1-31-2024

An SOA-Based Approach of Adaptive E-Tutoring Systems

Parth Hetalkumar Mistry

University of Windsor

An SOA-Based Approach of Adaptive E-Tutoring Systems

By

Parth Hetalkumar Mistry

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2024

An SOA-Based Approach of Adaptive E-Tutoring Systems

by

Parth Hetalkumar Mistry

APPROVED BY:

L. Oriet
Department of Mechanical, Automotive and Materials Engineering

O. Syrotkina
School of Computer Science

X. Yuan, Advisor
School of Computer Science

January 16, 2024

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The educational technology landscape continually evolves, and e-tutoring systems are pivotal in modern pedagogy. Traditional e-tutoring methods often need help with adaptability and user-friendliness across various devices and platforms. To address these challenges, this research introduces a novel approach that leverages service-oriented architecture (SOA) principles, enhancing scalability and flexibility.

The SOA configuration streamlines communication between system components, optimizing question delivery and response evaluation. Additionally, the research contributes adaptive interfaces that intelligently engage users based on their device configurations and preferences, offering facial, vocal, or textual interactions. These interfaces ensure a consistent and tailored learning experience across PCs, laptops, and mobile devices.

The study also considers critical success factors like User-Friendly Design and Technical Competence. This research presents a comprehensive solution to enhance e-tutoring systems for modern, adaptive, and engaging learning environments.

DEDICATION

I dedicate this thesis to my beloved parents, who are the backbone of my family. My father's incredible love and support, selfless hard work, and unwavering dedication have inspired me. Similarly, my mother's nurturing guidance, which has allowed me to spread my wings and pursue my dreams, has been instrumental in my academic journey.

I extend this dedication to my grandfather and grandmother, whose trust in me and support during my most trying times have significantly impacted my academic path. The road would have been much more challenging without their wisdom and encouragement.

To my dear uncle, aunt, and younger brother, the beacons of support and encouragement. Their guidance and belief in my potential have provided a solid foundation, while my brother's companionship has added a unique and cherished dimension to my academic journey.

To my friends, your devoted friendship, late-night study sessions, and occasional moments of much-needed laughter have made this challenging journey more bearable.

To my esteemed professors and mentors, your invaluable guidance and wisdom have profoundly influenced my intellectual growth. Your enthusiasm for education has been a constant source of inspiration.

This thesis also serves as a tribute to the countless individuals who stand to benefit from the research presented here. I aspire that this work contributes, in some small way, to advancing knowledge and improving our world.

ACKNOWLEDGEMENTS

I want to sincerely thank the thesis committee members, whose expertise and insights significantly contributed to developing and refining this research project.

Special thanks to Dr. Leo Oriet, whose valuable perspective brought this thesis a unique, enriching dimension. His thoughtful feedback and guidance have been pivotal in ensuring the research findings' robustness and applicability.

I am also deeply grateful to Dr. Olena Syrotkina for her expertise, which greatly enhanced the computational aspects of my work. Her constructive input and keen understanding of technological nuances have been instrumental in elevating the overall quality of this thesis.

I also appreciate the supportive research environment Dr. Yuan created and the opportunities for me to collaborate on projects, attend conferences, and present my work. These experiences have been invaluable in my academic development.

Dr. Yuan provided insightful feedback and constructive criticism, encouraging me to think critically and approach problems creatively. His dedication to the pursuit of knowledge and commitment to academic excellence has been instrumental in shaping the quality of this thesis.

I am thankful for Dr. Yuan's patience and willingness to invest time in discussions and consultations, which greatly enriched the depth and scope of my research. His mentorship extended beyond academics, as he consistently demonstrated care and concern for my personal and professional growth.

Once again, I extend my heartfelt thanks to all the committee members for their outstanding guidance, mentorship, and unwavering support, without which this thesis would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ITS	Intelligent Tutoring Systems
E-Tutoring	Electronic Tutoring
SOA	Service Oriented Architecture
CD	Compact Disk
UI	User Interface
GUI	Graphical User Interface
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
AI	Artificial Intelligence
CITS	Conversational Intelligent Tutoring Systems
LLM	Large Language Models
ECA	Embodied Conversational Agents
POMDP	Partially Observable Markov Decision Process
BSH	Belief State History
DWT	Discrete Wavelet Transform
NCP	Number of Change Points
TTS	Text to Speech
STT	Speech to Text
UI	User Interface

CHAPTER 1

Introduction

In the modern world, e-learning, often called online learning, has seamlessly integrated into our daily lives, encompassing many participants, including educators, learners, and a vast array of online platforms and applications. The digital tools employed within this educational sphere are instrumental in fostering a substantial transformation across various facets of human society. The ongoing evolution in the education landscape has been significantly influenced by the profound impact of the COVID-19 pandemic, with the return to conventional, in-person teaching in schools and colleges still fraught with uncertainty. The multi-billion-dollar online/digital learning industry exhibits major potential and tremendous growth in its context.

The merits of online learning extend to its unparalleled accessibility, affordability, and flexible scheduling, making it a robust contender in contemporary education. Notably, online learning is often seen as a panacea during times of crisis [22]. In the year this thesis was written, the world has witnessed several phases and intensities of lockdowns. The adoption of online modalities and a surge in research activities in the same domain characterize this shift.

Despite its many advantages, online learning presents its formidable challenges. These obstacles include issues related to internet connectivity, download errors, audio and video problems, subpar course content, and a lack of motivation in the absence of a physical teacher [22].

Among the digital tools transforming the landscape, Virtual, Augmented, and Mixed Realities, often powered by cloud computing technologies, have emerged as leading technologies, making online learning more immersive, realistic, and effective compared to traditional models. The internet has given rise to "Intelligent Tutoring Systems", often called ITS, which aim to address common issues associated with conventional learning. These issues encompass engaging learners for longer durations, enabling self-paced learning, mitigating classroom bias and favouritism, and enhancing motivation [5]. These ITS systems often leverage the scalability and resource availability offered by cloud computing, which plays a pivotal role in providing the computational power and data storage required for intelligent, adaptive E-Tutoring systems [52].

Education is undergoing a profound transformation, spurred by the rapid evolution of technology and the growing imperative for personalized and effective learning experiences. Within this transformative landscape, E-Tutoring, a subset of e-learning, stands out as a dynamic educational paradigm that harnesses digital platforms to provide students with customized, interactive, and engaging support, fundamentally enhancing their comprehension and retention of subject matter. A critical development within the E-Tutoring domain is the integration of adaptability, which has paved the way for innovative approaches to address individual learners' diverse learning needs and preferences. Service Oriented Architecture (SOA), a framework rooted in modularity and interoperability [64], has emerged as a promising avenue for developing adaptive E-Tutoring systems, offering the prospect of scalable and flexible solutions that can evolve with the ever-changing landscape of education.

As traditional educational models give way to the possibilities presented by technology, educators increasingly recognize E-Tutoring as a potent instrument for delivering tailored educational experiences, transcending geographical boundaries and demographic constraints [8]. It is a powerful medium for educators and institutions to bridge gaps, reach remote learners, and provide learners with targeted assistance. Nonetheless, achieving true adaptability and personalization in E-Tutoring systems remains a formidable challenge. It demands a design that can swiftly adapt to each student's unique learning patterns, strengths, and weaknesses while simultaneously integrating with a broad array of tools and services, ensuring a cohesive and holistic learning ecosystem.

The introduction of SOA into educational technology represents a pivotal shift away from monolithic, one-size-fits-all approaches towards more dynamic and personalized learning environments. SOA, as a flexible architectural framework, empowers the creation of a network of independent, reusable, and interoperable services, each specialized in catering to specific facets of E-Tutoring. This architecture not only boosts the system's adaptability but also streamlines the incorporation of new educational tools and services, thereby offering a comprehensive learning environment that aligns with the diverse needs of learners and the evolving educational landscape.

This Master of Science thesis explores the conceptualization, design, and implementation of an SOA-based approach to adaptive E-Tutoring systems, aiming to connect the chasm between traditional education and contemporary, individualized learning experiences. The proposed SOA-based method includes strategies for ensuring cross-platform compatibility, enabling students to access E-Tutoring services on a diverse array of devices, including but not limited to desktop computers, laptops, tablets, and smartphones. This approach caters to the diversity of learner preferences and ensures that educational content remains readily available in the digital age, transcending the confines of physical locations and hardware specifications. This thesis will serve as a holistic guide to SOA's integration within the E-Tutoring paradigm, offering insights into its theoretical underpinnings, design principles, practical implementation, and empirical evaluation.

The rest of the thesis is structured in such a way that Chapter 2 reviews the relevant literature, exploring e-learning, service-oriented architecture, tutoring systems, and cloud computing. Chapter 3 highlights prior works in tutoring systems, SOA-based e-learning systems, and research conducted within our group. Chapter 4 presents the problem statement and the proposed methodology, setting the foundation for the study. Chapter 5 covers the implementation and testing of the proposed method. Chapter 6 presents the results, and Chapter 7 concludes our study, summarizing our findings and contributions.

CHAPTER 2

Literature Review

In this Literature Review section, we embark on an in-depth exploration of crucial topics in the fields of E-Learning, Tutoring Systems, and SOA. From the need and benefits of E-Learning to the intricacies of SOA, we delve into the comprehensive landscape of modern education, shedding light on key concepts and their interconnections.

2.1 E-Learning

Online and distance education has emerged as a transformative force, grounded in the core belief that education should be liberated from the confines of in-person interaction [37]. A few decades ago, the notion of learning outside the traditional classroom setting may have appeared far-fetched. However, online education has transitioned from a choice to a vital necessity in today's rapidly evolving digital age.

2.1.1 The Need for E-Learning

Computer-assisted learning originated in the 1960s when the first computerized training programs were introduced. It marked the inception of a form of education that no longer demanded the physical presence of a tutor, a concept with historical roots dating back centuries. However, it was only in the 1990s that online education began to flourish, thanks to a surge in technological advancements and innovations within the e-learning field [8].

Bringing us into the contemporary era, we find online learning exceptionally portable, conveniently accessible through mobile applications, and readily available. The once concept of an on-demand, blended knowledge delivery model has now transitioned from innovation to an established trend in education. Online learning employs diverse devices and tools, ranging from traditional mediums such as CDs to the vast resources offered by the Internet and the interactive platforms of social media. These tools collectively augment theoretical perspectives within the digital learning landscape [40].

As we delve into the dynamic world of e-learning, we see that this educational paradigm has reshaped how knowledge is disseminated, making learning more accessible, flexible, and interactive. The reach and potential of e-learning are extensive, offering a wide array of benefits that transcend the boundaries of traditional education. However, despite its many advantages, e-learning has its challenges.

2.1.2 The Benefits of E-Learning

E-learning, driven by the Internet and digital communication systems, has ushered in a transformation in education. This paradigm shift allows remote access to educational resources, making learning accessible beyond physical boundaries [8]. E-learning is a concept that encompasses the use of digital platforms and multimedia tools to engage learners remotely and interactively. It has evolved from traditional web-based learning to embrace cloud-based e-learning, representing a step forward in online education [25].

Leveraging the Internet as an educational resource and medium has unlocked immense academic potential. It offers learners flexible and personalized learning experiences, making education more adaptable to individual needs and preferences [45]. This widespread use of the Internet in education has fostered active engagement, personalization, and the seamless integration of multimedia elements, enhancing the richness of the learning process [58].

Cloud-based e-learning, an integral part of this digital transformation, offers several distinct advantages [23]. It provides consistency in the learning experience, adapting to the dynamic needs of modern learners. The cloud-based approach enhances accessibility and ease of access, ensuring that educational resources are available when and where learners need them, ultimately enhancing the overall learning experience [8]. This adaptability, along with its widespread accessibility, positions cloud-based learning as a powerful tool in modern education, catering to learners' diverse and evolving needs in today's digital age.

However, as we explore the remarkable benefits of e-learning, navigating the challenges accompanying this transformative mode of education is equally crucial.

2.1.3 The Challenges of E-Learning

While offering many advantages, E-learning presents distinctive challenges that must be addressed for a truly effective digital learning experience. Four primary challenges stand out: **connectivity**, **availability**, **adaptability**, and **personalization**. The connectivity challenge revolves around ensuring that learners have consistent and reliable internet access, which is essential for participating in online courses and accessing digital resources. Availability pertains to making educational materials and platforms accessible to all, irrespective of geographical or socioeconomic constraints, ensuring that no one is left behind in the digital transition. Adaptability encompasses the need to keep pace with rapidly evolving technology and the demands of the modern learning landscape, requiring educators, institutions and solutions to remain agile. Finally, personalization is a complex endeavour in e-learning, tailoring educational content and experiences to individual learners' diverse needs and preferences of individual learners and providing the students with feedback. This task is crucial for engaging and effective online education. Addressing these challenges is integral to harnessing the full potential of e-learning and making it a universally beneficial educational tool.

2.1.3.1 Connectivity

E-learning has emerged as a transformative force, notably amplifying accessibility to education. A key facet of this paradigm is Connectivity, which forms the foundation for a thriving digital learning environment. In contrast to conventional in-person classes, virtual programs, championed by the e-learning model, offer an optimal learning environment with increased flexibility and frequency of access for those who can access their learning materials online [7]. E-learning systems break down geographical and temporal barriers, enabling participants to engage in distance learning from virtually anywhere, ensuring that education knows no bounds. Furthermore, integrating online tutoring programs has enriched the teaching process, offering valuable support and guidance to learners in the digital realm [45]. The Connectivity challenge is not just a technological consideration but a crucial element in levelling the educational playing field and extending the reach of knowledge to a global audience.

2.1.3.2 Availability

Notably, many of these E-Learning systems are proprietary Software exclusively distributed by developer companies. While these systems offer numerous benefits, they can also introduce certain limitations for students who seek the convenience of accessing educational materials from the comfort of their homes and on their schedules [19]. These proprietary platforms may inadvertently hinder the flexibility and accessibility of educational resources, posing challenges for learners striving to adapt their education to their unique needs.

To address the challenges posed by proprietary E-Learning systems, students are increasingly turning to an array of gadgets optimized for mobile learning. These gadgets include smartphones, cellphones, handheld computers, tablets, personal computers, laptops, iPods, and personal media players, reflecting the diverse technological ecosystem that modern learners rely on [2]. This wide spectrum of devices showcases the evolving nature of E-Learning, with adaptability and accessibility emerging as vital cornerstones in contemporary education. In a digital era where personalization and flexibility are paramount, the utilization of various devices highlights the determination of learners to transcend the limitations posed by proprietary systems and explore the full potential of E-Learning.

2.1.3.3 Adaptability

In achieving platform independence, ensuring adaptability becomes an imperative goal. Seamlessly operating across different platforms, regardless of the device or system, is crucial for delivering a frictionless and inclusive educational experience. Many existing E-Learning systems exhibit a monolithic architecture, which often hinders their capacity to make software changes or enhance functionality [49]. This limitation underscores the pressing need for a more agile and adaptable approach to E-Learning. As technology advances and user preferences diversify, it becomes increasingly evident that software solutions must evolve to cater to the dynamic and evolving digital landscape. One promising solution is implementing an adaptive E-Learning system that considers device configurations and user preferences. Such a system promotes a seamless and personalized learning experience and transcends the constraints of monolithic architectures, ensuring that educational resources are accessible and engaging, regardless of the platform or device chosen by the learner. The pursuit of adaptability and platform independence stands at the forefront of the digital education revolution, fostering innovation and inclusivity in the ever-evolving world of E-Learning.

2.1.3.4 Personalization

The concept of personalization has taken a significant stride through personalized learning. This approach goes beyond adapting content to individual preferences; it tailors the entire learning experience to each learner's unique needs, strengths, and interests. Personalized learning leverages advanced technologies to create adaptive curricula, customizing the pace, style, and content delivery for optimal comprehension. By providing targeted resources, individualized challenges, and timely feedback, personalized learning within E-learning strives to create an educational environment that addresses diverse learning styles and maximizes student engagement and knowledge acquisition. This approach is instrumental in fostering a more inclusive and effective learning experience, transcending the limitations of traditional education

[45]. The pursuit of personalization is a critical step in the evolution of E-Learning, ensuring that education is tailored to the unique needs of each learner.

2.2 Tutoring Systems

Within tutoring systems, four core components play pivotal roles in shaping the learning experience: the domain model, student model, tutoring model, and interface/user interaction, as shown in Fig 2.2.1 [4].

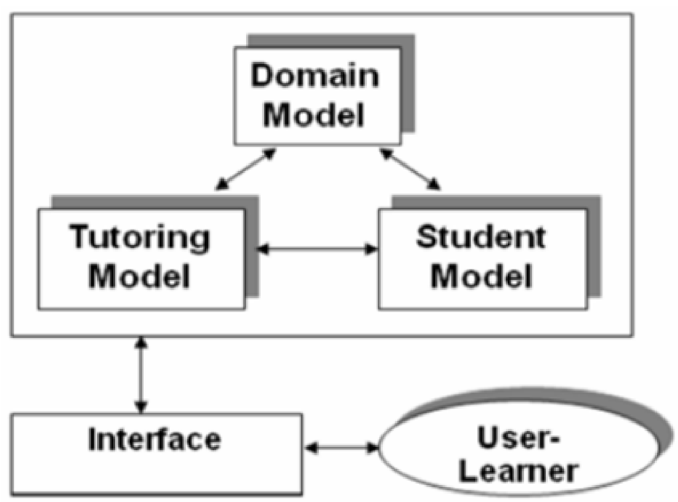


Fig. 2.2.1: Components of a Tutoring System

The domain model is the foundation, representing the knowledge and expertise within the subject matter, defining what the system aims to teach. It encapsulates the content and structure of the educational material, ensuring the system imparts accurate and relevant information to students. The student model is a dynamic element that personalizes the learning journey. It tracks and assesses the student's progress, tailoring the content and pace to their needs. This adaptability enhances engagement and comprehension. The tutoring model is the pedagogical engine, determining how the system delivers instruction and support. It encompasses strategies for presenting content, providing feedback, and assessing the student's performance. Lastly, the interface and user interaction bridge students and the system, affecting the overall usability and user experience. Collectively, these components create a robust foundation for effective and efficient tutoring systems.

