technologies

Article

A Design-Based Research Approach to Streamline the Integration of High-Tech Assistive Technologies in Speech and Language Therapy

Anna Lekova [1,*], Paulina Tsvetkova [1,2], Anna Andreeva [1,3], Georgi Dimitrov [1,2], Tanio Tanev [1], Miglena Simonska [3], Tsvetelin Stefanov [1,2], Vaska Stancheva-Popkostadinova [3], Gergana Padareva [4], Katia Rasheva [2], Adelina Kremenska [1] and Detelina Vitanova [2]

[1] Institute of Robotics, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria; p.tsvetkova@ir.bas.bg (P.T.); anna_andreeva@swu.bg (A.A.); g.dimitrov@unibit.bg (G.D.); t.tanev@ir.bas.bg (T.T.); a.kostova012@gmail.com (A.K.)
[2] Faculty of Information Science, University of Library Studies and Information Technologies, 1784 Sofia, Bulgaria; k.rasheva@unibit.bg (K.R.); d.vitanova@unibit.bg (D.V.)
[3] Public Health, Health Care and Sport, South-West University, 2700 Blagoevgrad, Bulgaria; miglena_simonska@swu.bg (M.S.); v_stancheva@abv.bg (V.S.-P.)
[4] Faculty of Philology, South-West University, 2700 Blagoevgrad, Bulgaria; gerypadareva@swu.bg
* Correspondence: a.lekova@ir.bas.bg

Abstract

Currently, high-tech assistive technologies (ATs), particularly Socially Assistive Robots (SARs), virtual reality (VR) and conversational AI (ConvAI), are considered very useful in supporting professionals in Speech and Language Therapy (SLT) for children with communication disorders. However, despite a positive public perception, therapists face difficulties when integrating these technologies into practice due to technical challenges and a lack of user-friendly interfaces. To address this gap, a design-based research approach has been employed to streamline the integration of SARs, VR and ConvAI in SLT, and a new software platform called "ATLog" has been developed for designing interactive and playful learning scenarios with ATs. ATLog's main features include visual-based programming with graphical interface, enabling therapists to intuitively create personalized interactive scenarios without advanced programming skills. The platform follows a subprocess-oriented design, breaking down SAR skills and VR scenarios into microskills represented by pre-programmed graphical blocks, tailored to specific treatment domains, therapy goals, and language skill levels. The ATLog platform was evaluated by 27 SLT experts using the Technology Acceptance Model (TAM) and System Usability Scale (SUS) questionnaires, extended with additional questions specifically focused on ATLog structure and functionalities. According to the SUS results, most of the experts (74%) evaluated ATLog with grades over 70, indicating high acceptance of its usability. Over half (52%) of the experts rated the additional questions focused on ATLog's structure and functionalities in the A range (90–100), while 26% rated them in the B range (80–89), showing strong acceptance of the platform for creating and running personalized interactive scenarios with ATs. According to the TAM results, experts gave high grades for both perceived usefulness (44% in the A range) and perceived ease of use (63% in the A range).

Keywords: high-tech assistive technologies; socially assistive robots; virtual reality; conversational AI; software platform; visual programming; speech and language therapy; communication disorders; children; technology and usability assessment

check for updates

Academic Editors: Jeffrey W. Jutai and Daniele Giansanti

Received: 2 April 2025
Revised: 25 June 2025
Accepted: 9 July 2025
Published: 16 July 2025

Citation: Lekova, A.; Tsvetkova, P.; Andreeva, A.; Dimitrov, G.; Tanev, T.; Simonska, M.; Stefanov, T.; Stancheva-Popkostadinova, V.; Padareva, G.; Rasheva, K.; et al. A Design-Based Research Approach to Streamline the Integration of High-Tech Assistive Technologies in Speech and Language Therapy. *Technologies* **2025**, *13*, 306. https://doi.org/10.3390/technologies13070306

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The present study follows a Design-Based Research (DBR) approach [1] to explore how high-tech assistive technologies (ATs), such as Socially Assistive Robots (SARs), virtual reality (VR) and conversational AI (ConvAI), can be easily integrated into and used in Speech and Language Therapy (SLT) for children and adults with communication disorders (CDs), as well as to identify the design principles that support the successful implementation of ATs in SLT. Scientific research shows that therapy with high-tech ATs improves the skills of children with CDs. According to Kouroupa [2], most interventions used humanoid (67%) robotic platforms; were predominantly based in clinics (37%), followed home, school, and laboratory (17%, respectively) environments; and targeted at improving social and communication skills (77%). In a 2023 survey of 53 speech and language therapists specializing in autism, approximately 92% reported being aware of VR; however, only 1.8% (1 therapist) had actually used it clinically [3]. The use of ATs shows low clinical adoption, likely due to the lack of a standardized methodology, high costs that make many technologies unaffordable, fast technological development, need for specialized training for therapists, and technical limitations related to the integration and programming of ATs. Additionally, there are increasing expectations for SAR autonomy and human-like speech recognition and response capabilities, allowing them to understand, generate and vocalize text. This highlights the significance of Natural Language Processing (NLP), Natural Language Generation (NLG) (a type of Generative AI), and cloud technologies as key assistive technologies in SLT. According to recent studies [4,5], the rapid progress of Large Language Models (LLMs) and Large Reasoning Models (LRMs), such as Generative Pre-trained Transformers (GPTs) with Chain-of-Thought (CoT) prompting for step-by-step decision making, will further enhance ConvAI and its relevance in SLT.

Despite the generally positive public perception of SARs, VR and ConvAI among therapists [6–15], one of the most significant challenges is how therapists individually integrate AT into their practice. Related difficulties include software programming, user-friendliness, and ease of setup and use. Santos et al. [15] highlighted these challenges in their discussion on interaction scenarios with ATs in future studies, emphasizing the need for user-friendly systems that minimize the requirement for technical skills among therapists. Additionally, there is a lack of ergonomic software platforms to effectively support SLT professionals in streamlining the design of interactive scenarios during intervention. User-friendly platforms often include visual programming features with pre-programmed graphical blocks, which is particularly helpful in educational and therapeutic settings [16].

