1 Preparation of the Knowledge Base

Prepare all the files that will compose the chatbot knowledge base. The required materials are:

- Lecture slides converted into **Markdown (.md)** format
- **Video lecture transcriptions** in Markdown format
-
- **Bridge Notes** in Markdown format
- **Glossary** in Markdown format

Create a folder in **OneDrive or SharePoint** and upload all the files into that folder.

When organizing the folder, follow these rules:

- All files must be placed **inside the same folder**
- **No subfolders** must be created
- The **name of the folder is not relevant**

After uploading the files, obtain the **shareable link** of the folder:

1. Select the folder.
2. Click **Share**.
3. Copy the generated sharing link.

2 Creating the Chatbot in Copilot Studio

Open the **Microsoft Copilot** website.

Perform the following steps:

1. Navigate to the **Agents** section.
2. Click **New Agent**.

3 Agent Configuration

In the chatbot creation interface, complete the following fields:

Title

Insert the name of the chatbot.
Example:

- AMPS Chatbot

Description

Insert a short textual description of the chatbot.

Image (optional)

Upload an image to be used as the chatbot cover.

4 Inserting the Instruction Prompt

In the **Instructions** section:

- Copy the complete prompt used to configure the chatbot.
- Paste the prompt into the **Instructions** field.

5 Connecting the Knowledge Base

Scroll to the **Knowledge** section and configure the knowledge base using the following steps:

1. Select **Add specific websites**.
2. Paste the previously copied **OneDrive or SharePoint folder link**.
3. Activate the option **Use only specified sources**.

6 Creating the Chatbot

Complete the creation process:

- Click the **Create** button located in the **upper-right corner** of the interface.

7 Testing the Chatbot

During the creation process, use the **preview window on the right side of the interface** to test the chatbot.

Insert sample questions and verify the generated responses.