2.2.1 Domain Model

The Domain Model within the context of E-Tutoring plays a pivotal role in shaping the structure and content of the tutoring system. The tutoring system is designed to teach a meticulously structured and organized representation of the subject matter or knowledge domain [4]. This model is instrumental in ensuring the educational content is logically structured and seamlessly aligned with the defined learning objectives, providing students with a clear and coherent learning path.

The Domain Model acts as a detailed map of the entire curriculum, meticulously outlining the sequence of topics, learning outcomes, and dependencies between various concepts within the domain [4]. This comprehensive overview allows the system to be adaptable to a wide spectrum of educational disciplines, ranging from mathematics to languages, making it a versatile tool for diverse learning areas [59].

Furthermore, the Domain Model is the bedrock for personalization within the tutoring system. It enables the system to tailor instruction based on individual learner needs and progress within the specific domain, creating a more personalized and effective learning experience [3]. The development of the Domain Model often involves instructor-aided techniques, where instructors contribute their domain-specific expertise to define knowledge components, relationships, and content structure, ultimately enriching the system's ontology and ensuring the system's accuracy and relevance within the chosen domain.

2.2.2 Tutoring Model

The Tutoring Model within the domain of E-Tutoring is a cornerstone of the system, encompassing a diverse array of strategies, methods, and approaches devised to instruct and guide learners throughout their educational journey efficiently and effectively. This model is the pedagogical engine that propels the system's ability to facilitate learning [4].

At its core, the Tutoring Model integrates various instructional strategies, including problem-solving techniques and feedback mechanisms. These strategies enhance learning and help learners grasp complex concepts [41]. Incorporating adaptive learning techniques within the model further enriches the system's capabilities. By continuously assessing individual learner progress and adapting instruction to suit their needs, the system ensures a tailored and responsive learning experience.

Feedback loops are integral to the Tutoring Model, offering learners constructive insights into their performance, thus guiding them toward improvement. This real-time feedback mechanism not only aids in correcting errors but also fosters a deeper understanding of the subject matter.

Additionally, the Tutoring Model may incorporate motivational elements, such as gamification or rewards, to engage and inspire learners [15], ultimately enhancing the learning experience. These motivational elements increase learner engagement and retention by making learning interactive and enjoyable.

Furthermore, the Tutoring Model employs run-time assessment techniques to continuously evaluate the correctness of learners' calculations during the session. This feature provides real-time feedback and guidance, ensuring learners receive immediate support and correction, fostering a more effective and efficient learning experience.

2.2.3 Student Model

The Student Model within the realm of E-Tutoring assumes a pivotal role in understanding and tailoring the learning journey of individual learners. It is responsible for creating and maintaining detailed profiles of each learner, meticulously capturing their unique learning preferences, strengths, weaknesses, and overall progress [4]. This compilation of learner-specific data forms the foundation for personalization within the tutoring system.

Personalization, a key objective of the Student Model, is achieved using these learner profiles to adapt the learning experience [53]. The system tailors the content, pace, and difficulty level to match each student's specific needs and abilities, thus ensuring that the learning process is engaging and effective.

Moreover, the Student Model actively tracks and records each learner's progress, providing a comprehensive overview of their achievements and areas that may require further attention and improvement. It employs data analytics and trend analysis to identify learning patterns, allowing for adapting the tutoring experience accordingly. This data-driven approach benefits learners and provides educators with valuable insights into learner performance, enabling informed instructional decisions [70].

Furthermore, the Student Model supports ongoing adaptation, ensuring the system's instructional strategies evolve in response to learner knowledge and skills changes. This dynamic approach is fundamental to providing a personalized, efficient, and effective E-Tutoring experience.

2.2.4 Interface Model

The Interface Model, a critical component in the landscape of E-Tutoring systems, has evolved in its quest to enhance the user experience. In the early stages, computer systems predominantly employed text-based command line interfaces, demanding users to memorize intricate commands, making interaction cumbersome and intimidating, especially for non-expert users. Recognizing the need for a more efficient and user-friendly alternative, Graphical User Interfaces (GUIs) emerged as a pivotal innovation. They offered a visual and intuitive interaction, greatly enhancing accessibility and usability [10].

The Interface Model in E-Tutoring Systems extends its influence beyond mere visual design. It encompasses dynamic content adaptation, allowing for the scaling of content delivery based on user preferences and device capabilities [31]. This adaptive approach ensures that the learning experience is tailored to the specific needs of each user, promoting a more engaging and effective learning journey.

Moreover, the Interface Model integrates advanced handwriting recognition techniques, enabling learners to input answers, equations, or annotations using digital ink. This sophisticated feature allows for recognizing and interpreting handwritten content, enhancing the natural interaction method and fostering an interactive and engaging learning environment.

The Interface Model plays a pivotal role in ensuring the usability and accessibility of E-Tutoring systems [20], catering to diverse user needs and enhancing the overall learning experience. It stands as a testament to the commitment of E-Tutoring to providing an intuitive, adaptive, and engaging interface for both novice and experienced users.

In conclusion, tutoring systems stand at the forefront of modern education, offering innovative solutions to enhance learning experiences. These systems are reshaping the educational landscape by integrating advanced technologies, personalized approaches, and adaptive learning models. Their ability to cater to individual needs, provide real-time feedback, and foster interactive engagement marks a step towards more effective student-centred learning. As we navigate the digital era, tutoring systems continue to evolve, ensuring that education remains accessible, engaging, and tailored to the unique requirements of learners worldwide.

2.3 Service Oriented Architecture

It is a fundamental paradigm in modern software design and integration. At its core, SOA promotes the development of modular and interoperable services that can be seamlessly combined to create complex, distributed systems. By focusing on service reusability and flexibility, SOA offers a dynamic framework for building agile, scalable, and efficient software solutions [30, 34, 60, 61].

2.3.1 Cloud Computing and SOA

The fusion of SOA with cloud computing presents a potent synergy that catalyzes adaptability and efficiency in modern e-learning systems. Cloud computing, a transformative technology paradigm, provides a dynamic platform for delivering various computing services, from storage and processing power to applications accessible over the Internet. This architectural shift ushers in a new era of flexibility and scalability, efficiently enabling on-demand access to computing resources. Notably, the advantage

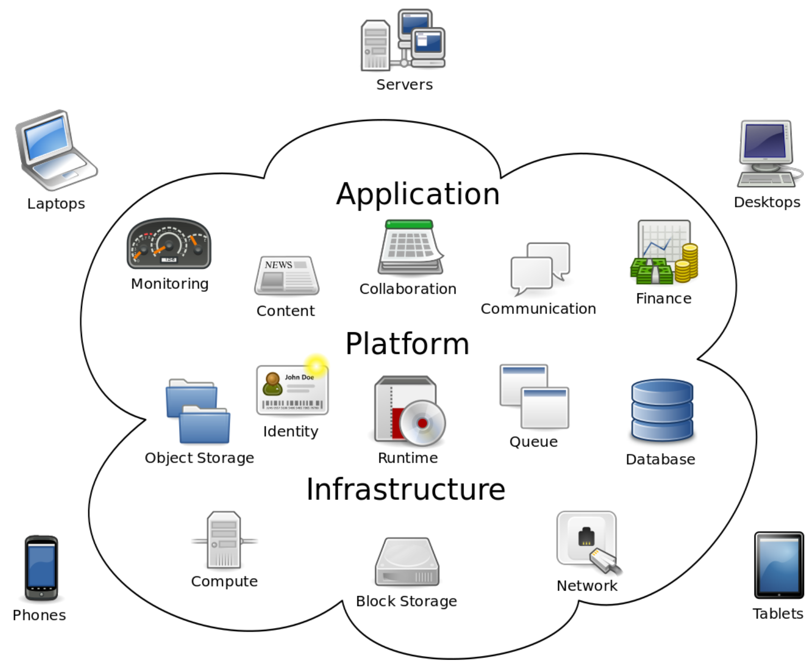


Fig. 2.3.1: General Cloud Computing Architecture

of cloud computing transcends the boundaries of space and time, allowing users to access applications and data at any location and time, thereby ensuring unrestricted access to information and resources [52].

When integrated into cloud-based e-learning, service-oriented systems further extend the benefits of cloud computing. This union allows for the creation of modular, interoperable, and scalable educational services. Learners gain the advantage of having access to educational content and applications at any time and anywhere, thereby enhancing the learning experience. The inherent flexibility of SOA, combined with the resource accessibility of cloud computing, promotes adaptability in e-learning systems. This adaptability is essential in catering to the diverse needs of learners and addressing the challenges posed by the rapidly evolving landscape of digital education.

Moreover, the symbiosis of cloud computing and SOA opens the door to innovative solutions in e-learning, offering a promising avenue for developing agile, scalable, and efficient educational platforms. It empowers educators and institutions to deliver learning resources and services with unprecedented flexibility and cost-efficiency, ultimately contributing to the evolution of modern education.

2.3.2 Cloud Computing Systems

There are various aspects of Cloud Computing Systems, from their different categories and classifications to the use of analogies to enhance understanding. Additionally, we will undertake a comparative analysis to highlight the strengths and weaknesses of those systems concerning alternative architectural approaches.

2.3.2.1 Categories

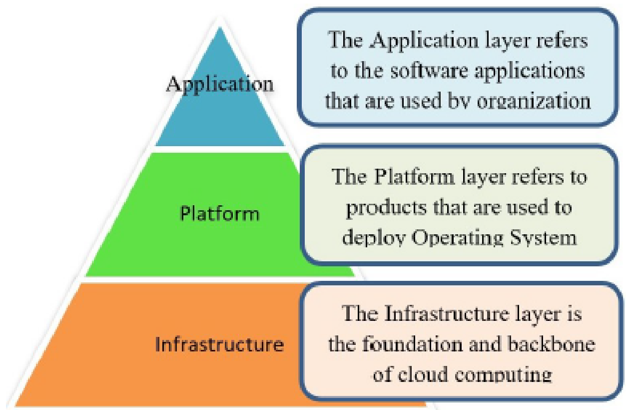


Fig. 2.3.2: Categories of Cloud Computing Systems [8]

Its systems comprise three main categories: Software, Platform, and Infrastructure as a Service (SaaS, PaaS, and IaaS). As stated in Fig 2.3.2, these categories represent the different levels of abstraction within the framework, each offering a unique set of benefits and capabilities.

- **Infrastructure as a Service (IaaS)**: IaaS represents the foundational layer of cloud computing, offering virtualized computing resources to users over the Internet. As stated in [44], IaaS providers deliver essential infrastructure components, including virtual servers, storage, and networking, enabling customers to lease these resources without needing physical hardware or data centers. Prominent IaaS providers like Amazon EC2 and Microsoft Azure have redefined how businesses and researchers access computational resources, offering scalability and flexibility for many use cases. This category forms the building blocks for constructing higher-level cloud services, fostering cost-effective and on-demand resource provisioning.

- **Platform as a Service (PaaS)**: PaaS occupies the middle layer in the cloud service hierarchy and focuses on providing a comprehensive environment for application development and deployment, as noted in [48]. PaaS offerings facilitate collaboration among multiple developers and streamline the software development cycle by integrating tools for coding, testing, and deployment. Leading PaaS providers, such as Google App Engine, Amazon Web Services, Heroku, and OpenShift (Red Hat), enhance development speed and efficiency by abstracting infrastructure management complexities, allowing developers to focus on their code. PaaS empowers businesses to create, deploy, and scale applications easily, making it a preferred choice for software development teams.

- **Software as a Service (SaaS)**: SaaS is the top layer in the cloud computing hierarchy, focusing on delivering software applications through the Internet on a subscription basis. Users access applications hosted on the provider's servers, relieving them of local software management tasks, as elucidated by Palos-Sanchez [50]. SaaS offers various applications tailored for businesses, individuals, and educational institutions, accessible from any internet-connected device. This category has reshaped the software delivery model, providing convenient and cost-effective solutions for various domains, including office productivity tools, customer relationship management, and collaborative platforms.

2.3.2.2 Analogy

Imagine diving into cloud services through a delightful pizza analogy using Fig 2.3.3. In this analogy, cloud services are categorized into four sections, each represented by

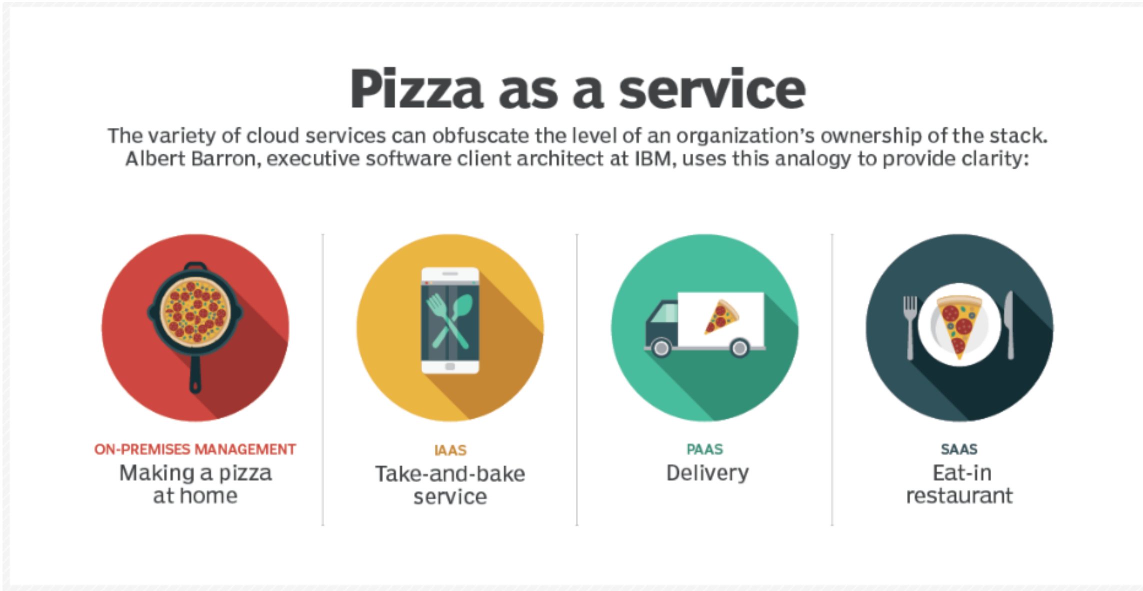


Fig. 2.3.3: Analogy of Cloud Computing Categories

a different stage of the pizza-making process.

- **On-Premises Management: Making a Pizza at Home (Red Section)**: Picture this as cooking a pizza at home. In on-premises management, the individual is responsible for everything - from sourcing the raw ingredients to baking and serving. It entails much work but offers complete control and customization, similar to managing on-site IT infrastructure.

- **IAAS: Take-and-Bake Service (Yellow Section)**: Cloud services resemble purchasing a ready-to-cook pizza from a store. At the same time, the effort of preparing the dough and toppings is spared. The need to bake it at home remains. This aligns with IaaS, where the service provider handles the Infrastructure while retaining control over the configuration and management.

- **PAAS: Delivery (Light Green Section)**: Think of this as ordering a pizza for delivery. The pizza arrives ready to eat without worrying about the cooking process. In the cloud context, this corresponds to PaaS, where the provider takes care of the underlying Infrastructure, allowing developers to focus on coding and deployment.

- **SAAS: Eat-In Restaurant (Dark Green Section)**: Consider dining in a restaurant. Order your favourite pizza and enjoy it without concern with the cooking or serving. Similarly, SaaS provides fully managed applications accessible over the Internet, making it hassle-free for users.

This infographic provides an informative illustration of how cloud services can vary in terms of what is managed by users and what is managed by service providers. Like pizza, cloud services have an option to suit everyone's taste. Whether you prefer complete control or a fully managed experience, cloud services offer a range of choices to cater to your specific needs and preferences.