Since there is a gap between the potential of ATs and their practical application in SLT, this study aims to contribute to the conceptualization of a design for integrating high-tech ATs and creating interactive, playful learning scenarios that support therapists in their practice. We report on a DBR study conducted over three years to develop and implement an ergonomic platform called ATLog, following the four phases of the DBR approach [1] and the logic of effective iteration outlined in [17]. According to [17], educational and therapeutic innovations are context-dependent and should be designed and implemented within real-world settings, requiring collaboration among researchers, practitioners, and stakeholders. In the frame of ATLog project [18], we systematically involved SLT professionals from the Logopedic Center at the South-West University in the iterative process of designing, developing, implementing and evaluating SLT interventions using ATs. This collaboration also focused on streamlining the design of interactive scenarios for playful learning with SARs, VR and ConvAI. Previous iterations have been reported in earlier work [19–23], detailing the building–testing cycles of the ATLog architecture, including the integration of various SARs, integration of AI cloud services for NLP/NLG into the SARs' native software, and implementation of VR scenes for interactive speech and lis-

tening experiences. During these iterations, we optimized the client–server architecture to better integrate ATs and enhanced their communication protocols with pre-configured graphical blocks within a visual programming interface. These blocks were customized for specific treatment domains, therapy goals, and language skill levels, enabling dynamic combinations and adaptable scenarios tailored to the therapeutic needs of each child.

To ensure the effectiveness of the ATLog platform, it is essential to evaluate its acceptability and usability. According to the literature review, we performed a System Usability Scale (SUS) questionnaire [24], a highly effective instrument for evaluating the user experience of a novel platform. In addition, to examine the application's perceived usefulness and perceived ease of use, the Technology Acceptance Model (TAM) questionnaire was employed [25]. These questionnaires typically use a Likert scale to measure user opinions and are widely recognized instruments for analyzing and interpreting user behavior and the effectiveness of technological systems. The SUS and TAM questionnaires are often used together in such evaluations [26].

The research objectives of this study are as follows: (1) to design and (2) to evaluate an ergonomic platform to streamline the integration of high-tech ATs in SLT following the DBR approach. To achieve these objectives, the first phase in DBR involved consultations with SLT professionals to identify the challenges they face in integrating high-tech ATs, such as technical barriers, platform complexity and the need for an intuitive and flexible user interface tailored to therapeutic practice. A literature review was also conducted to examine existing technological solutions and systems relevant to CDs. In the second phase, iterative versions of the ATLog platform were developed. During the third phase, feedback on usability, logic flow and scenario effectiveness was collected from the pilot deployment in real SLT environments with children and therapists. The TAM and extended SUS questionnaires were employed to identify platform acceptance, usability issues and areas for improvement. Based on the data collection and analysis in this phase, the main design principle was identified during the fourth phase. This principle, along with a comprehensive framework for conceptualizing the design process and technical implementation guidelines for developers (provided in the Appendices A and B), are presented to support replication and further development.

The paper is organized as follows: Section 2 presents the related works. Section 3 describes the methodology of the study, materials and methods, and the validation approach. Section 4 presents the result implementation of the ATLog platform and the results from the TAM and SUS questionnaires, while Section 5 presents the Discussion. Then, conclusions, limitations and future directions follow.

2. Related Works

A literature review was conducted during the first phase of DBR to achieve the following: (1) explore the technical, ethical and cultural issues that hinder the acceptance and adoption of high-tech ATs by speech and language therapists, along with the methods used to validate technology acceptance and adoption; (2) analyze case studies demonstrating how high-tech ATs have been integrated into SLT; (3) review existing visual-based programming platforms that explore similar design solutions for intuitive and modular interfaces to diverse ATs.

2.1. Technical, Ethical and Cultural Issues That Hinder the Acceptance and Adoption of High-Tech ATs by Speech and Language Therapists

Despite the generally positive public perception of SARs, VR and ConvAI among therapists [6–15], one of the most significant technical challenges is the real integration of ATs into their practice, where flexibility, reliability and ease of use are critical. Concerns

involve the need for an easy setup that does not require technical skills beyond a therapist's standard training or any software programming expertise. Further concerns include the absence of clear clinical guidelines, unfamiliarity with the technology, risk of system failures, and short training. Therapists' acceptance and use of ATs are strongly influenced by their confidence, the complexity of the ATs, technical limitations and the availability of support. Szabó et al. [7] point out that while speech and language therapists have access to a wide range of digital tools with potential therapeutic benefits, actual adoption depends heavily on several factors: the therapist's self-confidence in using digital tools, the time and effort required to learn them, insufficient technical infrastructure, and lack of peer or supervisory support. To address these barriers, user-friendly design and practical training, such as tutorials, troubleshooting guides and video lessons are essential. Santos et al. discussed in [15] the characteristics of interaction scenarios and the need for systems to be user-friendly for therapists, thus minimizing the requirements for technical skills. This supports the observations in [7], where therapists are seen as dual-role stakeholders—both as users of ATs and as facilitators who guide patients in accessing and using them effectively. Therefore, the experiences of both therapists and patients are essential to successful implementation. If therapists have negative initial experiences using ATs in therapy, they are more likely to discontinue their use. Additional concerns when integrating ConvAI include the accuracy of AI-generated outputs and the need for continuous monitoring to prevent the dissemination of incorrect information [27].

Concerns regarding the integration of SARs into SLT include the complexity of programming SARs for therapeutic purposes. This challenge lies not in simply making the robot operate, it requires robot behavior to align with complex therapeutic goals for enhancing communication and social skills, such as refining speech, improving listening, supporting turn-taking, and recognizing social cues. These are subtle, context-dependent abilities, and programming them is often time-consuming or exceeds the typical technical expertise of therapists, making a predefined, user-friendly interface essential.

Rupp et al. identified the key concerns and expectations of professionals in [10]:

- Technical expertise—therapists may lack the skills needed to program robots, raising concerns about their use in therapy;
- Software complexity—programming tools for SARs are often too complex or overly limited;
- Robot behavior—the robot must adapt to children's abilities, requiring customization based on individual needs;
- User-friendliness—the ease of setup and interface design are critical for adoption in clinical settings.

Conflicting perceptions arise among therapists when using SARs in real-world therapy sessions. The third research question in [9] explicitly asked: "*To what extent do therapists who have experienced SARs believe in their efficacy and intend to use them in their speech-language therapy practice to train children with language impairments?*", and the results revealed that therapists generally remained skeptical about the efficacy of SARs for improving the communication and social skills of children with language impairments. The study analyzed the usability, engagement and perceived benefits of SARs and virtual SARs (V-SARs) through questionnaires and group interviews. While therapists found both SARs and V-SARs easy to use, they noted that SAR setup was time-consuming and they remained careful about fully adopting such tools for improving communication and social interactions. Therapists observed that although both SARs and V-SARs effectively engaged children, SARs had more appeal and were seen as a companion, whereas V-SARs felt more like a training tool. Although therapists appreciated the engagement benefits of SARs, they were not ready to fully adopt them as primary therapeutic tools, expressing a preference for traditional,

therapist-led approaches. Nonetheless, they did see value in SARs for supplementary roles in therapy, particularly in home environments.