2.3.2.3 Comparison

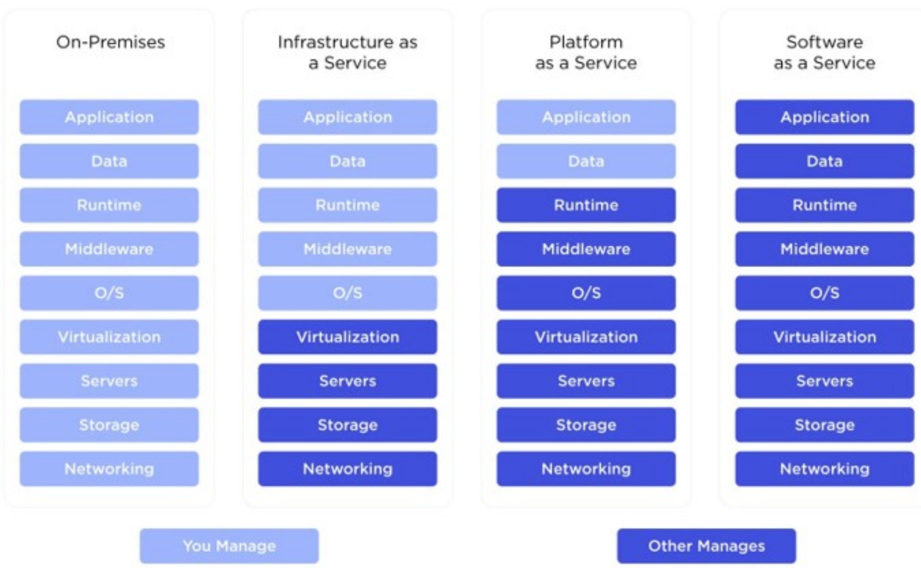


Fig. 2.3.4: Comparison of Cloud Computing Systems

In comparing these three fundamental categories of cloud services, it is imperative to discern their distinct characteristics and tailored applications. IaaS caters to organizations seeking flexible infrastructure solutions while alleviating the cumbersome responsibility of managing physical hardware. IaaS is the foundation for other cloud services, offering scalability and resource provisioning on demand.

PaaS is primarily designed to meet the requirements of developers and teams in search of a streamlined and efficient environment for application development. PaaS abstracts infrastructure complexities, enabling developers to focus on coding and deployment, significantly expediting the software development cycle.

SaaS is optimized for end-users who crave convenient access to ready-made software applications via the Internet. SaaS relieves users from the intricacies of local software management and places the responsibility on the service provider. This category provides various applications, from office productivity tools to customer relationship management systems, all readily available from virtually any internet-connected device.

Ultimately, selecting these categories hinges on specific requirements, available resources, and overarching objectives. In their multifaceted forms, cloud services offer a versatile and customizable solution, accommodating the diverse needs of a wide spectrum of users, from businesses to developers and individual users. This flexibility ensures that cloud services remain pivotal and adaptable in the contemporary technology landscape.

2.3.3 SOA Components

SOA has three main components described in [61] and in Fig. 2.3.5, service, broker, and consumer. These components work together to enable a flexible and modular approach to designing and implementing software systems. Let us understand each component in detail.

- **Service:** A service within an SOA system is a fundamental building block. It represents a self-contained, modular, and well-defined unit of functionality or business logic that users can access and execute over a network. Services can perform various tasks. They are designed to be platform-independent, allowing them to run on different operating systems and technologies. Services expose their functionality through standardized interfaces, often utilizing web service protocols, which makes them accessible online. Furthermore, services are

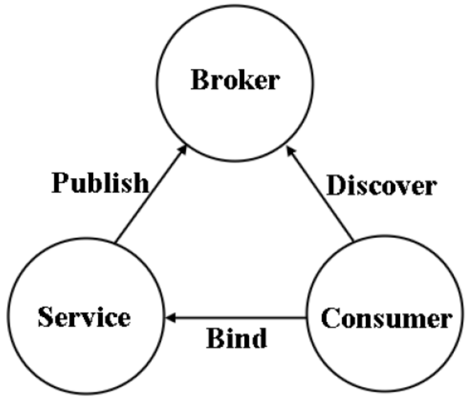


Fig. 2.3.5: Components of an SOA System

intentionally designed to be loosely coupled, enabling them to be developed, deployed, and maintained independently. This independence fosters flexibility in system design and encourages reusability.

- **Broker:** The broker, sometimes called a service broker or service registry, acts as an intermediary within an SOA system. It is critical in managing and facilitating interactions between services and consumers. Brokers provide a centralized registry where benefits can be discovered and located. They maintain metadata about available services, including their descriptions, locations, and access points. When a consumer (a component or application that needs a specific service) requires access to a service, it can query the broker to find the appropriate service. The broker then helps establish the connection between the consumer and the service, ensuring the proper assistance is invoked.

- **Consumer:** The consumer in an SOA system is any component or application that uses or "consumes" services to perform specific tasks or functions. Consumers can range from web applications and mobile apps to other services within the system. Consumers interact with services by sending requests and receiving responses. These interactions are typically mediated through the broker, which helps locate the required service and facilitates communication. Consumers benefit from the flexibility and modularity of the SOA approach because they can access and use services independently without needing to understand the internal details of the services they consume.

The evolution of E-Learning has traversed various technological landscapes, with a notable progression from client-server architectures to the contemporary embrace of SOA, as highlighted in [Hinchliffe]. This paradigm shift in E-Learning systems signifies a move towards distributed environments, where functions are implemented as standalone web services. This architectural transformation enhances system capabilities, fostering modularity and flexibility.

In the realm of SOA-based E-Learning systems, the adoption of web services is pivotal. These services, designed as standalone entities, contribute to a modular and interoperable ecosystem. The innovation lies in the composition of web services, amalgamating platform-independent services from diverse web corners.

Security and knowledge enrichment emerge as focal points in the design of web services within E-Learning systems, as emphasized in [38]. The integration of robust security measures ensures the protection of sensitive educational data, fostering a secure online learning environment. Simultaneously, the emphasis on knowledge enrichment aligns with contemporary interactive and Artificial Intelligence (AI) trends and enhanced learning experiences. These AI-infused functionalities contribute to a more dynamic and personalized educational journey, catering to the evolving needs of modern learners.

The transition to SOA-based E-Learning systems signifies a paradigmatic shift toward distributed and modular architectures. Implementing standalone web services and the innovative practice of web service composition opens new frontiers in functionality and interoperability. Security and knowledge enrichment take center stage, ensuring not only the protection of educational data but also the delivery of interactive and AI-enhanced learning experiences, thus shaping the future landscape of digital education.

In summary, an SOA system's core components, services, brokers, and consumers work together to create a flexible, modular, and efficient software architecture. Services provide well-defined functionality, brokers enable service discovery and management, and consumers leverage these services to perform various tasks, ultimately contributing to the scalability and adaptability of the system.

CHAPTER 3

Prior Works

In this chapter, we journeyed through prior works in the dynamic fields of tutoring systems, SOA-based e-learning systems, and personalized e-tutoring. This retrospective exploration allows us to trace the evolution and development of these domains, shedding light on the milestones, innovations, and research contributions that have shaped the landscape of modern education technology. As we delve into the wealth of knowledge from earlier studies and projects, we understand these crucial areas' challenges, solutions, and future directions. Join us as we uncover the valuable insights and experiences accumulated by researchers, educators, and practitioners, all of which have paved the way for innovative solutions and approaches in contemporary e-learning and tutoring systems.

3.1 Tutoring Systems

This chapter delves into a selection of prior works in ITS, each contributing unique insights and innovations. The exploration begins with the study by Cai et al., introducing bandit algorithms to personalize educational chatbots [18], followed by Lippert et al.'s research on multiple agent designs in Conversational Intelligent Tutoring Systems [42]. Eryılmaz and Adabashi's work on developing an Intelligent Tutoring System using Bayesian Networks and Fuzzy Logic for higher student academic performance is also examined [24]. Lastly, N. Shi et al. present their language chatbot design, implementing English Language Transfer Learning Agent Apps [57]. All these approaches are mentioned in Table 3.1.1

Table 3.1.1: Prior Works in Intelligent Tutoring Systems

Author(s)	**Title**	**Contribution**
Cai, William & Grossman, Josh & Lin, Zhiyuan & Sheng, Hao & Wei, Johnny & Williams, Joseph & Goel, Sharad. (2021) [18]	Bandit algorithms to personalize educational chatbots	Rule-based chatbot that explains math, questions, & provides personalized feedback
Lippert, Anne & Shubeck, Keith & Morgan, Brent & Hampton, Andrew & Graesser, Arthur. (2020) [42]	Multiple Agent Designs in Conversational Intelligent Tutoring Systems	Uses technologies like MACTIS, iDrive, CITS, and AutoTutor to provide results on the effectiveness of multiple agent designs
Eryılmaz M, Adabashi A (2020) [24]	Development of an Intelligent Tutoring System Using Bayesian Networks and Fuzzy Logic for a Higher Student Academic Performance	Uses Bayesian Networks and Fuzzy Logic to create an intelligent tutoring system that can be used to improve student academic performance
N. Shi, Q. Zeng & R. Lee (2020) [57]	Language Chatbot – The Design and Implementation of English Language Transfer Learning Agent Apps	Transfer learning-based English Language learning chatbot

The study by Cai, William and his co-authors introduces a rule-based chatbot designed to explain mathematical concepts, answer questions, and deliver personalized feedback [18]. This chatbot is driven by bandit algorithms, which enable adaptability and personalization based on individual learning needs. The key contribution of this work lies in its emphasis on personalization and adaptability, which can significantly enhance the learning experience for students. The chatbot tailors responses to each learner's unique requirements by leveraging bandit algorithms, promoting a more engaging and effective learning environment. However, a notable limitation of this chatbot is its reliance on text input, with no provision for voice input. While text-based interactions can be effective, incorporating voice recognition capabilities could further enhance accessibility and user engagement, especially for learners who prefer or require voice-based communication. To address this limitation, future iterations of the chatbot could explore the integration of voice interfaces, offering a more comprehensive and inclusive learning experience.

Lippert, Anne, and her team delve into the development of Conversational Intelligent Tutoring Systems (CITS) and explore various agent designs, including MACTIS, iDrive, and AutoTutor [42]. Their research aims to evaluate the effectiveness of these diverse agent designs in enhancing the interactivity and responsiveness of educational systems. The key contribution here lies in thoroughly examining multiple agent designs, shedding light on their potential to provide personalized and effective tutoring experiences. However, it is essential to acknowledge the computational demands of these technologies, particularly when dealing with extensive datasets and complex interactions. These resource-intensive requirements could present challenges, particularly for institutions or settings with limited computational power. Addressing this limitation may require optimizing the algorithms and models used in these conversational agents to ensure efficient operation while maintaining their effectiveness.

Eryılmaz and Adabashi's research introduces an Intelligent Tutoring System (ITS) that leverages Bayesian networks and fuzzy logic to enhance student academic performance [24]. This approach combines probabilistic reasoning with fuzzy sets to create an adaptive learning system. The key contribution here is the development of an ITS that offers personalized and adaptive learning experiences, enabling students to achieve higher academic performance. One notable limitation of this approach is the monolithic software architecture, primarily relying on Bayesian networks. This reliance on a specific technology can result in high computational power demands, which may pose challenges in resource-constrained environments. To address this limitation, future work could focus on optimizing the computational efficiency of the system while preserving its adaptability. Exploring alternative technologies or hybrid approaches may provide a more balanced solution.

N. Shi, Q. Zeng, and R. Lee's study [57] centers on designing and implementing an

English Language learning chatbot that employs transfer learning. This innovative chatbot leverages pre-trained large language models (LLM) to enhance its language understanding and generation capabilities, offering personalized language learning experiences. The key contribution here is the exploration of transfer learning, a technique that holds the potential to provide learners with highly adaptive and effective language learning experiences. However, it is important to recognize that the success of transfer learning models often hinges on the availability of substantial datasets and computational resources for training. This limitation underscores the importance of accessible data and computing power for effective implementation. To overcome this limitation, researchers could focus on developing strategies to make transfer learning more efficient and adaptable to varying data and resource constraints.

The selected prior works in intelligent tutoring systems demonstrate the evolution of this field, marked by approaches to enhance personalized learning experiences. These contributions pave the way for more effective and engaging educational strategies. While each study brings unique strengths, they also acknowledge limitations that serve as valuable insights for future research. These limitations underscore the ongoing challenges and opportunities in developing intelligent tutoring systems as the field continues to evolve and innovate to meet the diverse needs of learners.

3.2 SOA-Based E-Learning Systems

Service-oriented architecture, or SOA, has ushered in an era of flexibility, scalability, and customized learning experiences, tremendously impacting e-learning systems. In this investigation, we examine some earlier studies mentioned in Table 3.2.1 that have aided in creating e-learning systems based on SOA. Each of these research sheds light on the development of this dynamic discipline by introducing novel ideas and methods along with their acknowledged limits.

An SOA-based tutoring system is presented by Wijekumar, Kausalai [67], and her co-authors to improve content area reading comprehension for struggling fourth and fifth-grade readers while also supplementing instructor expertise. With an em-

Table 3.2.1: Prior Works in SOA-Based E-Learning Systems

Author(s)	**Title**	**Contribution**
Wijekumar, Kausalai & Meyer, Bonnie & Lei, Pui-Wa and Beerwinkle, Andrea & Joshi, R Malt (2019) [67]	Supplementing teacher knowledge using web-based Intelligent Tutoring System for the Text Structure Strategy to improve content area reading comprehension with fourth and fifthgrade struggling readers	An SOA-based tutoring system focusing on comprehensive learning
Araka, Eric & Muchemi, Lawrence (2017) [11]	An SOA Framework for Web-based E-learning Systems – A case of Adult Learners	A web-based E-Learning system for Adult Learners which uses a MOODLE pattern
Hong, Chin-Ming & Chen, Chih-Ming & Chang, Mei-Hui & Chen, Shin-Chia (2007) [27]	Intelligent Web-based Tutoring System with Personalized Learning Path Guidance	Web-based learning system uses a genetic algorithm for personalized e-learning

phasis on holistic learning, this system seeks to enhance students' comprehension of expository texts. This work's primary contribution is its emphasis on content area reading comprehension and comprehensive education. Through SOA, the system offers struggling readers a methodical and flexible way to improve their reading abilities. Acknowledging a constraint inside this system is imperative - it does not provide a customized learning environment. Although it is useful for enhancing reading comprehension in the content area, tailoring it to each student's unique learning needs might be improved.

With an emphasis on adult learners, the research of Araka, Eric [11], and Muchemi presents an SOA framework for web-based e-learning systems. Their effort creates a web-based e-learning environment customized for adult learners, utilizing the Moodle pattern. The primary contribution is creating an SOA framework that supports adult learners and provides them with an adaptable and easily available online learning environment. The system's immediate online availability is one drawback to consider, which might be problematic for students with inconsistent internet service. Furthermore, the system's communication is one-way only, which could restrict interaction. The accessibility and engagement of the system for adult learners should be further increased by investigating mobile ways to access and improve communication features.

An intelligent web-based tutoring system with an emphasis on individualized learning path assistance is presented by Hong, Chin-Ming, and his co-authors in [27]. This system uses a genetic algorithm to provide users with individualized e-learning experiences. The work's main contribution is incorporating a genetic algorithm for personalization, which enables learners to get content and advice adapted to their needs. Nevertheless, this system's lack of a solid and expandable design is one of its drawbacks. A key component of e-learning systems is scalability, which guarantees that the platform can support increasing users and resources. Furthermore, it does not seem like the system allows for voice input, which might improve accessibility and user engagement—especially for students who prefer voice-based interactions.

In conclusion, the selected prior works in SOA-based e-learning systems underscore the transformative impact of service-oriented architecture on education. These contributions introduce innovative approaches to online learning, catering to diverse learners' needs. While each study brings unique strengths, they also acknowledge limitations, providing valuable insights for future research and development in this ever-evolving domain.

3.3 Personalized E-Tutoring

The landscape of personalized e-tutoring has witnessed remarkable advancements, with researchers and practitioners striving to create more tailored, engaging, and effective learning experiences. This review explores prior works that have contributed to developing personalized e-tutoring systems. Each of these studies mentioned in Table 3.3.1 introduces unique insights and approaches alongside their recognized limitations, shedding light on the evolution of this dynamic field.

Ashwitha's research [33] presents an innovative approach to personalized e-tutoring

Table 3.3.1: Prior Works in Personalized E-Tutoring

Author(s)	**Title**	**Contribution**
Jawahar, Ashwitha Vichuly (2023) [33]	Personalized ECA Tutoring with Self-Adjusted POMDP Policies and User Clustering	Implemented machine learning algorithms for better dialogue management and knowledge levels
Vyas, Niyati (2022) [66]	An Approach of Using Embodied Conversational Agent for Personalized Tutoring	ECA interaction with users in quiz style Construction of tutoring ontology enables the dialogue management of ECA to track the history of belief states
Szucs, Tristan (2020) [63]	Lip Synchronization for ECA Rendering with Self adjusted POMDP Policies	Self-adjusting policies for dialogue management. Established lip synchronization architecture to create an interaction experience of ECA

through Embodied Conversational Agents (ECAs). The system incorporates machine learning and reinforcement learning algorithms for improved dialogue management and knowledge level tracking. The key contribution of this work is the application of self-adjusted Partially Observable Markov Decision Process (POMDP) policies, enabling ECAs to adapt their interactions to individual user needs. However, it is crucial to recognize a limitation in this approach. The processing times can increase when machine learning algorithms deal with large and complex datasets. As these personalized e-tutoring systems continue to evolve, optimizing the computational efficiency of such algorithms will be an important consideration, ensuring that real-time interactions remain seamless and responsive.

Vyas, Niyati's research [66] leverages ECAs for personalized tutoring. The system employs a quiz-style interaction approach, allowing learners to engage with the ECA dynamically and interactively. Additionally, the creation of a tutoring ontology enables the dialogue management of ECAs to track the history of belief states, fostering a personalized learning journey. One area for improvement to consider is the absence of a robust architecture designed for scalability. Scalability is a critical aspect of personalized e-tutoring systems, especially as they cater to a growing number of learners. A scalable architecture can ensure that the system remains efficient and responsive as user numbers increase. Future iterations could explore architectural enhancements to address this limitation.

Szucs, Tristan's research [63] delves into lip synchronization for ECAs. This work introduces self-adjusted POMDP policies for dialogue management, enabling ECAs to adapt their interactions to the learners' needs dynamically. The system also establishes a lip synchronization architecture to enhance the interactive experience of ECAs. A limitation to note is that while the system offers audio output, the input from users remains primarily textual, lacking vocal information. Incorporating verbal input can enhance the naturalness of interactions and cater to learners who prefer voice-based communication. Future developments may explore this avenue to offer a more comprehensive and inclusive learning experience.

In conclusion, prior works in personalized e-tutoring highlight the transformative potential of ECAs and advanced dialogue management strategies. These contributions offer unique strengths, such as the application of self-adjusted POMDP policies, interactive quiz-style interactions, and lip synchronization, creating more engaging and personalized e-tutoring experiences. While these works bring innovation to the field, they also acknowledge certain limitations, providing valuable insights for future research and development in this ever-evolving domain.

Our retrospective exploration allowed us to trace the evolution and development of these domains, shedding light on the milestones, innovations, and research contributions that have shaped the landscape of modern education technology. We better understood these crucial areas' challenges, solutions, and future directions as we delved into studies and projects.

CHAPTER 4

Problem Statement and Proposed Method

This chapter begins by explaining the problem this thesis aims to solve, followed by the suggested approach

4.1 Problem Statement

The current project is positioned to tackle several obstacles that hinder the effectiveness and accessibility of Intelligent ECA Tutoring Systems. These difficulties include the following problems:

1. **Absence of a Standardized Architecture**: One of the foremost challenges is the absence of a standardized architecture for ECA-based e-tutoring systems. This deficiency has far-reaching implications, primarily hampering the scalability, flexibility, and interoperability of e-tutoring platforms. The absence of a unified architecture can lead to issues integrating various components, limiting the system's ability to adapt and expand efficiently.

2. **Geographical Barriers**: Geographical barriers present another concern. Access to e-learning platforms is often constrained by geographical factors, denying students educational opportunities, irrespective of location. Overcoming these barriers and ensuring that all students have equal access to quality education is a fundamental challenge that must be addressed.

3. **Lack of Voice Interaction**: The absence of a robust voice interaction system within ECA-based e-tutoring systems poses a notable challenge. Effective and natural communication is hindered, as more than textual interactions are needed to meet learners' diverse needs and preferences. Voice interactions can enhance the learning experience, making it more engaging and accessible.

4. **Software Adaptability**: Another crucial challenge lies in the adaptability of the software. ECA systems must function seamlessly across a wide range of devices and platforms. The lack of software adaptability can result in operational challenges and inconsistencies across different environments, ultimately affecting the user experience negatively.

The recognition of these challenges underscores the significance of this project, as it endeavours to devise innovative solutions that will not only mitigate these issues but also advance the field of e-tutoring with ECAs. The proposed methodologies and architecture aim to create a standardized, accessible, and adaptable system that fosters natural and effective interactions while transcending geographical boundaries. This chapter further elucidates our contributions and the methods to address these challenges effectively.

4.2 Proposed Contribution

This thesis sets out to make contributions that can modify the Intelligent ECA Tutoring Systems field. The following are the anticipated contributions:

1. **Standardized Architecture with an SOA Approach**: This research proposes designing and implementing a standardized architecture, leveraging an SOA approach tailored for e-tutoring systems. Introducing a standardized architectural framework is expected to resolve the prevailing challenges of scalability, flexibility, and interoperability. This contribution provides a solid foundation for seamless integration and system component expansion.

2. **Cloud-Based Computation**: An innovative approach will be explored to offload heavier computational tasks to cloud servers. This strategic shift is envisioned to empower the e-tutoring system to handle complex tasks efficiently without compromising the user experience. By harnessing cloud computing capabilities, learners can expect responsive interactions and timely access to valuable resources, ultimately enhancing the educational journey.

3. **Voice-Based Interaction**: A notable contribution of this research is enabling learners to interact with the virtual tutor. This breakthrough enhances the naturalness and interactivity of communication within the e-tutoring system. The study aims to foster a more immersive and engaging learning experience by incorporating voice-based answering. Learners will have the opportunity to articulate their thoughts, questions, and responses, thus promoting a richer educational interaction.

4. **Software Adaptability for Cross-Platform Compatibility**: A pivotal aspect of this research is dedicated to ensuring software adaptability. The goal is to create a system that seamlessly operates across various devices and platforms. By addressing compatibility issues, this contribution aims to provide optimized user experiences, regardless of the device or platform learners choose. It ensures that the benefits of ECA-based e-tutoring are accessible to all.

These proposed contributions signify a step forward in ECA Tutoring systems. They are designed to surmount existing challenges and pave the way for a more standardized, adaptable, and user-friendly educational environment. The subsequent sections delve into the architecture and methodologies that will bring these contributions to fruition.

4.3 Current State of the System

In the following section of this thesis, we will navigate the intricacies of the system developed thus far by our research group, offering a comprehensive understanding of its

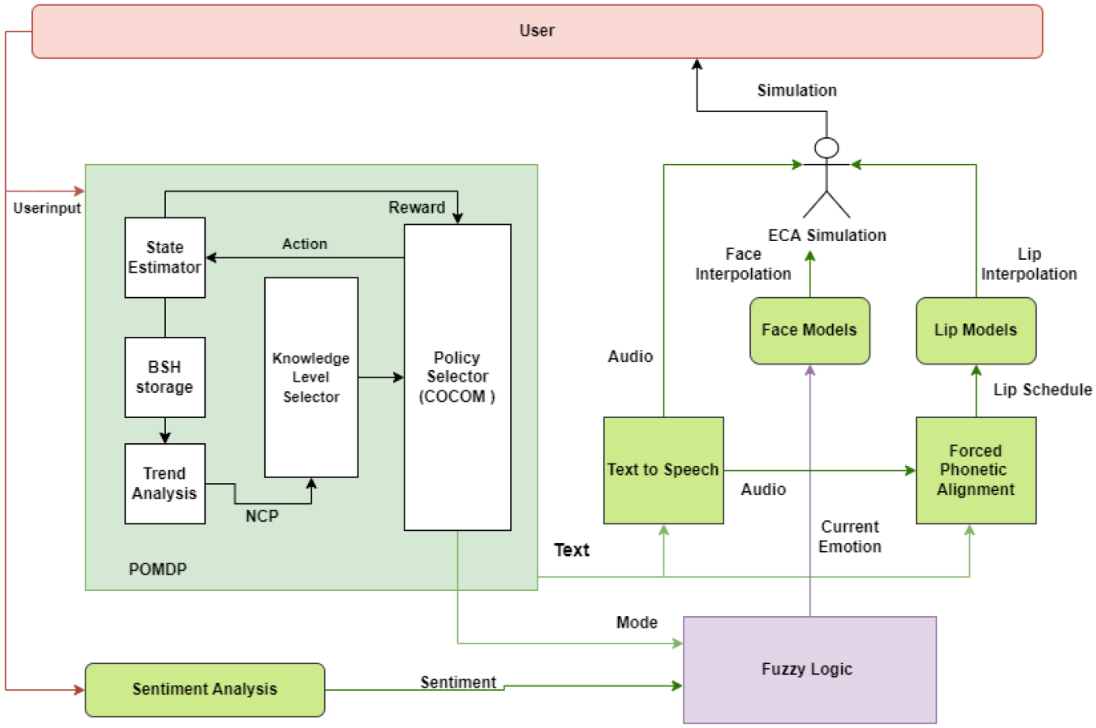


Fig. 4.3.1: Simplified Architecture [39]

components, design principles, and operational functionalities. The exploration will provide detailed insights into the system's current state, elucidating today's progress. By delving into the architecture shown in Fig 4.3.1, algorithms, and interface, aiming to present a holistic view of the system's development, showcasing its potential impact on addressing the identified problems in the existing ECA-based e-tutoring landscape. This exposition will lay the groundwork for a nuanced analysis of the proposed contributions and methodologies. It will offer readers a clear trajectory of the project's evolution and its implications for educational technology.

In Fig 4.3.1, the user's input text initiates a multifaceted process within the system. The POMDP and ECA systems come into play to process this input and formulate a meaningful response. Let us delve into how each component comes into action to facilitate this process.

- **State Estimator**: The state estimator utilizes student input to assess and determine the next system state. Subsequently, the Belief State History (BSH) storage acts as a recipient for these states from the state estimator.

- **BSH Storage**: The BSH functions as a repository, storing a comprehensive history of belief states in which the system has encountered or currently resides. This stored information is a valuable resource, enabling in-depth analysis to derive additional insights about the user and enhance the system's understanding and responsiveness.

- **Trend Analysis**: Applying Discrete Wavelet Transform (DWT) to the input facilitates trend analysis, revealing the number of sharp variation points known as change points (NCP). This process enables the system to identify and quantify changes, determining the user's knowledge level for the subsequent selection of an appropriate knowledge policy.

- **Number of Change Points (NCP)**: The NCP is a crucial component of the system, as it provides a quantitative measure of the user's knowledge level. This information is leveraged to select the appropriate knowledge policy, which is then used to formulate the system's response.

- **Knowledge Level Selector**: The policy selector module, informed by the knowledge level, dynamically selects a new policy. It interfaces with the state estimator, receiving rewards. This interaction ensures adaptive decision-making, as the module leverages the knowledge level to optimize policy selection.

- **Policy Selector**: This pivotal module generates the mode and optimal action text, integral for decision-making beyond the POMDP system. Integration with fuzzy logic in the diagram's bottom right employs it to determine the emotion the ECA should exhibit, contributing to adaptive and emotionally aware interactions.

- **Sentiment Analysis Module**: Employing fuzzy logic, extracts sentiment from the user's input. The fuzzifier processes the user's observation, deciding the appropriate emotion for the ECA to display. This emotional intelligence enhances the system's ability to respond appropriately, fostering a more engaging and empathetic interaction.

- **ECA Module**: Situated in the upper right corner, this module receives inputs from three key sources: the text-to-speech system, emotion-to-display, and lip sync schedule. The text-to-speech system generates audio synchronized with the lip sync schedule, creating a simulation that displays motions and produces audio with lip movements. This comprehensive module ensures a dynamic and interactive conversational loop with the ECA, enriching the user experience.

4.4 Proposed Architecture

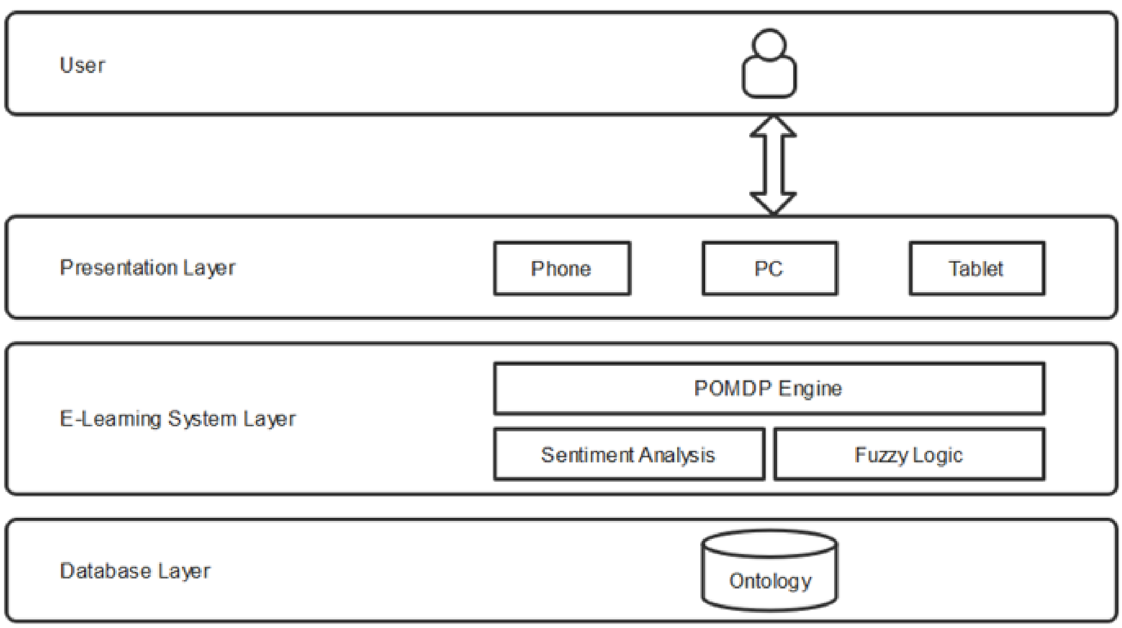


Fig. 4.4.1: General framework for the Proposed Architecture

Fig 4.4.1 shows the framework of the proposed architecture. It introduces a layered approach that strategically intertwines techniques for a robust e-learning system architecture, promising a transformative learning experience. The presentation layer is at the forefront of this framework, where users interact directly with the system. This layer serves as the interface for learners, facilitating a seamless and intuitive engagement with the educational content. Communication between the presentation and e-learning system layers is pivotal, ensuring the effective delivery of personalized learning materials. The e-learning system layer, positioned at the framework's core, is the backbone for content delivery, leveraging sophisticated algorithms and adaptive methodologies.

What sets this method apart is its incorporation of advanced sentiment analysis techniques, enabling the system to gauge and respond to the emotional nuances of user interactions. This emotional intelligence enriches the learning experience, allowing the system to tailor its responses based on the content and the user's emotional state. Such personalized and adaptive features hold immense potential for creating a more engaging and effective learning environment.

The database layer is integral to the entire process, where all data is stored and retrieved. This layer serves as the reservoir for user profiles, learning histories, and sentiment data, forming the basis for informed decision-making and continuous improvement. By seamlessly integrating sentiment analysis into the robust e-learning system architecture, this method strives to transcend traditional educational models, offering users a dynamic and personalized learning journey. In essence, it represents a step towards the future of e-tutoring, where technology not only imparts knowledge but also understands and responds to the human aspect of learning.

As elucidated in Fig 4.4.2, this thesis proposes an architecture that emerges as a sophisticated and intricately linked system, representing an approach to an intelligent tutoring system. The interconnectivity of its components underscores a seamless synergy aimed at providing an advanced and adaptive learning experience.

At the pinnacle of the system architecture lies the **Domain Mode**l in blue, serving as the domain's bedrock of knowledge and information. This model, housing the domain ontology, lays the foundation for all other components, ensuring a comprehensive understanding of the subject matter.

The **Student Model**, a central component on the right side of the architecture in yellow, encapsulates the system's learner understanding. BSH Storage meticulously tracks and stores the learner's progress, providing a rich source of historical data.

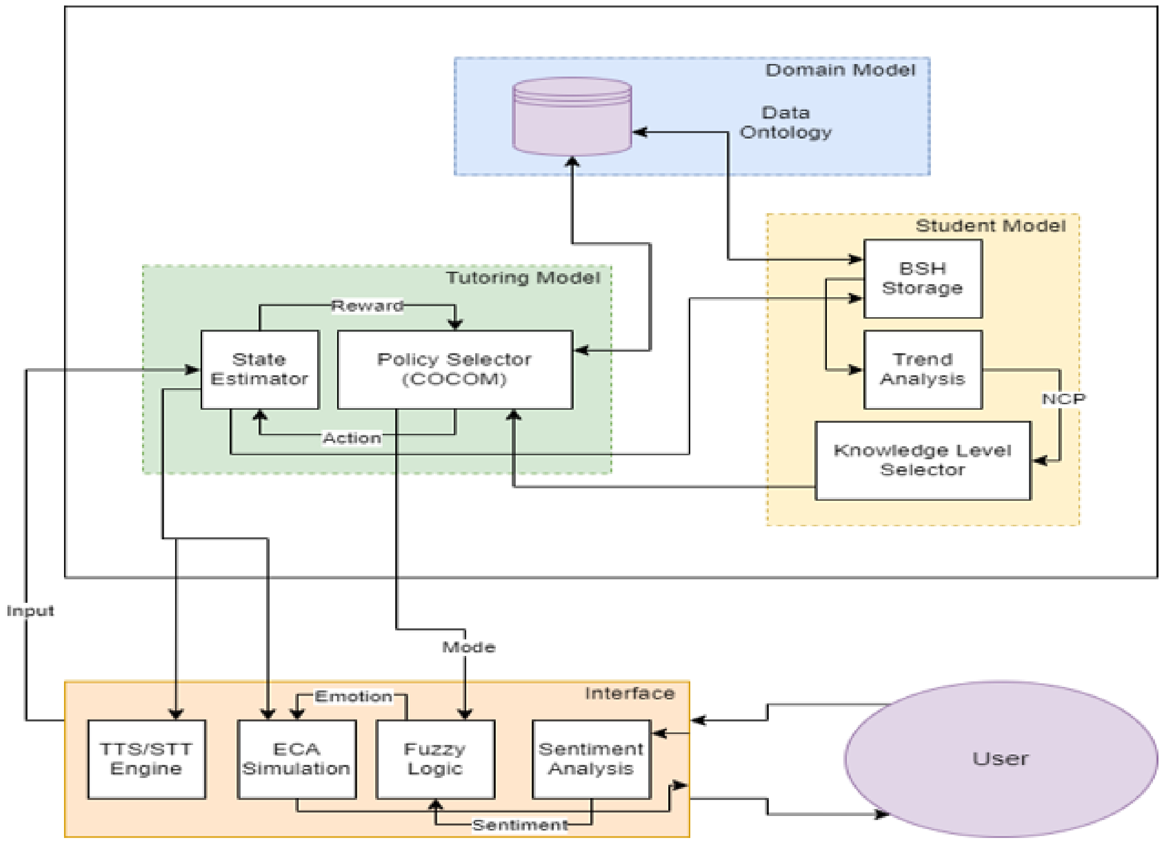


Fig. 4.4.2: Proposed Architecture

Integrated with Trend Analysis and Knowledge Level Selector, this model adapts to the student's learning pace, offering personalized feedback.