Concerns regarding the integration of VR in SLT include the need for a stronger neuro-affirming evidence base supporting VR effectiveness, the absence of clear clinical guidelines, and a lack of adequate training for therapists to effectively integrate VR into their practice. Vaezipour et al. [28] examined speech and language therapists' acceptance of VR in communication rehabilitation by involving fifteen professionals in communication activities within an immersive VR kitchen environment. The study evaluated usability and motion sickness, and gathered qualitative feedback. While participants expressed positive attitudes toward VR's potential, the concerns raised in [28] are why the scores from the usability test were only average. Although the motion sickness was minimal, there were barriers such as unfamiliarity with the VR technology, insufficient training and technical complications, such as complex setup procedures, hardware limitations and risk of system failures.

Cost emerged as another significant concern, as the development of 3D applications, along with the acquisition, maintenance of VR equipment and required training, can be excessively expensive. Furthermore, many therapists remain uncertain about VR's actual clinical effectiveness, and this skepticism contributes to its limited adoption. Although public attitudes toward VR and AR are generally positive, as shown in [11], with both students and teachers highlighting educational benefits, these perceptions have not yet translated into widespread clinical use. The study in [3] found that despite VR being viewed as a promising tool for practical exercises, it has not been widely adopted in SLT. In fact, while 92% of speech–language therapists were aware of VR, only 1.8% had used it in therapy for autistic children. Further concerns were raised by Hashim et al. [29], who investigated stakeholder perspectives on applications with AR developed for children with CDs. Although the tool was seen as promising for supporting language learning among autistic children, stakeholders emphasized that substantial improvements would be needed to make vocabulary learning clinically useful. The review in [30] discusses the lack of tools to measure the outcomes of VR interventions in SLT aimed at improving communication skills.

Concerns regarding the integration of ConvAI in SLT practice are complex. One of the primary issues is the need for continuous monitoring to prevent the dissemination of inaccurate or inappropriate information [27]. Integrating generative AI tools (such as ChatGPT) into SLT also presents challenges, primarily due to the technical expertise required to refine these LLMs with existing therapeutic workflows and AT software. The integration of ConvAI into SARs is considered a key feature for enabling natural, human-like interactions with children. However, the complexity of implementing such integration remains a significant barrier. Robots such as Probo [31], QTrobot [32], FurhatAI [33], ARI [34], and Milo [35] have shown promising levels of acceptance due to their ConvAI capabilities. Advancements in LLMs and deep learning (DL) [36,37] are gradually improving user experiences and reducing skepticism toward ConvAI integration in SARs. Study [36] explores the enhancements of the Nao robot's language abilities using DL techniques, while [37] introduces FurChat—an embodied conversational agent powered by LLM and dialog management that enables open domain-specific conversations with facial expressions. Although these innovations demonstrate the growing potential of ConvAI, therapists often prefer generative AI (such as ChatGPT, DALL·E or Make-A-Video) for creating customized therapy materials rather than real-time interaction through ConvAI-enhanced SARs.

We examined the careful optimism among SLT professionals regarding AI adoption [12,13,38–40]. The identified concerns are as follow:

- Accessible training in AI basics—many therapists lack the background in AI or programming needed to effectively utilize or troubleshoot ConvAI systems. Continuous

technical support and directions on how to use ChatGPT for speech therapy are essential to facilitate this transition [39].

- Time constraints—according to [12], 32.4% of professionals expressed interest in adopting new AI tools but mentioned limited time as a barrier. Some therapists (e.g., participants S27, S41) openly admitted that they tend to avoid using technology.
- Data privacy and compliance—therapists expressed concerns about compliance with privacy laws such as HIPAA and FERPA, especially when using AI tools for transcription or data processing (e.g., participant S10 in [12]).
- Rapid technological change—the speed at which AI technologies evolve makes it difficult for practitioners to stay up-to-date. One participant (S17) cited "keeping up with technology" as a top challenge.
- System compatibility and latency—the integration of AI systems into existing platforms can be technically demanding. Authors in [40] describe in detail the limitations of integrating ChatGPT into the Pepper robot. Compatibility issues (Pepper uses Python 2.7; ChatGPT APIs require Python 3.x) and dependency on stable internet connections often lead to latency, which disrupts smooth and real-time interaction.

Some speech and language therapists have raised additional concerns about inadequately programmed SARs or VR in interactive scenarios with children with special needs [2,41,42]. Poorly designed interactions can increase anxiety in children and reduce both therapist confidence and the overall acceptance of ATs. Successfully integrating SARs into existing healthcare or educational frameworks, therefore, demands considerable customization and ongoing efforts to ensure that these systems are aligned with the specific needs of both children and therapists. Key factors for successful adoption include adequate training for therapists, user-friendly and intuitive system design, adaptability to specific therapeutic goals, familiarity with the technology, and clear evidence of its clinical effectiveness. These elements are essential to foster trust; however, across the surveyed studies, there was a noticeable lack of simplified programming interfaces, seamless setup and operation, and adaptable features for designing interactive scenarios tailored to different therapeutic needs. Furthermore, to make the technology more accessible, software developers, engineers and therapists need to work together and to improve the usability of therapeutic robots.

How people view and accept ATs can differ depending on their culture [6,43,44]. The findings in [43] show that Chinese participants generally expressed positive feelings, viewing ConvAI agents more as sources of enjoyment and perceiving voice-based and physically embodied agents as warmer, more competent and emotionally engaging. In contrast, US participants saw ConvAI agents more functionally, with mixed or uncertain feelings. In Germany, cultural attitudes shifted notably during the COVID-19 pandemic, where the integration of ATs into SLT became not just an option, but a necessity [6]. Two online questionnaire-based studies are reported in [6], investigating the acceptance of technology among speech and language therapists in Germany. In the first study, external factors that inhibited or facilitated the adoption of video therapy and its potential future utilization were identified. In the second study, one of the research question is "*To what extent are speech and language therapists intending to use ICT in therapy in the future?*" and the modified model of the Unified Theory of Acceptance and Use of Technology (UTAUT) explained 58.8% of the variability in therapists' behavioral intention to use digital media. Both studies investigated facilitating and inhibiting factors for including video therapy in future speech and therapy services, and the results demonstrated an accelerated digital adoption in SLT services for telepractice.

Last but not least, integrating ATs into SLT raises important ethical and practical concerns. Although Severson et al. [45] identified social robots as among the most effective ATs

in education and therapy for children, they emphasized that SARs must not give children the false impression of possessing emotions or consciousness. This raises concerns about emotional deception and unrealistic expectations, especially in vulnerable populations. There are also concerns about equity and accessibility—robots should be designed to serve all users fairly, avoiding over-personalization through machine learning that may reinforce biases or limit adaptability. Severson et al. further recommend that policymakers adopt evidence-based approaches to guide the certification and deployment of SARs in educational and clinical settings. Another concern is the ethical handling of child data, especially when used to train perceptual tools like speech or expression recognition systems. Ensuring data security, anonymization and regulatory compliance is critical. The survey in [46] reinforces these concerns, highlighting factors that affect therapists' adoption of ATs like SARs. These include trust, privacy, and ethical considerations, as well as cultural sensitivities and past experiences that may influence acceptance.

In exploring how to validate technology acceptance and adoption among therapists, we found that a range of qualitative methods were commonly used. To examine users' attitudes and concerns towards how constantly evolving hi-tech digital media technologies are used and accepted, the authors in [6] applied a modified model of UTAUT, while the authors in [7] used the original UTAUT model to assess the acceptance of apps and digital resources in SLT. The authors in [9] tested the usability and adoption of SARs among therapists via the Adoption of Technology (AoT), Quebec User Evaluation of Satisfaction with Assistive Technology (QUEST), and SUS questionnaire. In [28], the authors used the SUS, the NASA Task Load Index (NASA-TLX) for assessing subjective mental workload, and the Simulator Sickness Questionnaire (SSQ) to evaluate the acceptance, usefulness and usability of the SIM:Kitchen VR application for SLT. In [38], the authors applied the TAM2 model, which extends the original TAM, to assess the acceptance of chatbots as AI conversational partners in language learning, given their focus on specifically evaluating chatbots' role as an interactive agent. The TAM2 consists of 25 items from seven dimensions: perceived ease of use (PEU), perceived usefulness (PU), attitude (ATT), perceived behavior control (PBC), behavioral intention (BI), self-efficacy (SE), and personal innovativeness (PI). In the systematic review in [42], where the attitudes, acceptance, anxiety and trust toward social robots were analyzed, the measures employed by authors were the Almere Model of robot acceptance and UTAUT.

Based on the reviewed evaluation methods, we identified the SUS and TAM as the most suitable frameworks for assessing therapists' acceptance and adoption of the ATLog platform in Phase 3.

2.2. Findings from Case Studies Demonstrating How Hi-Tech ATs Have Been Integrated into SLT

While there are case studies that explore the integration of SARs or VR in SLT for children, to the best of our knowledge, no research papers have reported case studies involving the use of ConvAI, exploiting ChatGPT, Gemini, DeepSeek, etc., in SLT for children. As mentioned above, therapists have shown openness to the potential of generative AI for creating customized therapy materials, such as tailored stories, word lists, and sentences targeting specific speech or language goals, as well as educational resources like flashcards and interactive games. Du et al. [13] explores the potential adoption of ChatGPT as a promising tool in SLT for individuals with receptive and expressive language disorders. ChatGPT can help with vocabulary development through definitions and synonyms, generating stories for narrative skill practice, providing resources for literacy activities like reading and writing, assisting with bilingual therapy through translations, and offering information to enhance cultural competence during sessions. Therefore, we anticipate that the adoption of text-generative AI for conversations in SLT will increase in the near

future also. Du et al. [13] advise therapists that, in addition to creating therapy materials, ChatGPT can simulate human-like interaction, allowing children to practice grammar and syntax, engage in social communication tasks such as turn-taking and topic maintenance, and learn by experience in a low-pressure conversational environment. Table 1 in [12] presents the speech and language therapists' perceptions of AI technologies. From 105 SLT practitioners, 79 are excited about AI's potential to improve communication outcomes and save time and 75 respondents see promise in its applications. S36 said, "*I would like to better understand AI and how it can support my work with students to improve their communication abilities*". However, in response to RQ1: "*What aspects do they enjoy most about their job that should not be negatively impacted by AI?*", most expressed frustration over the lack of time for preparation, planning and direct intervention with children and families.

Case studies exploring the integration of SARs in SLT are presented in Table 2 in the survey [47], which summarizes twelve empirical use cases for interactive scenarios with SARs [48–59]. Table 1 in the survey [46], which includes rows focused on children as end users, also presents examples of applications and experiments involving the use of social robots. Additional recent case studies can be found in [9,11,46,60]. Spitale et al. in [9] studied the use of SARs in SLT for children with language impairments. Over an 8-week period, 20 children and 6 speech-language therapists participated, with children randomly assigned to either a physical or virtual SAR condition. Key findings in [11] include the following: (1) significant improvements in linguistic skills in both SAR conditions; (2) greater engagement and speech occurrences observed in children interacting with the physical SAR. The study conducted by [60] demonstrated that humanoid educational robots can foster social and emotional skills. The study focuses on designing humanoid robots to engage students from both functional and affective perspectives and reports a pilot test involving 64 primary school students in Hong Kong, divided into control and experimental groups. The validity of the findings was ensured through a combination of questionnaires, observations, and language proficiency tests. The study employed motivational engagement theory to analyze the outcomes, which revealed that the experimental group, which interacted with humanoid robots, showed significant improvements in behavioral engagement (+13.24%), emotional engagement (+13.14%), cognitive engagement (+21.56%), and intrinsic motivation (+12.07%).

Two papers focused on case studies integrating VR in SLT for children and adolescents [61–65] examine the role of VR in enhancing speech and language skills, showcasing its interactivity, engagement and efficacy. A pilot randomized controlled trial (n = 32) comparing traditional speech therapy to VR-based therapy (using the VRRS system) in children with developmental language disorders showed greater gains in the VR group for language comprehension, naming, syntax and spontaneous speech [61]. The study by Macdonald [65] evaluated the impact of a single 30 min session of overexposure therapy on 29 adolescent students. The results showed a substantial decrease in public speaking anxiety and a notable increase in confidence and enjoyment of public speaking. We see its great potential for stuttering therapy, as well.