Functioning as the pedagogical core, the **Tutoring Mode**l, the green one, represents the tutor's knowledge, skills, and instructional abilities. Responsible for providing tailored guidance, this model incorporates the state estimator and policy selector. The state estimator utilizes student input to calculate the next system state, while the policy selector dynamically chooses teaching strategies based on the learner's knowledge level.

The **Interface** in the orange, designed for user interaction, is a user-friendly gateway to the system. It contains important modules used to run the system, like Text-To-Speech(TTS)/Speech-To-Text(STT) Engine, ECA Simulator, Fuzzy Logic, and Sentiment Analysis System. The User's interaction and feedback are pivotal in refining and enhancing the system's effectiveness.

At the heart of this architecture is the **User**, the individual interacting with the intelligent system for learning purposes. The User's engagement and feedback are integral to the system's continuous improvement. This user-centric design philosophy positions the learner at the forefront, emphasizing the importance of a dynamic and interactive learning experience.

Crucially, this envisioned architecture is poised to address the challenges identified in the problem statement. The absence of a standardized architecture hindering scalability and flexibility is mitigated through the integration of components. The geographical barriers limiting access are transcended by intelligent and adaptive features, ensuring a personalized learning experience irrespective of the learner's location. Incorporating voice interaction systems and robust adaptability mechanisms addresses operational challenges across diverse devices and platforms, enhancing user experiences.

To ensure seamless adaptability across diverse devices, our major objective is centralizing the system by deploying key components on the cloud platform transforming it into an SOA system. The Domain, Student, and Tutoring Model become independent services within this framework, enhancing flexibility and scalability. User interaction occurs through the Interface, acting as an intermediary between the user and services on Heroku, making the system device-agnostic. This centralized SOA approach ensures efficient communication and data exchange, facilitating intuitive user interaction across various devices. Aligned with contemporary trends, this deployment enhances scalability, flexibility, and adaptability, providing users with a versatile and user-friendly learning experience tailored to specific device configurations.

This architecture combines advanced analysis techniques with a robust system structure and aligns seamlessly with the proposed contributions. This architecture positions itself as a leap forward from the current standalone system by fostering adaptability, personalization, and user-centricity. The subsequent sections will delve into the algorithms and interface, offering a more comprehensive understanding of the system's inner workings.

4.4.1 Algorithm: Domain Model

The Algorithm for the Domain Model intricately orchestrates the intelligent tutoring system's utilization of domain-specific knowledge. At its core, the algorithm processes the domain ontology, a structured representation of knowledge within a specific subject area. It meticulously analyzes and contextualizes this information to tailor instructional content and formulate responses. The Domain Model algorithm plays a pivotal role in adapting the system's educational approach based on the user's interactions, ensuring a dynamic and personalized learning experience. Through a sophisticated interplay of data processing and decision-making, this algorithm embodies the system's ability to offer targeted and contextually relevant insights, thereby enhancing the efficacy of the intelligent tutoring system in delivering a comprehensive and adaptive e-learning environment.

Figure 4.4.3 illustrates the sequential steps of the Algorithm for the Domain Model.

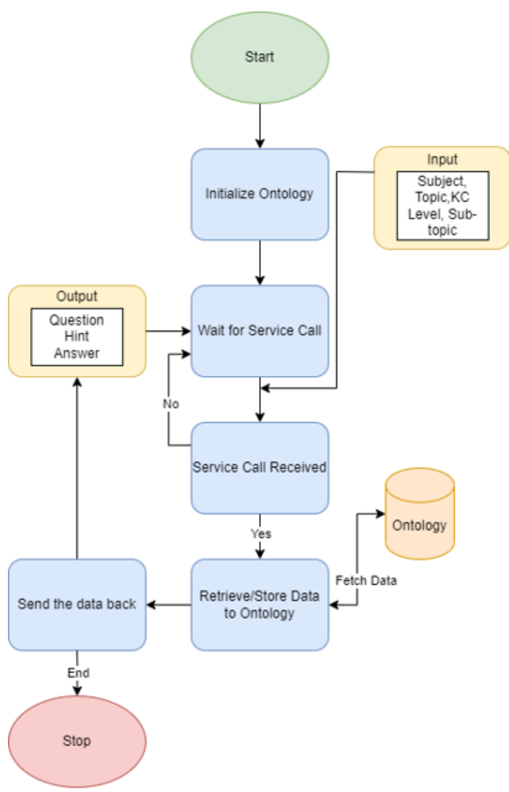


Fig. 4.4.3: Algorithm: Domain Model

Algorithm 4.4.1 Algorithm: Domain Model

Data: Ontology in Memory, Service Call (Subject, Topic, Knowledge Component Level, Subtopic)

Result: Question, Hint, Answer

Initialize the ontology in memory
Wait for the Service Call
Receive Service Call (Subject, Topic, Knowledge Component Level, Subtopic)
Query the answer from the ontology for the Subject, Topic, Knowledge Component Level, Subtopic
Send the Question, Hint, and Answer to the Tutoring Model

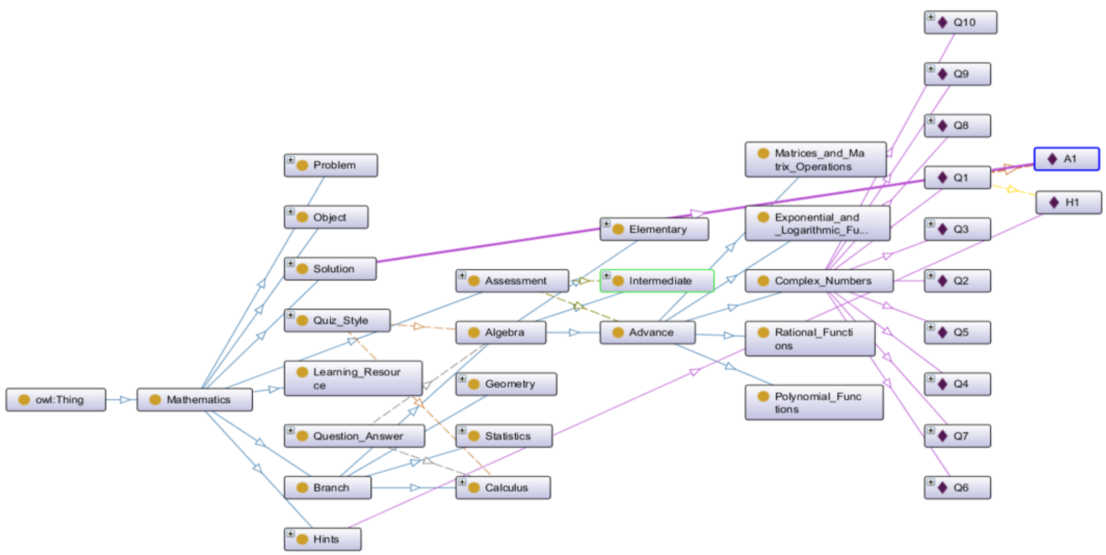


Fig. 4.4.4: Data Ontology

The flow, as depicted in Algorithm 4.4.1 and Fig 4.4.3, follows a structured sequence of steps. It commences by initializing the ontology in the system's memory, establishing a foundational knowledge base. The structure of the data ontology developed by A. Jamal [32] is shown in Fig 4.4.4. Subsequently, the algorithm enters a wait state, anticipating a service call. Upon receiving the service call containing parameters such as Subject, Topic, Knowledge Component Level, and Subtopic, the algorithm queries the ontology for the corresponding question, hint, and answer. This dynamic retrieval ensures that the educational content aligns precisely with the specified subject and contextual details. Finally, the algorithm sends the retrieved Question, Hint, and Answer to the Tutoring Model, facilitating the seamless flow of information within the intelligent tutoring system. This iterative process underlines the adaptability and responsiveness of the Domain Model, catering to the user's specific learning requirements based on the provided parameters.

4.4.1.1 Domain Model: Example

Upon the program's initiation, users are prompted to specify the subject, topic, subtopic, and KC Level. These inputs are from an ontology created by A. Jamal [32] and serve as criteria for fetching relevant data from the ontology. The algorithm then queries the ontology for the corresponding question, hint, and answer, as shown in Fig 4.4.1.

Table 4.4.1: Example Run for Domain Model

No.	**Subject**	**Topic**	**Subtopic**	**KC Level**	**Question**	**Answer**	**Hint**
1	Mathematics	Geometry	Euclidean geometry	Intermediate	What regular polygon has an exterior angle that measures 60 degrees?	Hexagon	Shape with 6 sides
2	Mathematics	Geometry	Euclidean geometry	Intermediate	How much paper is needed to cover a rectangular bulletin board that is 29 in. wide and 37 in. high?	1,073 square inches	Multiply the width and height
3	Mathematics	Geometry	Euclidean geometry	Advanced	Joe's garden is the shape of a hexagon. The measures of 5 of the angles are: 160°, 90°, 60°, 160°, and 80°. What is the measure of the remaining angle?	170	Add all the angles and subtract from 720

4.4.2 Algorithm: Student Model

The Algorithm for the Student Model serves as the cognitive backbone of the intelligent tutoring system, intricately managing and tracking the learner's progress. Operating with the Belief State History (BSH) Storage, Trend Analysis, and Knowledge Level Selector, this algorithm constantly evolves to encapsulate the user's knowledge state. It diligently captures the learner's interactions, incorporating personalized feedback based on their pace and understanding. Integrating BSH Storage ensures a comprehensive user progress record, facilitating adaptive responses and tailored guidance. The Student Model algorithm epitomizes the system's commitment to individualized learning experiences, dynamically adjusting instructional strategies to optimize the user's educational journey. This algorithm underscores the system's adaptability, providing nuanced insights into the learner's evolving knowledge landscape and contributing to an interactive and responsive e-learning environment.

Figure 4.4.5 carefully illustrates the sequential steps of the Algorithm for the Student Model.

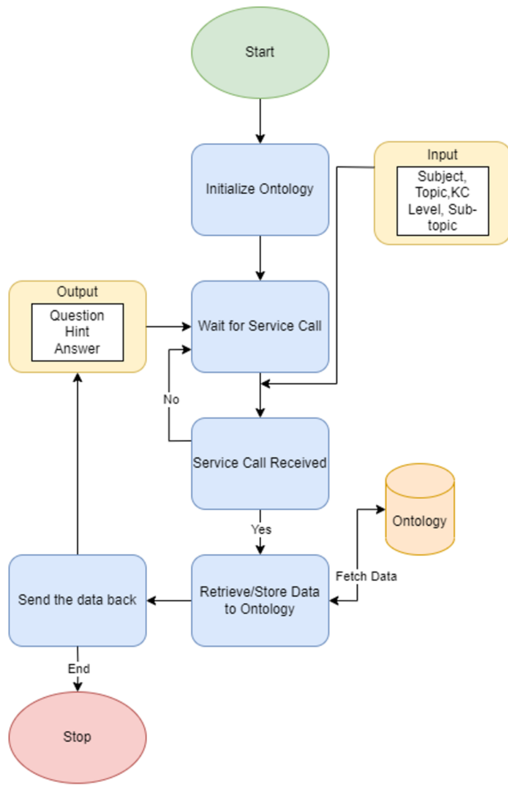


Fig. 4.4.5: Algorithm: Student Model

Algorithm 4.4.2 Student Model Algorithm

Data: BSH (Belief State History), Knowledge Component Level
Result: Updated Knowledge Component Level, NCP (Number of Change Points)

Initialize the BSH and Knowledge Component Level
Wait for the Service Call
Receive Service Calls with states from the state estimator
Store the states in the BSH and perform trend analysis
Get NCP and update the Knowledge Level
Send the updated data back to the Tutoring Model

The Student Model Algorithm, as encapsulated in Algorithm 4.4.2, functions as the intellectual hub of the intelligent tutoring system, orchestrating the assimilation and evolution of the learner's knowledge. It begins by initializing the Belief State History (BSH) and the Knowledge Component Level, creating a foundation for tracking the learner's progress. Upon receiving a service call, the algorithm stores the states from the state estimator in the BSH, facilitating trend analysis. The extraction of NCP and subsequent Knowledge Level updates showcase the algorithm's adaptability and responsiveness to the user's evolving comprehension. Ultimately, the algorithm communicates the updated data to the Tutoring Model, perpetuating a dynamic loop that refines instructional strategies based on the learner's interaction history. This intricate interplay of data processing and decision-making epitomizes the Student Model's role in tailoring the learning experience, offering a nuanced and personalized educational journey within the intelligent tutoring system.

4.4.2.1 Student Model: Example

The Student Model Algorithm, as depicted in Fig 4.4.5, is executed upon receiving a service call from the Tutoring Model. The algorithm then updates the states in the BSH and performs trend analysis to extract the NCP. This information is leveraged to update the Knowledge Component Level, as shown in Table 4.4.2.

Table 4.4.2: Example Run for Student Model

No.	**State**	**isCorrect?**	**NCP**	**COCOM Mode**
1	Beginner:0.6 Intermediate: 0.4 Advanced: 0.5	False	5	Strategic
2	Beginner: 0.2 Intermediate: 0.7 Advanced: 0.5	True	5	Tactical
3	Beginner: 0.1 Intermediate: 0.2 Advanced: 0.75	True	3	Opportunistic

4.4.3 Algorithm: Tutoring Model

The Algorithm for the Tutoring Model is the guiding force within the intelligent tutoring system, orchestrating the delivery of instructional guidance. At its core, this algorithm encapsulates the tutor's knowledge, skills, and abilities, ensuring a seamless interaction with the learner. Integrated with the State Estimator and Policy Selector modules, the Tutoring Model dynamically evaluates the user's input, determines the system's next state, and adapts teaching strategies accordingly. Leveraging the Policy Selector tailors its responses based on the learner's knowledge level, fostering an adaptive and effective tutoring experience. This algorithm embodies the system's commitment to providing targeted and contextually relevant guidance, enhancing the overall efficacy of the intelligent tutoring system. Through the intricate interplay of data processing and decision-making, the Tutoring Model algorithm creates a dynamic, responsive, and personalized e-learning environment, fostering optimal learning outcomes for the user.

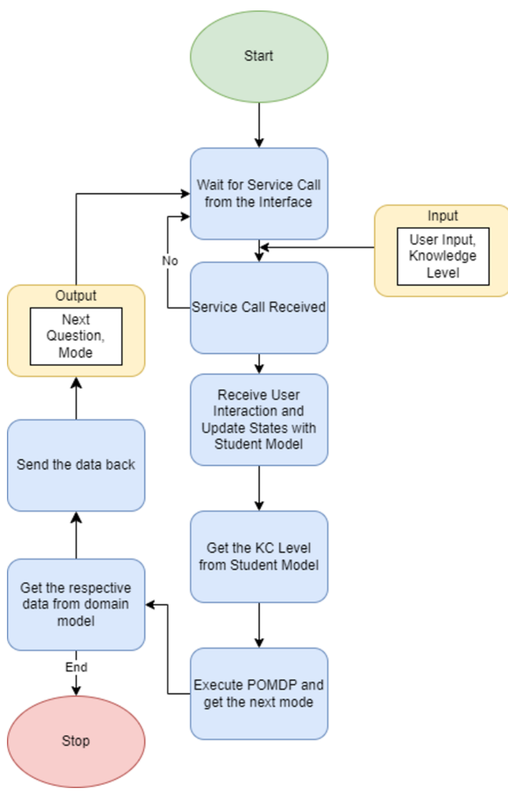


Fig. 4.4.6: Algorithm: Tutoring Model

Algorithm 4.4.3 Algorithm: Tutor Model

Data: User Input, Knowledge Level
Result: Question, Mode

Wait for a Service Call from the Interface
Receive Service Calls with User Input and Knowledge Level
Update States in the State Estimator and send to Student Model for Knowledge Component Level
Receive Knowledge Component Level from Student Model
Execute POMDP using the received Knowledge Component Level and generate Mode
Use Mode to get the respective Question from the Domain Model
Send the Question and Mode back to the Interface

The Tutoring Model Algorithm shown in Fig 4.4.6, outlined in Algorithm 4.4.3, serves as the orchestrator of the intelligent tutoring system, ensuring a dynamic and personalized learning experience. Upon receiving a service call from the interface, the algorithm processes user input and knowledge level. It then updates the states in the State Estimator and communicates with the Student Model to obtain the Knowledge Component Level. Leveraging this information, the algorithm executes the Partially Observable Markov Decision Process (POMDP) to generate a mode that influences the system's response. This mode is instrumental in fetching a contextually relevant question from the Domain Model. The resulting Question, answer, hint and Mode are then transmitted back to the interface, completing a cohesive interaction loop. This iterative process encapsulates the system's adaptability as it dynamically adjusts its instructional strategy based on user input and knowledge levels, fostering an engaging and tailored educational journey.

4.4.3.1 Tutoring Model: Example

As depicted in Fig 4.4.6, the Tutoring Model Algorithm is executed upon receiving a service call from the Interface. The algorithm then updates the states in the State Estimator and communicates with the Student Model to obtain the Knowledge Component Level. Leveraging this information, the algorithm executes the Partially. Observable Markov Decision Process (POMDP) to generate a mode that influences

Table 4.4.3: Example Run for Tutoring Model

No	**User Input**	**KC Level**	**Student Model Response**	**POMDP Action**	**Domain Model Response**
1		Intermediate	Strategic	Intermediate	What regular polygon has an exterior angle that measures 60 degrees?
2	Triangle	Intermediate	Strategic	Intermediate	How much paper is needed to cover a rectangular bulletin board that is 29 in. wide and 37 in. high?
3	1,073 square inches	Intermediate	Tactical	Advanced	Joe's garden is the shape of a hexagon. The measures of 5 of the angles are: 160°, 90°, 60°, 160°, and 80°. What is the measure of the remaining angle?

the system's response. This mode is instrumental in fetching a contextually relevant question from the Domain Model. The resulting Question, answer, hint and Mode are then transmitted back to the interface, completing a cohesive interaction loop. This iterative process encapsulates the system's adaptability as it dynamically adjusts its instructional strategy based on user input and knowledge levels, fostering an engaging and tailored educational journey.

4.4.4 Algorithm: Interface

The Interface serves as the gateway to the intelligent tutoring system, shaping the user's interaction and experience. Designed intuitive and user-friendly, the Interface Algorithm, as detailed in Algorithm 4.4.4, orchestrates a seamless exchange between the user and the system. By discerning the user's device type and preferences, the interface tailors its presentation, offering a range of options that accommodate device constraints while aligning with individual priorities. This adaptive approach ensures that users, regardless of their technological environment, encounter an interface that optimally caters to their unique needs according to the device type. The Interface Algorithm sets the stage for a dynamic and engaging learning journey within the intelligent tutoring system through a well-defined initiation process, device identification, preference gathering, and connection establishment with the Tutoring Model.

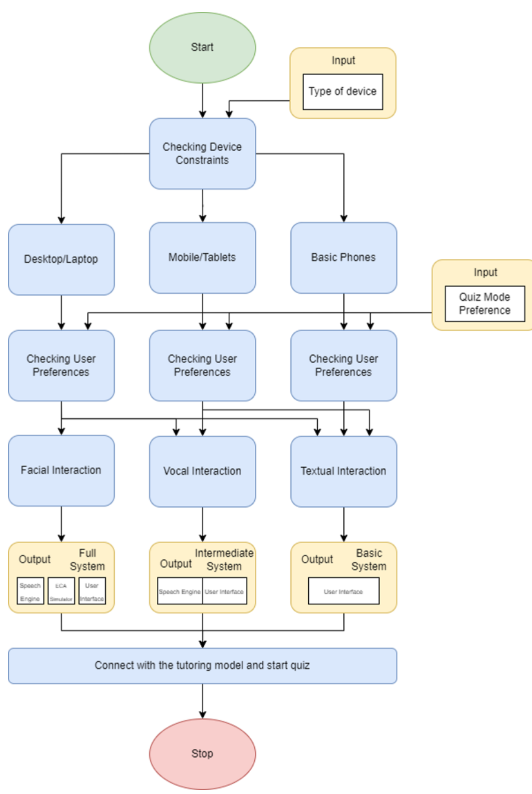


Fig. 4.4.7: Algorithm: Interface

It commences by initializing the Interface System, marking the initiation of the

Algorithm 4.4.4 Algorithm: Interface

Data: User Device Type, User Preferences
Result: Interface Presentation, Question

Initiate the Interface System
Determine User's Device Type
Take User's Preferences as Input based on the Identified Device Type
Present Interface to the User with Options for Full System, Intermediate System, or Basic System, depending on Device Constraints and User Preferences
Establish a Connection between the Interface and Tutoring Model to Retrieve the First Question

user interface. Subsequently, the algorithm dynamically determines the User's Device Type, enabling the system to adapt its presentation to varying device configurations. User Preferences are then gathered and influenced by the identified Device Type, allowing for a tailored and user-centric experience. The Interface System presents the user with options, offering a choice between a Full System, an Intermediate System, or a Basic System. This dynamic presentation considers device constraints and user preferences, ensuring an optimal and personalized interface.

For users on a computer or laptop, the system offers the flexibility to run all three modes: facial, voice, and text, providing a comprehensive and immersive experience. If the user is on a mobile phone, the algorithm adjusts to allow the execution of voice and text modes, optimizing the device's capabilities. The system gracefully adapts to offer a text-only mode for users on basic machines, ensuring accessibility and functionality even in limited technological environments. This device-specific adaptation enhances the user's engagement by aligning the system's features with the capabilities of their chosen device, fostering a seamless and personalized learning interaction.

A pivotal aspect of the algorithm involves establishing a robust connection between the Interface and the Tutoring Model. This connection facilitates the retrieval of the Question data, marking the initiation of the interactive and educational experience for the user. Overall, this algorithm underscores the system's adaptability, user-friendliness, and capacity to deliver a tailored learning journey based on individual preferences and device capabilities while adapting to geographical barriers.

The culmination of the “Problem Statement and Proposed Method” chapter establishes a foundational understanding of the challenges addressed and the innovative approach proposed. The ensuing chapter, “Implementation and Tests”, embarks on the practical realization of the proposed method. It delves into the intricacies of system development, providing insights into the implementation process and rigorous testing methodologies. As we transition from conceptualization to application, the forthcoming chapter unravels the tangible aspects of the intelligent tutoring system, shedding light on its operational framework and validating its efficacy through comprehensive testing protocols.

CHAPTER 5

Implementation and Tests

5.1 Software Tools and Services

Table 5.1.1 overviews the software tools and services for implementing the ECA tutoring system.

Table 5.1.1: Software Tools and Services

Function	**Software**
Programming Language – Service	Python – 3.10.9
Framework – Desktop Application	PyQT5 (Python)
Framework – Mobile and Web Application	Flutter (Dart)
Text-to-Speech	Google Text-To-Speech API
Speech-to-Text	speech_recognition
Backend Service Framework	Flask
Code Editor	VS Code
Environment Manager	Anaconda
Interaction Medium	.json
Deployment Platform	Heroku

The programming language selected for service development is Python 3.10.9, while the desktop application framework is PyQT5 (Python). We are using Flutter for mobile and web frameworks. Text-to-speech functionality is achieved through the

Google Text-To-Speech API, and Speech-to-Text is implemented using the speech recognition library. The backend service framework is Flask, and the Visual Studio Code editor facilitates coding activities. Anaconda serves as the environment manager. Interaction between components is structured using the .json format, and the deployment platform chosen is Heroku. This comprehensive selection of tools and services forms the technological foundation of the ECA Tutoring System, ensuring a robust and versatile implementation across various applications and functionalities.

5.2 Overall Architecture

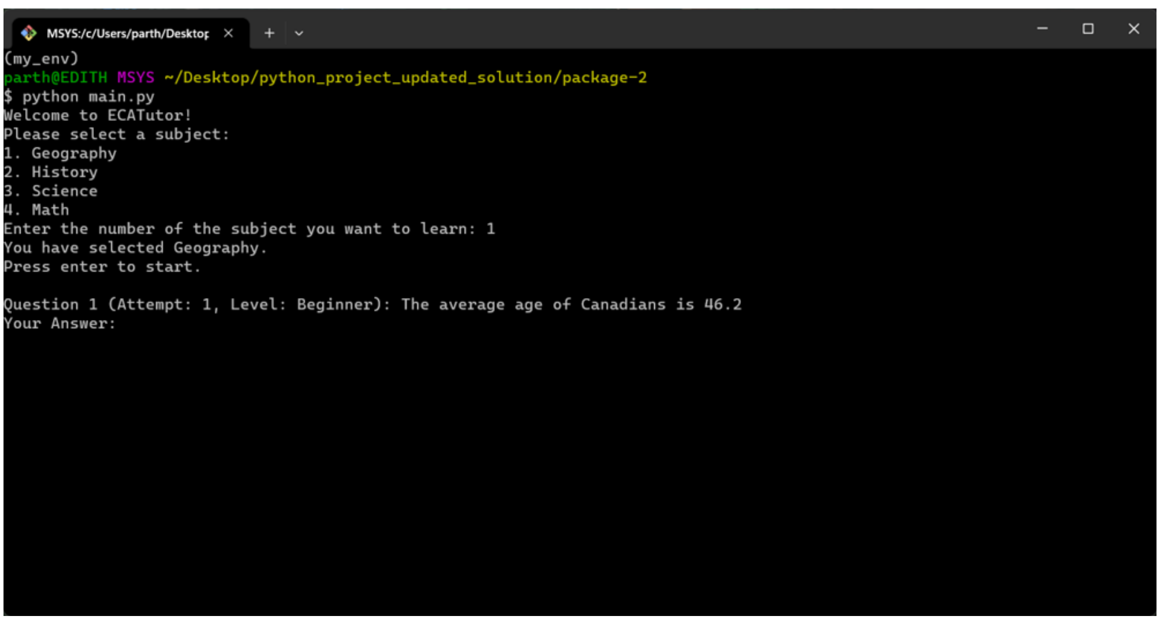


Fig. 5.2.1: Command Line Based System

The transformation from the previous version of the tutoring system to the current iteration signifies a substantial advancement in accessibility, usability, and overall architecture.

In the earlier version, the tutoring system functioned as a standalone application exclusive to the Windows platform, utilizing a command-line interface with a unique avatar system, as seen in Fig 5.2.1 and Fig 5.2.2. However, limitations such as platform exclusivity, manual environment management, extensive dependency in-

Fig. 5.2.2: Initial Avatar System

stallations,and local storage requirements constrained its widespread use.

The evolution of the new tutoring system marks a paradigm shift, introducing a device-independent architecture that liberates users from platform constraints. Unlike its predecessor, the current system deploys its core logic on a server, ensuring compatibility with various devices. This inclusivity extends beyond traditional PCs and laptops to encompass mobile phones, facilitating a versatile learning experience.

One of the most notable enhancements is the introduction of an interactive user interface, departing from the command-line interface of the previous version. This change caters to a broader user base, making the system more user-friendly and accessible, especially for individuals unfamiliar with command-line operations.

Regarding avatar representation, the new system introduces a revamped avatar designed within our research group [39]. This update brings a fresh and improved visual identity to the tutoring system, contributing to a more engaging and personalized learning environment.

Furthermore, the new architecture streamlines user responsibilities, requiring only the user interface installation. The complexities of environment management, dependency installations, and storage requirements are now seamlessly handled at the cloud level. This streamlined approach simplifies the user experience and ensures efficient resource utilization and easier system maintenance.

Critically, the new tutoring system's compatibility with various devices, including PCs, desktops, laptops, and mobile phones, underscores its adaptability to modern learning preferences. This flexibility empowers users to engage with the system on their preferred devices, providing a dynamic and personalized educational journey.

Moreover, the new system's architecture facilitates usage from anywhere, breaking down geographical barriers and eliminating dependence on specific carriers. This flexibility aligns with the contemporary need for remote and decentralized learning experiences, allowing users to access educational resources regardless of physical location.

In summary, transitioning from the previous version to the current iteration of the tutoring system represents a comprehensive enhancement in accessibility, usability, and architectural efficiency. The shift towards a device-independent, cloud-supported model, an interactive user interface and a redesigned avatar system positions the new tutoring system as a versatile, user-centric, and globally accessible educational tool.

5.3 Software Adaptability

The upcoming enhancements to the tutoring system will introduce a tailored user experience across a diverse range of devices. Specifically, a PyQT application is being developed for PCs and laptops, offering users a rich and seamless interface on these platforms. Simultaneously, we create a solution with optimal functionality and accessibility using Flutter to cater to mobile phones and basic devices.

Recognizing the diverse and dynamic situations that students may find themselves in, the tutoring system prioritizes adaptability and user comfort. If a student encounters a scenario where one mode of interaction is impractical or uncomfortable, the system offers the flexibility to switch to an alternative interaction method seamlessly. For instance, if students are in a noisy environment where vocal interaction becomes challenging, they can effortlessly transition to textual interaction, ensuring uninterrupted learning. Similarly, if a student is in a private space where speaking aloud might be convenient, the facial interaction mode can be an effective alternative. This emphasis on flexibility ensures that users can choose the interaction mode that best aligns with their immediate circumstances, fostering a user-centric and accommodating learning experience. Whether it is a preference for vocal engagement, the convenience of textual input, or the subtlety of facial expressions, the system empowers users to navigate their learning journey easily, irrespective of their surroundings or constraints.

This strategic development approach empowers users with the flexibility to choose their preferred mode of interaction based on the configuration of their device. The result will be three distinct solutions with specific exchange methods, as stated in Table 5.3.1.

Table 5.3.1: Software Adaptability

Version Type	**Device Configuration**	**User Preference**		
		Facial	**Vocal**	**Textual**
Full	Windows/Mac/Linux	Yes	Yes	Yes
Intermediate	Android/iOS	No	Yes	Yes
Basic		No	No	Yes

The system's overall architecture will remain unchanged, with modifications limited to the interface level, as depicted in Figure 5.3.1.

5.3.1 Full Version

A feature-rich PyQT application like Fig 5.3.2 is designed for personal computers, laptops, and desktops. Users on these devices will benefit from a comprehensive and immersive learning experience.

All personal computers, laptops, and desktops with specific operating systems will be compatible with all three modes of interacting with the tutoring system.

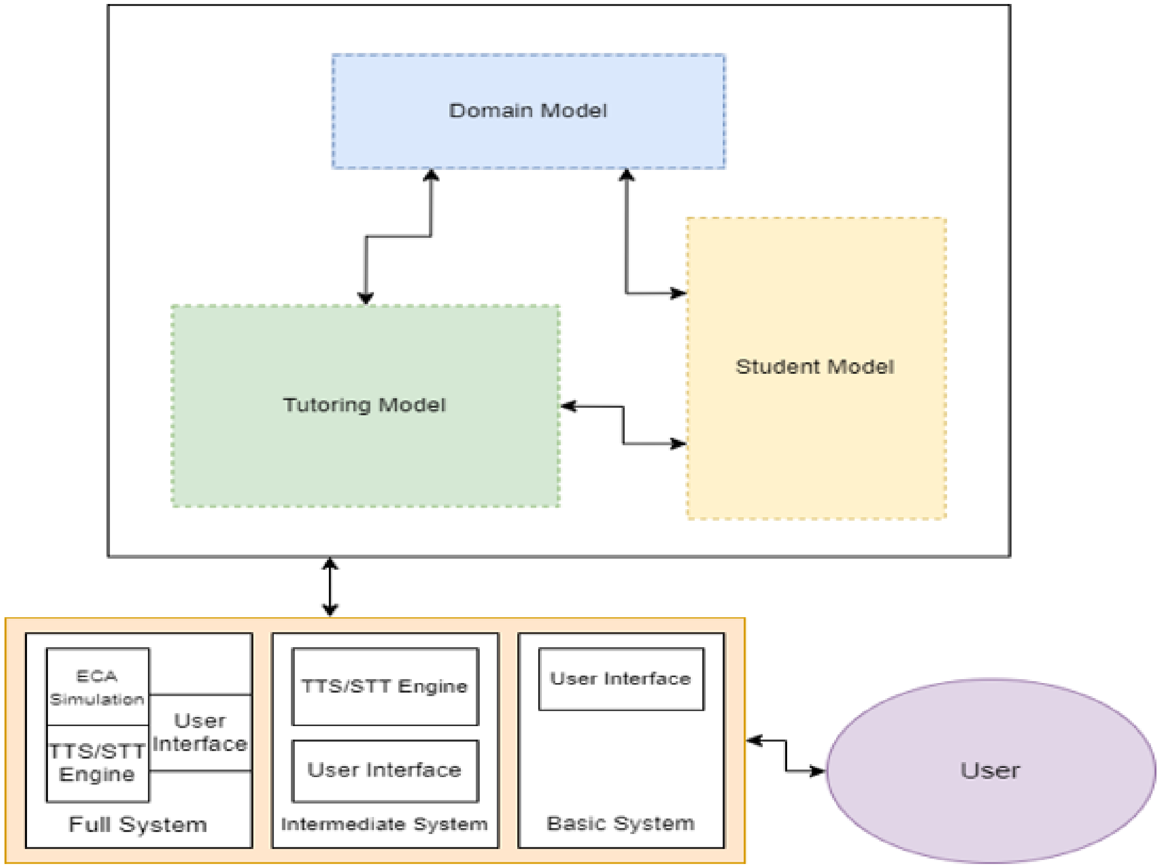
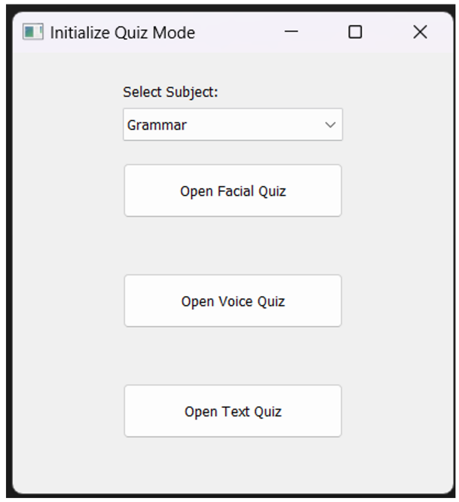


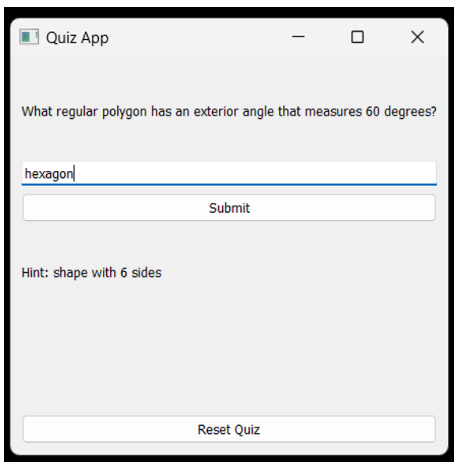
Fig. 5.3.1: Overall Architecture

Upon launching the full version, users will encounter a screen like Fig 5.3.2a, welcoming interface featuring three distinct interaction modes. Opting for the Facial Mode triggers the simultaneous activation of the speech engine and ECA simulator [39] as seen in Fig 5.3.2c, and user interface 5.3.2b. Meanwhile, selecting Vocal Mode initiates the speech engine and user interface, while Textual Mode exclusively runs the user interface.