Pergantis in [66] presents 36 case studies involving different high-tech ATs, which are summarized in Table 2. These case studies provide detailed information on the type of study, participant characteristics (age, gender and diagnosis), intervention, type of technology, measurement tools, results, limitations, effect sizes, longitudinal effects and country. The studies emphasize the importance of user-centered design and therapist involvement to ensure effectiveness and promote adoption. They highlight the growing role of high-tech ATs, such as VR, serious games, and SARs, in supporting emotional regulation and communication development in children with ASD. Key findings from selected case studies in [66] relevant to CDs are as follows: (1) Virtual reality scenarios

aimed at emotional regulation and social interaction demonstrated significant positive outcomes—a 14-week program involving children with ASD showed improvements in emotional expression and reduced stress responses, indicating VR's therapeutic potential for developing communication and coping strategies; (2) studies involving social robots such as KASPAR and PARO revealed encouraging results. KASPAR helped reduce anxiety-related behaviors in children with ASD, as indicated by physiological measures like heart rate variability. PARO, used with inpatient adolescents, also showed positive effects on mood and anxiety levels.

2.3. Existing Visual-Based Programming Platforms That Explore Similar Design Solutions for Intuitive and Modular at Interfaces

Many platforms propose interactive tools for designing and developing interactive scenarios with low-tech ATs in SLT for children and adolescents with CDs, such as Augmentative and Alternative Communication (AAC), customizable communication books with words and pictures, adaptive writing tools, interactive software for practicing skills, communication and social interaction, and mobile applications for smartphones and tablets. High-tech ATs are less commonly applied, although Tobii Dynavox [67] offers products and software specifically designed to support individuals with speech and language disorders, such as "TD Snap" and "Pathways" for visual scene displays. These tools enable therapists and caregivers to customize communication boards, symbols and workflows in order to meet a child's specific needs. However, the price is high and there is currently no dedicated visual-based programming platform.

Klavina et al. in [68] studied an additional type of high-tech AT with visual-based programing platform for VR, AI-empowered mobile applications and other AAC software to improve daily living skills and communications. Although the authors claimed that these tools offer engaging, user-friendly platforms that meet individual needs and improve learning, few were tailored to assist cross-disciplinary professionals with limited programming skills in integrating ATs and designing interactive scenarios for their interventions. Robotics platforms, like Ask Nao [69], Furhat Blockly [70], Vittascience Interface [71] and LEKA [72], focus on interaction and the development of interactive scenarios, but with a *high degree of universality*. Ask Nao, Vittascience and Furhat Blockly offer a simple drag-and-drop interface for building and controlling interactive behaviors for NAOv6 or Furhat without needing to write text code; however, they are not typically focused on a specific therapy, even though the blocks represent a variety of actions and interactions for creating dynamic and complex behaviors, along with the robot's capability to respond with empathy to the user's behavior. For instance, the Furhat visual platform facilitates speech and conversational interactions by design of facial expressions and lip synchronization, while also detecting user presence and recognizing user emotions. Ask Nao and LEKA's interfaces allow users to design and customize robot behaviors using graphical blocks, including gestures, speech, movements and sensory interactions. These tools are used for activities such as emotional engagement, sensory tasks and social skill development, but are not specifically tailored for SLT.

Open Access Virtual Reality Overexposure Therapy [73] is an open-access platform, compatible with various devices for VR public speaking, aimed at reducing public speaking anxiety. Platforms like ThingLink [74] enable users to create interactive VR scenarios for learning and training, offering opportunities for communication skill development through immersive and real-world simulations. However, despite their potential, limited accessibility remains a concern due to high costs and non-intuitive user interfaces, which can pose barriers for educators and therapists with limited technical expertise. The MagicItem tool for the dynamic behavior design of virtual objects with LLMs, presented in [75], explores how LLMs can shape the behavior of virtual objects in VR environments. By integrating

LLMs with platform-specific scripting tools, therapists with minimal coding skills can easily define and control object behaviors, adding depth and interactivity to VR scenes. Similarly, therapy with VR [76] offers a set of customizable virtual reality scenarios designed to assist speech therapists in helping individuals enhance their vocal abilities, providing a versatile toolkit for therapeutic interventions in VR.

The platforms for integrating text-generative AI in SARs or VR usually exploit cloud services for enhancing the conversational capabilities of ATs. For example, the authors in [77] present a cloud-based platform designed to provide knowledge-based autonomous interactions for social robots and conversational agents, particularly beneficial for low-cost robots. LuxAI for QTrobot proposes a seamless integration of ChatGPT for language understanding and generation [78]. The Furhat conversational platform is also built on cloud services, utilizing NLP for understanding and NLG to generate responses.

2.4. Identified Issues and Areas for Improvement

During the first phase of DBR, we identified the following issues that need to be addressed:

- There is a need for an ergonomic platform that supports therapists in using various ATs. This platform should serve as a general hub, allowing therapists to seamlessly integrate ATs.
- The platform should offer a GUI that allows therapists to personalize the robot's receptive and expressive language, as well as interactions, to align with each child's therapeutic goals and developmental stage.
- The platform should provide secure data management, including storing and analyzing progress through quantitative performance metrics and session summaries.
- The platform should be tailored to SLT, which would save time for therapists in their intervention preparation. Visual programming platforms are generally universal in design and do not typically focus on specific therapies. They often lack pre-designed content, such as a content library with therapeutic interactive scenarios enhanced by ATs.
- There is a need for robot cooperation, which arises when a robot lacks certain functionalities, such as QR code reading or Automatic Speech Recognition (ASR). This calls for shared repositories and direct or message-based input/output channels to facilitate communication between robots and support such cooperation.
- There is a need for a remote mode in the platform for monitoring and controlling robots for children receiving therapy at home.
- There is a need for providing technical support for therapists, and training on how to integrate ATs through tutorials and video lessons. Therapists also need special training to use block-based GUIs, even though they are often intuitive.

To address the identified challenges, we initiated Phase 2 of the DBR process to design and develop ATLog—an innovative platform to facilitate the integration of ATs through a user-friendly visual programming editor, which uses graphical blocks tailored for SLT and allows therapists to configure and control various AT devices and resources without requiring advanced technical skills. Thus, the platform will enable customization by a modular combination of robot actions and VR scenes, broken down into micro-skills and micro-scenes, to suit the individual needs of each child. It will also include features like multi-robot cooperation, access to content repositories, secure communication protocols with authentication, built-in tools for quantitative assessment and a remote mode to support SLT from a distance.