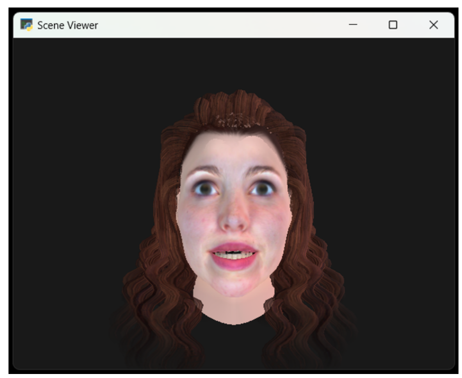
Following the mode selection, a seamless connection with the tutoring service is established, prompting the system to present the user with the question. The avatar dynamically animates in response to the questions, creating an engaging visual experience. Users can then respond using either the keyboard for text input or the microphone for vocal interaction. This comprehensive approach ensures that the full version operates harmoniously, catering to various user preferences and facilitating an interactive learning environment.



(a) Quiz Mode Page



(b) User Interface



(c) Embodied Conversational Agent

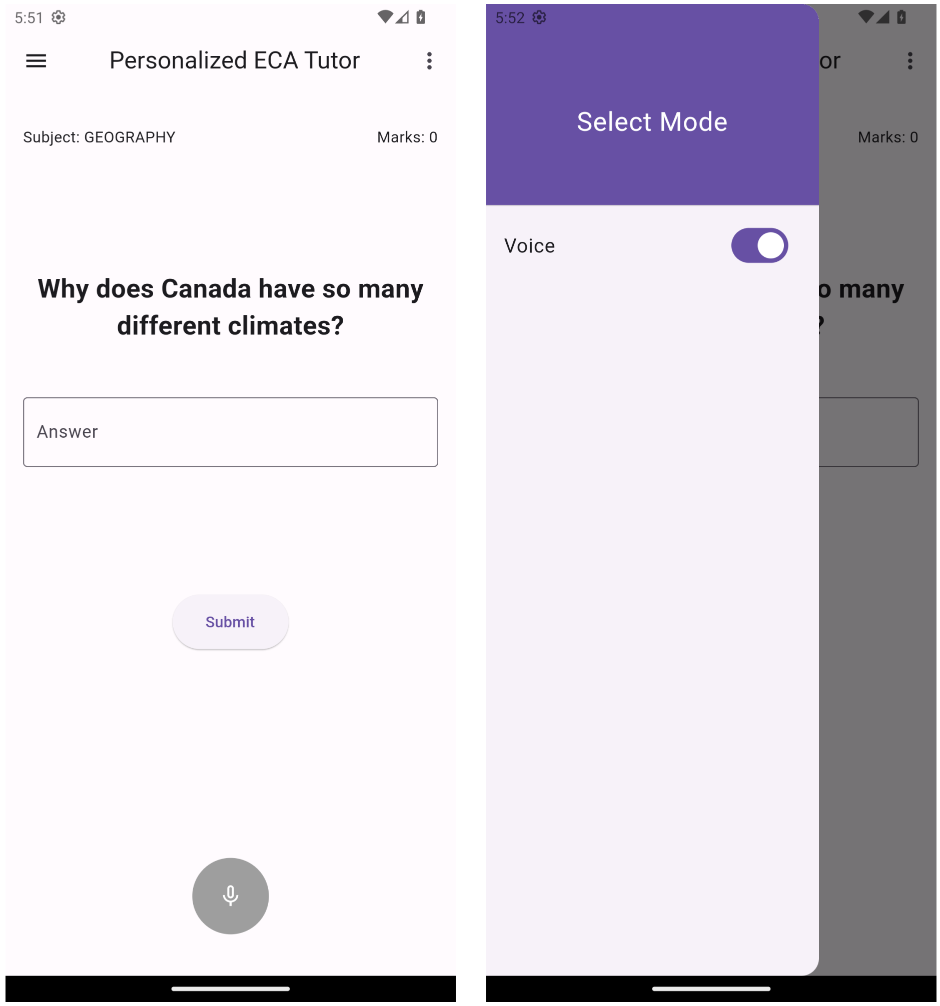
Fig. 5.3.2: Full Version

5.3.2 Intermediate Version

A Flutter app optimized for mobile phones, ensuring a smooth and responsive interface. This version as shown in Fig 5.3.3, will support both vocal and textual interactions, providing users with versatility in engagement.

Mobile phones running on Android and iOS will support both vocal and textual interactions, enhancing the user experience.

In the Intermediate Version, users will encounter a simplified yet effective interface with two interaction modes: Vocal and Textual Mode. Opting for Vocal Mode activates the speech engine and user interface, enabling users to engage with the sys-



(a) Quiz Interface (b) Speech Engine Toggle

Fig. 5.3.3: Intermediate Version

tem through vocal interactions. Conversely, Textual Mode exclusively runs the user interface shown in Fig 5.3.3a, providing users with a text-based interaction experience.

Once the interaction mode is selected, the system seamlessly connects with the tutoring service, presenting users with the relevant questions. Users can then comfortably respond using the keyboard for textual input or the microphone for vocal interactions utilizing the speech engine toggle, as seen in Fig 5.3.3b. This streamlined approach, omitting the Facial Mode, simplifies the user experience while maintaining the essential elements of interaction, making the Intermediate Version a versatile, adaptable and accessible option for various users and devices.

5.3.3 Basic Version

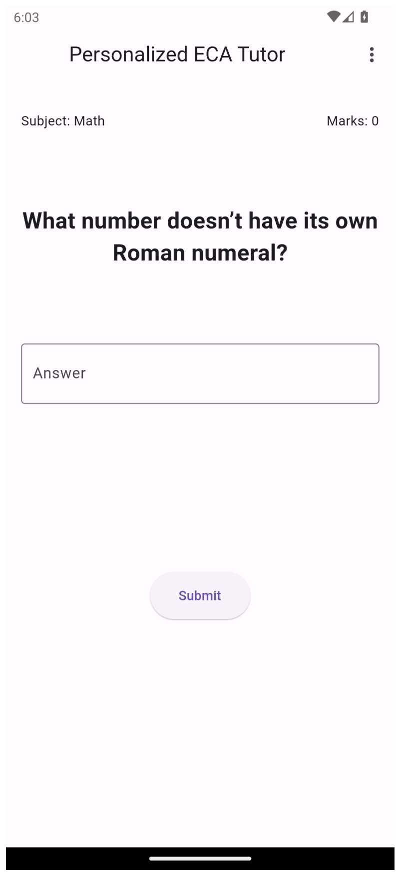


Fig. 5.3.4: Basic Version

Tailored for devices without vocal interaction capabilities, this version of the Flutter app will enable users to engage with the tutoring system through textual interactions. This ensures inclusivity for a broad spectrum of devices.

This version will have no vocal interaction capabilities; users can only interact with the system through textual input. This version will be compatible with all devices, including basic mobile phones and devices without vocal interaction capabilities. It is shown in Fig 5.3.4 that the only mode of interaction is textual.

For devices lacking vocal interaction capabilities, the textual interaction mode ensures accessibility, ensuring that users can engage with the system regardless of their device's capabilities.

5.4 Software Flexibility

Table 5.4.1: Variable Ontologies

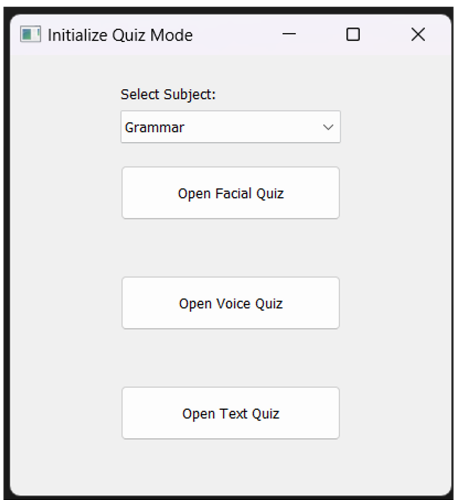
Subject	**Question**	**Answer**
Grammar	I have __ apple in my bag	an
History	What are the official languages of Canada?	English and French are the official languages of Canada
Science	Which planet is the biggest planet in our solar system	Jupiter
Math	What is the perimeter of a circle called?	Circumference
Geography	Which part of Canada is hottest?	Victoria is the warmest place in Canada, with an average annual temperature of 9.9°c

5.4.1 Variable Ontologies

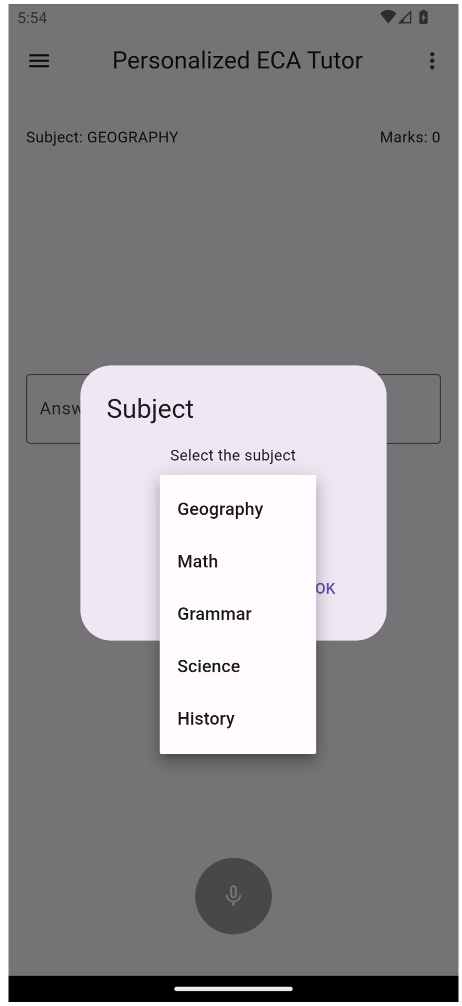
Adopting Service-Oriented Architecture (SOA) within the tutoring system heralds an approach—Variable Ontologies. This innovative concept dynamically empowers users to choose their preferred subject, shaping the tutoring experience according to individual learning needs. The system seamlessly interacts with distinct domain models corresponding to Grammar ontology created by Vyas, Niyati [66], Geography, Science, and History ontologies created by Jawahar, Ashwitha [33] , and Math ontology by Jamal, Asim [32], as mentioned in Table 5.4.1.

Users can select their preferred subject from a list of available ontologies, as seen in Fig 5.4.1. This selection triggers a seamless connection with the relevant domain model, ensuring a personalized and contextually relevant learning experience. This adaptability ensures that the tutoring system resonates with individual users, fostering an effective and engaging educational journey.

SOA principles provide the foundation for the tutoring system's adaptability to diverse subject matters without requiring extensive modifications. By dynamically



(a) Ontology Selector - Full Version



(b) Ontology Selector - Intermediate and Basic Version

Fig. 5.4.1: Variable Ontology Support

connecting to the relevant domain model service based on user-selected subjects, the system delivers a personalized and contextually relevant learning experience. This flexible design allows developers to integrate or remove services effortlessly, ensuring the system remains agile and easily extensible.

5.4.2 Variable Tutoring Models

The Variable Tutoring Models feature enriches user engagement and accommodates diverse learning preferences within the tutoring system. Users now enjoy the flexibility to respond to questions in ways that align with their comfort and understanding. The system supports two distinct modes of interaction based on the domain model selected:

1. **One-Word Response Mode**: Optimal for quick and focused interactions, users can choose this mode for concise one-word responses. It is ideal for reinforcing vocabulary or testing knowledge retention with brief answers.

2. **Sentence Response Mode**: This mode encourages users to respond with complete sentences, fostering a more comprehensive understanding. It provides an opportunity for elaboration, allowing users to express thoughts and showcase a deeper grasp of the subject.

Table 5.4.1 shows how users can respond differently to a question based on the ontology selected. The Variable Tutoring Models feature caters to a spectrum of learning styles and preferences. Whether users prefer succinct responses or detailed explanations, this adaptability ensures a tailored and dynamic learning experience. The versatility of the tutoring system aligns with the overarching goal of resonating with individual users, fostering an effective and engaging educational journey.

5.5 Tests and Results

The evolution from a command-line system to a user-friendly graphical interface marks a substantial improvement in the usability of the proposed approach. This transition empowers users to navigate and utilize the system effortlessly, eliminating the prerequisite of command-line proficiency. The core engine is strategically deployed on the Heroku Platform as a Service, liberating users from the burden of installing complex Python environments on their local systems.

Notably, the system's impact transcends traditional PC boundaries, extending its accessibility through dedicated Android and iOS applications. This adaptability ensures that a broad spectrum of devices, from desktop computers to mobile phones, can seamlessly engage with the system. Unifying services and establishing a well-defined communication medium enhance the system's versatility. The proposed system heralds a new era of user-friendly, device-independent, and interconnected educational tools by fostering seamless interaction with various devices and services. This holistic approach prioritizes user convenience and promotes a collaborative and integrated learning experience in diverse technological landscapes.

This multi-version approach aims to provide a versatile and inclusive learning experience, allowing users to interact seamlessly with the tutoring system on their chosen device. Whether on a PC, laptop, or mobile phone, users can access the tutoring system in a manner that best suits their preferences and device configurations.

CHAPTER 6

Evaluation and Discussions

6.1 Evaluation of Contributions

This research stands as a force in the realm of e-tutoring systems, making substantial contributions that redefine the landscape:

- **Standardized Architecture**: One of the fundamental advancements lies in addressing the absence of a standardized architecture in e-tutoring systems. By introducing a standardized architecture, this research significantly amplifies the scalability, flexibility, and interoperability of e-tutoring systems. The new approach ensures a cohesive and seamless learning experience for users.

- **Accessibility**: Overcoming geographical barriers is a pivotal achievement of this research. By mitigating accessibility challenges, the system enhances the reach of educational opportunities, making learning accessible to students irrespective of their location. This contribution aligns with the broader goal of fostering inclusivity in education.

- **Voice Interaction System**: The incorporation of a voice interaction system marks a paradigm shift in communication within e-tutoring systems. This innovative addition enhances natural and effective communication, creating a more immersive and engaging learning environment. The voice interaction system elevates the user experience, making learning interactions more dynamic and responsive.

- **Software Adaptability**: Recognizing the crucial need for software adaptability, this research ensures that learners can effortlessly access the tutoring system through a graphical user interface, transcending the constraints of device specificity. The system's adaptability is a testament to its commitment to providing a user-friendly and universally accessible platform.

Moreover, this work aligns with critical success factors outlined in [9], emphasizing user-friendly design and technical competence. The solution meets and elevates essential criteria by incorporating these factors, enhancing the mobile learning experience.

Fulfilling e-learning application guidelines proposed in [17], the system ensures accessibility and adaptability across various platforms and devices. Incorporating personalized feedback with dialogue communication adds individualized support, further enriching the learning journey. In essence, the contributions of this research extend beyond the technical realm, actively shaping a more inclusive, adaptive, and user-centric e-tutoring ecosystem.

6.2 Analysis and Discussions

Adopting SOA in this project brings forth myriad benefits, aligning with insights mentioned by Aung in [12]. The strategic use of SOA injects efficiency and agility into the system, fostering a robust foundation for the e-tutoring platform. Here, we delve into the advantages realized through the application of SOA principles:

- **Cost Efficiency**: SOA serves as a catalyst for cost efficiency by optimizing resource utilization. The system minimizes redundancy and promotes a lean operational model through modular and reusable services. This optimization translates into reduced operating expenses, a pivotal advantage in the context of the e-tutoring system.

- **Streamlined Business Process**: SOA technologies are pivotal in simplifying workflows and enhancing operational efficiency. The system becomes modular by breaking down complex processes into modular services. This streamlined approach improves the e-tutoring system's overall efficiency and contributes to a more agile and responsive operational environment.

- **Increased Flexibility**: The SOA solutions integrated into the project offer unparalleled flexibility. This flexibility empowers the e-tutoring system to adapt to changing requirements swiftly. The modular nature of SOA allows for quick adjustments, ensuring that the system remains dynamic and responsive to the evolving needs of users.

- **Improved Scalability**: Scalability is a cornerstone of SOA, and its integration into the project significantly enhances the system's ability to handle growing workloads effortlessly. As user demand increases, the modular services can be scaled independently, ensuring a seamless and responsive user experience even under heightened usage scenarios.

- **Rapid Development**: SOA accelerates development cycles, reducing the time market for new solutions. The modular and independent nature of services facilitates concurrent development, enabling faster iteration and deployment of new features. This rapid development cycle is instrumental in keeping the e-tutoring platform at the forefront of innovation.