3. Materials and Methods

3.1. Methodology

This section describes the research approach based on the four phases of the DBR process (Figure 1) using the Reeves model [1]. DBR focuses on applying research findings in practical, real-world settings with active stakeholder participation and continuous iterations. Iterations consist of repeated cycles of early and frequent building and testing, choosing flexible design versions that effectively support the goals and practices of professionals in each cycle. Thus, through the iterative process used to co-develop, test and refine the integration of SARs, VR and ConvAI into the novel ergonomic software platform ATLog, continuous improvements were achieved. Speech and language therapists actively contributed to streamlining the design of interactive scenarios for playful learning within the platform.

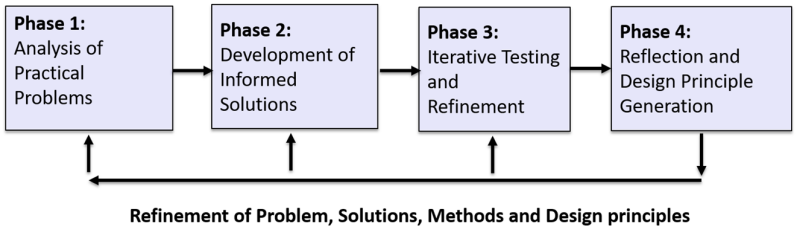


Figure 1. Four phases of the design-based research process using the Reeves DBR model, adapted from [1].

During Phase 1 of the DBR approach, we examined the objectives and therapeutic practices of therapists aiming to integrate high-tech ATs into SLT principles and methods. A literature review explored the technical, ethical and cultural barriers to adopting ATs and analyzed case studies of AT integration into SLT. The methodology of the design aligns with the theoretical foundations and principles of SLT: (1) Didactic—focusing on visibility and accessibility; (2) Ontogenetic—focusing on the systematic development of speech forms, functions and expressions. Other integrated principles include the Principle of Interactive Learning, the Principle of Student Activity, and the Principle of Personalized Learning. The methods integrated in the methodology are as follows: (1) the phonological awareness method, as a foundation for language and literacy development, focuses on identifying and manipulating language sounds, such as rhyming, blending, segmenting and changing individual phonemes within words; (2) the empirical method of knowledge acquisition, i.e., learning through practical interaction with the environment rather than theory; (3) embodiment learning, involving the body and physical actions in understanding; and (4) multisensory learning, which engages multiple senses to improve comprehension and knowledge retention. We also explored therapists' interest in using ConvAI with robots that lack built-in NLG abilities, like Nao, during SLT interventions. We analyzed the practical problems encountered and identified several technical challenges, including the need to improve speech recognition accuracy, generate context-aware human-like text, reduce response latency, implement multilingual TTS synthesis, and resolve communication issues between ATs that use different protocols.

During Phase 2, in the ongoing iteration of developing and testing the software architecture for integrating ConvAI in the Nao, Furhat and Emo robots, new technical solutions were implemented. The initially used Node-RED platform as central component had been replaced with 'Express server' in order to improve functionality and reduce system latency. This change took advantage of the lightweight client–server framework using Node.js for the backend, paired with a web server interface for the frontend. Using the 'spawn' module of Node.js, which allows the execution of Python or JavaScript files as child processes

via server endpoints, streamlined the integration of ATs into the ATLog platform. This network solution ensures that only the child processes need updates when technology evolves, makes the maintenance easy. Furthermore, using a Node.js-based server instead of Node-RED reduced communication latency between ATs and the platform's web interface. At this stage, this time is neglected; however, the preliminary results from the quantitative analysis in [22] showed the need for further efforts to improve response times from the integrated cloud services and their communication reliability. The response API delay from Bulgarian ChatGPT currently averages around 3–4 s, and we started working on implementing response streaming to improve it.

During Phase 3, feedback on usability, logic flow and scenario effectiveness was collected from the pilot deployment in real SLT environments with children and therapists. The TAM and extended SUS questionnaires were employed to identify platform acceptance, usability issues and areas for improvement.

During Phase 4, based on the data collection and analysis, we identified areas for improvement and identified the main principle for conceptualizing the design process to guide future development and replication by other developers.

3.1.1. Developing and Accessing ATLog Therapy Solutions Using a Cyclical Approach

The methodology followed a cyclic process of design, testing and improvement for developing and evaluating platform interventions (solutions) in real-world therapy settings, with previous successful iterations reported in [19–23]. The solutions were compared and assessed to gather data for developing a new framework for conceptualizing the design processes. The iterations were conducted during experiments (which are still ongoing) at a Logopedic Center of South-West University in Blagoevgrad (Figure 2). The research procedure for the experiments was preregistered and approved by the Ethics Committee of the South-West University with protocol N 2410-1/29.10.2024.



(**a**) (**b**)

Figure 2. Pilot experiments at the Logopedic Center of SWU: (**a**) Nao and Furhat cooperation during a playful scenario for 'Transport tools'; (**b**) VR Bingo 3 × 3 on topic 'fruits'.

Therapists actively contributed to the conceptualization and development of interactive scenarios with ATs, ensuring alignment with real SLT practices. At the beginning, they defined various scenarios for vocabulary acquisition and sentence construction with SARs, and scenes and interactions for language learning in the VR environment. Next, they prepared the robot replica to initiate and sustain conversations by programming predefined dialog structures and interactive prompts. These human-like interactions in language comprehension and speech production scenarios were continuously improved, resulting in a key innovation: fostering foundational vocabulary development through intensive but engaging verbal interaction with robots, making the process more enjoyable

and encouraging sustained participation. Building on this foundation, the playful scenarios involve question-and-answer sessions, narration of fairy tales with subsequent analyses, and engaging spoken social communications with SARs or in VR. Through human-like conversations, children practice speaking and listening skills in a supportive, non-judgmental environment—in the secure world of robots and VR immersion. Additionally, children felt less anxious and self-conscious when interacting with a robot that misunderstood them or made jokes. This increased opportunities for practice, including outside of therapy sessions (e.g., at home), because ConvAI can be integrated easily into low-cost robots, such as the emotionally expressive robot Emo developed within the project.

3.1.2. Participants and Setup for the Qualitatively Assessment of ATLog

An online study was conducted to examine, via qualitative assessment, the potential of adopting ATLog platform in SLT practice. ATLog was evaluated by 27 experts using the TAM and SUS questionnaires, along with ten additional questions specifically focused on the ATLog structure and effectiveness. The selection criteria required experts to be practicing speech and language therapists, while excluding therapists in the ATLog project. Eligible candidates needed to have at least five years of professional experience, be actively engaged in clinical practice and work specifically with children and adolescents. The ATLog platform was demonstrated in front of 50 participants; however, only 27 SLT filled the surveys.

The experimental procedure for the SUS, extended with additional questions focused on ATLog functionality, was preregistered and approved by the Ethics Committee of IR-BAS under protocol N7/05.02.2025, while for the TAM—protocol N11/06.06.2025. There was a one-day face-to-face training session, consisting of a 4 h presentation and 4 h of platform demonstrations. At the beginning, each block for different ATs was explained, followed by the design and execution of example interactive scenarios. Two videos from the training were shared with the participants. Additionally, participants received a step-by-step tutorial consisting of four video lessons, covering explanations of blocks in different categories, as well as how to design and run an example scenario in ATLog. These videos are provided in the Supplementary Materials Videos S1–S4.

The average age of the participants was 37 years (SD = 8.37). All of them were female because this profession is practiced mainly by women in Bulgaria. The youngest SLT professional was 25 years and the oldest one was 56 years

3.1.3. Instruments for the Qualitatively Assessment of ATLog

The methodology included the System Usability Scale (SUS) and Technology Acceptance Model (TAM) during the Phase 3 of DBR. The reliability of questionnaires used in the study is presented in Table 1. The SUS evaluates three key dimensions of usability—effectiveness, efficiency and satisfaction. This questionnaire enables the assessment of a wide range of products and services, such as hardware, software, mobile devices, websites and applications. It has several advantages: (1) it is simple to administer, (2) it provides reliable results even with small sample sizes, and (3) it effectively distinguishes between usable and unusable systems. The questionnaire includes 10 items, each rated on 5-point Likert scale ranging from ‘Strongly Agree’ to ‘Strongly Disagree’. They are focused on the usability of the platform, which content can be seen in Table 2. The remaining 10 questions (Table 3) are focused on ATLog structure, communication efficiency and the feasibility of designing interactive scenarios for SLT. The questionnaire evaluation confirmed therapists’ acceptance of integrating ATs into SLT by effectively addressing user-friendliness and ensuring alignment with SLT practices. The responses were rated also via a 5-point Likert scale, ranging from ‘Strongly Disagree’ to ‘Strongly Agree’, consistent

with the SUS questionnaire. Additionally, the survey finished with an open-ended question asking for any recommendations about the ATLog platform.

Table 1. Test results of the reliability of questionnaires used in the study.

Questionnaire	Cronbach's Alpha	N of Items	Description
SUS	0.798	10	Reliable
Design structure (communications and effectiveness in creating interactive scenarios in ATLog)	0.849	10	Reliable
PU subscale	0.906	6	Reliable
PEU subscale	0.922	6	Reliable

Table 2. SUS average scores for each statement.

No.	Question	Mean	SD	Interpretation
1	I think that I would like to use this system frequently.	4.1	0.8	Agree
2	I found the system is unnecessarily complex.	1.4	0.7	Disagree
3	I thought the system was easy to use.	4.3	0.8	Agree
4	I think that I would need the support of a technical person to be able to use this system.	2.4	1.2	Disagree
5	I found the various functions in this system were well integrated.	4.3	0.7	Agree
6	I thought there was too much inconsistency in this system.	1.4	0.8	Disagree
7	I would imagine that most people would learn how to use this system very quickly.	4.4	0.6	Agree
8	I found the system very cumbersome to use.	1.4	0.8	Disagree
9	I felt very confident using the system.	3.6	1.3	Agree
10	I need to learn a lot of things before I could get going with this system.	2.8	1.2	Disagree

Table 3. ATLog Design Structure evaluation average scores for each statement.

No.	Question	Mean	SD	Interpretation
11	I would use ATLog if I had access to video tutorials and a methodological guide.	4.2	1.1	Agree
12	The platform is adequately structured for creating interactive scenarios with AT.	4.4	0.7	Agree
13	Creating scenarios in the Platform saves time.	4.3	0.8	Agree
14	The platform is suitable for managing SAR in real conditions.	4.3	0.8	Agree
15	The platform allows the child to take initiative (through QR code, text, voice).	4.8	0.4	Agree
16	The therapist has control over the AT in the platform during the intervention (stop button, voice command, and touch sensor).	4.9	0.4	Agree
17	The delay in responses from the text-generative AI is acceptable.	3.7	0.8	Agree
18	Each of the graphic blocks in the platform contains information about its therapeutic application.	4.6	0.6	Agree
19	The graphic blocks in the platform are informative enough for their use.	4.4	0.9	Agree
20	The platform supports the creation of personalized interactive scenarios with AT.	4.7	0.6	Agree

The TAM is a widely used model for predicting and explaining user acceptance of technology before implementation. The TAM includes two key components: perceived usefulness (PU) and perceived ease of use (PEU). Both PU and PEU are assessed using six statements, the content of which can be seen in Table 4. Each statement is rated on a 7-point Likert scale, ranging from 'Strongly Disagree' to 'Strongly Agree'.

Table 4. TAM average scores for each statement.

	No.	Question	Mean	SD	Interpretation
PU	1	Using the ATLog platform would improve my performance at work.	6.1	0.7	Strongly agree
	2	Using the ATLog platform would increase my productivity.	6.1	0.7	Strongly agree
	3	Using the ATLog platform would increase my efficiency at work.	6.3	0.7	Strongly agree
	4	Using the ATLog platform would make my work easier.	6.4	0.7	Strongly agree
	5	Using the ATLog platform would be useful for my work.	6.4	0.7	A Strongly agree
	6	Using the ATLog platform in my work would allow me to complete my therapeutic tasks faster.	5.9	1.0	Strongly agree
PEU	7	Learning how to use the ATLog platform would be easy for me.	6.4	0.7	Strongly agree
	8	I could easily use the ATLog platform for therapy purposes.	6.4	0.8	Strongly agree
	9	The user interface to the ATLog platform is clear and understandable.	6.2	0.9	Strongly agree
	10	It would be easy for me to learn to work with the ATLog platform.	6.5	0.7	Strongly agree
	11	I think the ATLog platform is easy to use.	6.3	0.8	Strongly agree
	12	I think it will be easy for me to learn to use the ATLog platform.	6.4	0.7	Strongly agree