In conclusion, applying SOA principles has a transformative impact on the e-tutoring system. The advantages of cost efficiency, streamlined processes, increased flexibility, improved scalability, and rapid development collectively contribute to a robust and adaptive platform. SOA emerges as a technical framework and a strategic enabler, shaping the e-tutoring system into a dynamic and efficient educational ecosystem.

6.3 Limitations

While the e-tutoring system embodies numerous strengths, it is essential to acknowledge certain limitations inherent in its design and implementation. These limitations, while not diminishing the project's overall contributions, provide areas for potential future enhancements:

- **Internet Dependency**: The current architecture relies on a robust internet connection for optimal functionality. Users with limited internet access may need help utilizing the system seamlessly. Future iterations could explore offline capabilities to address this limitation.
- **Learning Content Diversity**: The tutoring system focuses on Grammar, Geography, Science, History, and Math. Expanding the array of subjects and learning content would contribute to a more comprehensive educational experience.

As with any innovative project, these limitations serve as opportunities for future refinement and development. Recognizing these aspects allows for a more nuanced understanding of the system's current scope and guides future iterations toward a more inclusive, adaptive, and user-friendly e-tutoring experience.

CHAPTER 7

Conclusion

7.1 Research Summary

In conclusion, this research marks a stride in e-tutoring systems by introducing a standardized architecture rooted in Service-Oriented Architecture (SOA). The adoption of SOA brings a paradigm shift, enhancing the scalability, flexibility, and interoperability of e-tutoring systems. The voice interaction system incorporated in this work is a testament to our commitment to effective and natural communication within the e-tutoring environment.

The emphasis on accessibility has propelled the project beyond geographical barriers, opening up educational opportunities for students worldwide. The software's adaptability, demonstrated through the graphical user interface and device independence, ensures that learners can seamlessly access the system, fostering a truly inclusive learning experience.

This research aligns with critical success factors such as user-friendly design and technical competence, reinforcing the commitment to delivering a solution that meets essential criteria and elevates the standards for an enhanced mobile learning experience. The adherence to e-learning application guidelines further solidifies the project's contribution by making the system accessible and adaptable across various platforms and devices.

7.2 Future Work

Looking ahead, the e-tutoring system presents numerous avenues for future exploration and refinement. The following areas offer promising directions for future work:

1. **Offline Capabilities**: Enhancing the system's offline capabilities would mitigate challenges associated with internet dependency, ensuring continuous accessibility in diverse connectivity scenarios.

2. **Subject Diversification**: Expanding the range of subjects and learning content accommodates a broader audience, catering to varied educational needs and preferences.

3. **Public Acceptance and Adoption**: Future work could explore strategies to enhance the public's acceptance and adoption of e-learning systems, particularly personalized tutoring systems, in educational practices. This involves exploring ways to bridge the gap between traditional educational methods and innovative e-learning approaches, considering cultural perceptions, societal norms, and institutional readiness. Understanding and addressing these aspects would contribute to the seamless integration of e-learning into mainstream education, fostering widespread acceptance and utilization.

This research lays a robust foundation for future advancements in e-tutoring systems, emphasizing adaptability, accessibility, and user-centric design. The journey continues toward creating an e-learning environment that seamlessly integrates technology and education for a diverse and global audience.

REFERENCES

[1] Abu-Salih, B., Wongthongtham, P., Zhu, D., Chan, K. Y., Rudra, A., Abu-Salih, B., Wongthongtham, P., Zhu, D., Chan, K. Y., and Rudra, A. (2021). Introduction to big data technology. *Social Big Data Analytics: Practices, Techniques, and Applications*, pages 15–59.

[2] Adzifome, N. S. and Agyei, D. D. (2023). Learning with mobile devices-insights from a university setting in ghana. *Education and Information Technologies*, 28(3):3381–3399.

[3] Ahmed, G. and KOVÁCS, L. (2020). Ontology domain model for e-tutoring system. *Journal OF Software Engineering & Intelligent Systems*.

[4] Ahmed, G. H. A., Alshboul, J., and Kovacs, L. (2022). Development of ontology-based domain knowledge model for it domain in e-tutor systems. *International Journal of Advanced Computer Science and Applications*, 13(5).

[5] Akyuz, Y. (2020). Effects of intelligent tutoring systems (its) on personalized learning (pl). *Creative Education*, 11(6):953–978.

[6] Al Nuaimi, K., Mohamed, N., Al Nuaimi, M., and Al-Jaroodi, J. (2012). A survey of load balancing in cloud computing: Challenges and algorithms. In *2012 second symposium on network cloud computing and applications*, pages 137–142. IEEE.

[7] Al Rawashdeh, A. Z., Mohammed, E. Y., Al Arab, A. R., Alara, M., and Al-Rawashdeh, B. (2021). Advantages and disadvantages of using e-learning in univer-

sity education: Analyzing students' perspectives. *Electronic Journal of E-learning*, 19(3):107–117.

[8] Ali, A. (2022). An overview of cloud computing for the advancement of the e-learning process. *Journal of Theoretical and Applied Information Technology*, 100(3):847–855.

[9] Alrasheedi, M. and Capretz, L. F. (2013). A meta-analysis of critical success factors affecting mobile learning. In *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 262–267. IEEE.

[10] Alves, T., Natálio, J., Henriques-Calado, J., and Gama, S. (2020). Incorporating personality in user interface design: A review. *Personality and Individual Differences*, 155:109709.

[11] Araka, E. N. (2015). *An SOA Framework for Web-based E-learning Systems: A case of Adult Learners*. PhD thesis, University of Nairobi.

[12] Aung, T. H. (2014). *SaaS in Business: Exploring Strategic Benefits and Considerations of Software as a Service (SaaS) Model in Business Organizations*. Phd thesis, University of Porto.

[13] Avila, K., Sanmartin, P., Jabba, D., and Jimeno, M. (2017). Applications based on service-oriented architecture (soa) in the field of home healthcare. *Sensors*, 17(8):1703.

[14] Azam, M. G. (2019). Application of cloud computing in library management: innovation, opportunities and challenges. *RESEARCH REVIEW International Journal of Multidisciplinary*, 4(1):51–58.

[15] Baah, C., Govender, I., and Rontala Subramaniam, P. (2023). Exploring the role of gamification in motivating students to learn. *Cogent Education*, 10(1):2210045.

[16] Bendahmane, M., El Falaki, B., and Benattou, M. (2017). Towards a personalized learning path based on learning style and collaborative filtering.

[17] Bikanga Ada, M. (2023). Evaluation of a mobile web application for assessment feedback. *Technology, Knowledge and Learning*, 28(1):23–46.

[18] Cai, W., Grossman, J., Lin, Z. J., Sheng, H., Wei, J. T.-Z., Williams, J. J., and Goel, S. (2021). Bandit algorithms to personalize educational chatbots. *Machine Learning*, 110(9):2389–2418.

[19] Dalle, J.-M. and Jullien, N. (2001). Open-source vs. proprietary software. *Guest lecture at ESSID Summer School, Cargèse*.

[20] Dave, J. (2023). Enhancing user experience in e-learning: Real-time emotional analysis and assessment. *International Journal Software Engineering and Computer Science (IJSECS)*, 3(2):57–64.

[21] Derzho, M., Shafarenko, A., and Lilley, M. (2022). Controlling an application via a hybrid interface. In *International Conference on Human-Computer Interaction*, pages 194–200. Springer.

[22] Dhawan, S. (2020). Online learning: A panacea in the time of covid-19 crisis. *Journal of educational technology systems*, 49(1):5–22.

[23] El-Sabagh, H. A. (2021). Adaptive e-learning environment based on learning styles and its impact on development students' engagement. *International Journal of Educational Technology in Higher Education*, 18(1):1–24.

[24] Eryılmaz, M. and Adabashi, A. (2020). Development of an intelligent tutoring system using bayesian networks and fuzzy logic for a higher student academic performance. *Applied Sciences*, 10(19):6638.

[25] Hendradi, P., Ghani, M. K. A., Mohamad, S. N. M., Aissaoui, K., and Yudatama, U. (2023). Review of e-learning service framework based on cloud computing. In *AIP Conference Proceedings*, volume 2706. AIP Publishing.

[Hinchliffe] Hinchliffe, D. Dion hinchliffe's soa blog: How can we best make "the writeable web" a responsible place? `http://soa.sys-con.com/node/173822`. microservices expo.

[27] Hong, C.-M., Chen, C.-M., Chang, M.-H., and Chen, S.-C. (2007). Intelligent web-based tutoring system with personalized learning path guidance. In *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, pages 512–516. IEEE.

[28] Ilić, M., Mikić, V., Kopanja, L., and Vesin, B. (2023). Intelligent techniques in e-learning: a literature review. *Artificial Intelligence Review*, pages 1–47.

[29] Ilyas, Q. M., Ahmad, M., Zaman, N., Alshamari, M. A., and Ahmed, I. (2021). Localized text-free user interfaces. *IEEE Access*, 10:2357–2371.

[30] Ingle, D. and Meshram, B. (2013). Soa-a generic approach for integrating loosely coupled systems with other systems. *International Journal on Web Service Computing*, 4(2):1.

[31] Jalal, S. M. A. (2015). *Educational multimedia adaptation for power-saving in mobile learning*. PhD thesis, University of Southampton.

[32] Jamal, A. (2024). *Automatic Construction of Ontology with Public-Domain Datasets for Personalized Tutoring with ECA*. M.sc. thesis, School of Computer Science, University of Windsor.

[33] Jawahar, A. V. (2023). *Personalized ECA Tutoring with Self-Adjusted POMDP Policies and User Clustering*. M.sc. thesis, School of Computer Science, University of Windsor.

[34] Kajko-Mattsson, M., Lewis, G. A., and Smith, D. B. (2007). A framework for roles for development, evolution and maintenance of soa-based systems. In *International Workshop on Systems Development in SOA Environments (SDSOA'07: ICSE Workshops 2007)*, pages 7–7. IEEE.

[35] Katal, A., Wazid, M., and Goudar, R. H. (2013). Big data: issues, challenges, tools and good practices. In *2013 Sixth international conference on contemporary computing (IC3)*, pages 404–409. IEEE.

[36] Kaur, K. (2016). *An Appr An Approach of F oach of Facial Expr acial Expression Modeling with Changing T ession Modeling with Changing Trend in the History of Belief States*. M.sc. thesis, School of Computer Science, University of Windsor.

[37] Kentnor, H. E. (2015). Distance education and the evolution of online learning in the united states. *Curriculum and teaching dialogue*, 17(1):21–34.

[38] Keserwani, P. K., Samaddar, S. G., and Kumar, P. (2020). e-learning web services and their composition strategy in soa. In *Smart Computing Paradigms: New Progresses and Challenges: Proceedings of ICACNI 2018, Volume 2*, pages 291–302. Springer.

[39] Khan, M. M. (2023). *An Approach of Lip Synchronization with Facial Expression Rendering for an ECA*. M.sc. thesis, School of Computer Science, University of Windsor.

[40] Lee, B. N., Abdullah, S., and Kiu, S. N. (2016). Supportive learning: Linear learning and collaborative learning. *Turkish Online Journal of Educational Technology-TOJET*, 15(2):97–105.

[41] Lim, C. and Han, H. (2020). Development of instructional design strategies for integrating an online support system for creative problem solving into a university course. *Asia Pacific Education Review*, 21:539–552.

[42] Lippert, A., Shubeck, K., Morgan, B., Hampton, A., and Graesser, A. (2020). Multiple agent designs in conversational intelligent tutoring systems. *Technology, Knowledge and Learning*, 25:443–463.

[43] Liu, M. and Yu, D. (2023). Towards intelligent e-learning systems. *Education and Information Technologies*, 28(7):7845–7876.

[44] Manvi, S. S. and Shyam, G. K. (2014). Resource management for infrastructure as a service (iaas) in cloud computing: A survey. *Journal of network and computer applications*, 41:424–440.

[45] Muddam, A. R. and Satti, L. R. (2022). A review of cloud computing for the evolution of the e-learning process.

[46] Mulpuri, V. K. (2016). *Trend Analysis of Belief-State History with Discrete Wavelet Transform for Improved Intention Discovery.* M.sc. thesis, School of Computer Science, University of Windsor.

[47] Mylonas, P., Tzouveli, P., and Kollias, S. (2004). Towards a personalized e-learning scheme for teachers. In *IEEE International Conference on Advanced Learning Technologies, 2004. Proceedings.*, pages 560–564. IEEE.

[48] Pahl, C. (2015). Containerization and the paas cloud. *IEEE Cloud Computing*, 2(3):24–31.

[49] Palamarchuk, Y. (2022). Methods of building microservice architecture of e-learning systems. *Information Technologies and Computer Engineering*, (1):43–54.

[50] Palos-Sanchez, P. R., Arenas-Marquez, F. J., Aguayo-Camacho, M., et al. (2017). Cloud computing (saas) adoption as a strategic technology: Results of an empirical study. *Mobile Information Systems*, 2017.

[51] Patel, K. K. (2020). *A Simulation Environment with Reduced Reality Gap for Testing Autonomous Vehicles.* M.sc. thesis, School of Computer Science, University of Windsor.

[52] Paul, P., Aithal, P., and Saavedra, R. (2023). Cloud computing and its impact in education, teaching and research-a scientific review. *Teaching and Research-A Scientific Review (June 10, 2023). Book chapter in "Emergence and Research in Interdisciplinary Management and Information Technology" edited by PK Paul et al. Published by New Delhi Publishers, New Delhi, India*, pages 1–26.

[53] Peirce, N., Conlan, O., and Wade, V. (2008). Adaptive educational games: Providing non-invasive personalised learning experiences. In *2008 second IEEE international conference on digital game and intelligent toy enhanced learning*, pages 28–35. IEEE.

[54] Prihartini, N., Soemitro, H. L., and Hendradjaya, B. (2016). Identifying aspects of web e-learning in lms-based for requirement engineering process modeling. In *2016 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6. IEEE.

[55] Rahmatulloh, A., Nugraha, F., Gunawan, R., and Darmawan, I. (2022). Event-driven architecture to improve performance and scalability in microservices-based systems. In *2022 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*, pages 01–06. IEEE.

[56] Raval, R. R. (2019). *An Improved Approach of Intention Discovery with Machine Learning for POMDP-based Dialogue Management*. M.sc. thesis, School of Computer Science, University of Windsor.

[57] Shi, N., Zeng, Q., and Lee, R. (2020). Language chatbot–the design and implementation of english language transfer learning agent apps. In *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pages 403–407. IEEE.

[58] Simonson, M. and Schlosser, C. (2019). Research in distance education. *The Quarterly Review of Distance Education*, 20(3):31–43.

[59] Sottilare, R. A., Graesser, A. C., Hu, X., Olney, A., Nye, B., and Sinatra, A. M. (2016). *Design recommendations for intelligent tutoring systems: Volume 4-domain modeling*, volume 4. US Army Research Laboratory.

[60] Sprott, D. and Wilkes, L. (2004). Understanding service-oriented architecture. *The Architecture Journal*, 1(1):10–17.

[61] Stack, P., Manzoor, F., Menzel, K., and Cahill, B. (2010). A service oriented architecture for building performance monitoring. *Architectural Journal, Microsoft.*

[62] Subaramaniam, K., Ern-Rong, J. L., and Palaniappan, S. (2021). Interface designs with personality types: an effective e-learning experience.

[63] Szucs, T. (2020). *Lip Synchronization for ECA Rendering with Self-Adjusted POMDP Policies.* M.sc. thesis, School of Computer Science, University of Windsor.

[64] Teich, T., Wolf, S., Neumann, T., Junghans, S., and Franke, S. (2014). Concept for a service-oriented architecture in building automation systems. *Procedia Engineering*, 69:597–602.

[65] Upadhyay, R. D. (2019). *An SOA-Based Framework of Computational Offloading for Mobile Cloud Computing.* M.sc. thesis, School of Computer Science, University of Windsor.

[66] Vyas, N. (2022). *An Approach of Using Embodied Conversational Agent for Personalized Tutoring.* M.sc. thesis, School of Computer Science, University of Windsor.

[67] Wijekumar, K., Meyer, B. J., Lei, P., Beerwinkle, A. L., and Joshi, M. (2020). Supplementing teacher knowledge using web-based intelligent tutoring system for the text structure strategy to improve content area reading comprehension with fourth-and fifth-grade struggling readers. *Dyslexia*, 26(2):120–136.

[68] Wolfert, P., Robinson, N., and Belpaeme, T. (2022). A review of evaluation practices of gesture generation in embodied conversational agents. *IEEE Transactions on Human-Machine Systems*, 52(3):379–389.

[69] Wu, L., Garg, S. K., Versteeg, S., and Buyya, R. (2013). Sla-based resource provisioning for hosted software-as-a-service applications in cloud computing environments. *IEEE Transactions on services computing*, 7(3):465–485.

[70] Yağcı, M. (2022). Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, 9(1):11.

[71] Yin, S., Yang, J., Zhang, Q., and Zhang, L. (2012). A survey of cloud computing research. *IEEE Transactions on Services Computing*, 5(4):539–549.

VITA AUCTORIS

NAME:	Parth H. Mistry
PLACE OF BIRTH:	Mumbai, Maharastra, India
YEAR OF BIRTH:	1999
EDUCATION:	Vadodara Institute of Engineering, Kotambi, India, 2021
	University of Windsor, M.Sc in Computer Science, Windsor, Ontario, 2024