The SUS score for each participant was computed through a standardized approach [79,80]. Specifically, for odd-numbered statements, *1* was subtracted from the response, while for even-numbered statements, the response was subtracted from *5*. To normalize the scores and facilitate comparability across different evaluations, the individual scores were then multiplied by *2.5*. The same formula was applied to calculate the score for the second part of the questionnaire to enable a comparative analysis of the results of both. Since all statements in the second part were positively worded, the score for each respondent was derived by subtracting *1* from the response and subsequently multiplying by *2.5*. The average score from the study was used to assess the acceptability of the SUS test, with a *score of 70 applied as the passing threshold*, which was defined in [81]. Finally, the computed SUS scores, along with the scores from the second part of the questionnaire about the overall framework (Design Structure) and how the preprogrammed blocks are organized and communicate with ATs in creating interactive scenarios within the ATLog, were mapped to an absolute grading scale as proposed by [81]. In this scale, *A* corresponds to scores of *90–100*, *B* to *80–89*, *C* to *70–79*, *D* to *60–69*, and *F* to scores below *60*. As recommended by Lewis [82] to compare the results between the SUS and TAM questionnaires, we put PU and PEU on a 0–100-point scale and then averaged PU and PEU to obtain the overall TAM result, using the formula provided in [82]:

$$\text{PU} = (\text{AVERAGE}(\text{Q1}, \text{Q2}, \text{Q3}, \text{Q4}, \text{Q5}, \text{Q6}) - 1) \times (100/6) \quad (1)$$

$$\text{PEU} = (\text{AVERAGE}(\text{Q7}, \text{Q8}, \text{Q9}, \text{Q10}, \text{Q11}, \text{Q12}) - 1) \times (100/6) \quad (2)$$

The results from the questionnaires were analyzed using *IBM SPSS v26* to ensure the reliability of the results. Cronbach's alpha was employed to measure the internal consistency of the scales. Also, the nonparametric Spearman's rank-order correlation test was applied to evaluate the strength of the relationship between the parts of the questionnaires and to determine their correlation.

3.2. Materials

The hardware and software used in the ATLog platform are briefly presented below.

3.2.1. Hardware

Social Robots

- Nao [83]: A small humanoid robot for multimodal interaction through speech, body language, barcode reading and tactile sensors. Nao is one of the most widely used SARs in therapy for individuals with developmental, language and communication disorders [47]. For that reason, the ATLog platform contains more predefined Blockly blocks for Nao compared to other ATs.
- Furhat [84]: Currently the most advanced conversational social robot with an expressive face. Its human presence tracking, visual contact by eye-tracking camera, intuitive face-to-face interaction, and customizable diverse characters and voices contribute to real human-like interactions. The touchscreen interface, enhanced with images and text, further enriches communications.
- Emo, our second version (v2) of the emotionally expressive robot, v1 can be seen [85]: The robot has a 25 cm head that displays emotional states using emoticon-like facial features, with a dynamically changing mouth, eyes and eyebrows. Emo has six degrees of freedom to replicate human head movements. It can track a person in front of it to detect attendance, read QR codes and play audio files, and access cloud services for NLP and NLG. These capabilities make Emo suitable for emotional teaching and multi-robot cooperation in SLT.
- Double3 [86]: A robot for telepresence, designed to provide remote interaction with a dynamic presence. It is also considered as a social robot because it enables assistive interactions with both the therapists and other ATs during telepractice. Double3 has a high-definition camera, two-way audio and video capabilities, the ability to navigate physical spaces, and an interactive display to present text, images and videos. Double3 can scan remote QR codes, thus acting as a mediator in communications during telepractice.

VR Devices

- Headsets: Meta Quest 2, 3, 3S and Pro. Headsets run on the "Meta Quest OS" operating system, which is based on "Android" and can be used independently or connected to a computer via the "Meta Quest Link" program. The Meta Quest Pro headset was used during the experiments due to its eye-tracking.
- Camera for 360° Virtual Tours: XPhase Pro S2 with 25 sensors, each with 8 megapixels, providing a total resolution of 200 megapixels. It supports 360° tours, hot spots and side-by-side view, with 16-bit lossless format output.

Computing Devices

ATLog data processing, communications, storage and online display were conducted on a laptop (11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz, 8.00 GB RAM, MS Windows 11 Prof. 64-bit). The external touch monitor is Dell-P2424HT, 23.8", FHD, IPS, Anti-Glare, with built-in speakers.

3.2.2. Software

- Choregraphe [87]: a multi-platform desktop tool that allows users to create animations, behaviors and dialogs for the Nao robot. It has block-based visual interface for creating, customizing and uploading personalized behaviors, which are then uploaded to the robot for subsequent use through the ATLog frontend.
- NAOqi 2.8 [88]: NAOqi is the operating system for the Nao robot. Through its remote APIs, developers can interact with the robot from external IDEs over a network. These APIs allow users to access features such as speech recognition, motion control, vision

processing and sensor data. Customized Python scripts in the ATLog backend enable these functionalities for subsequent use through the ATLog frontend.

- Furhat Kotlin SDK [89]: A software development kit designed for building interactive applications for the Furhat robot using Kotlin. This SDK is used to program the robot's real-time interactions, including speech, gestures, and dialog, as well as tablet-based interface to Furhat.
- Furhat Remote APIs [90]: The remote APIs enable developers to interact with the robot from external IDEs over a network. Using these APIs, the robot's movements, facial expressions and voice can be controlled remotely through the ATLog frontend.
- Double3 Developer SDK [91]: The D3 APIs, commands and events enable developers to use Double3 in developer mode. Thus, python scripts run directly in the Double3 operating system. The D3 software includes a built-in subscription/publishing system, allowing Double3 to send event messages or respond according to its programmed skills.
- EMO software: The control system of the robot is based on two Raspberry Pi 5 units and other Arduino-type microcontrollers. Running on Linux, it supports any compatible programming language.
- VR software for interactive scenarios in web environment: HTML, CSS and JavaScript for structuring the content, styling and layout. To enable virtual reality within the browser, webXR technology [92] is used to manage VR devices and handle controller inputs through an API that triggers events and interactions within the VR environment. Additionally, the THREE.js library, built on top of WebGL [93], accesses the underlying graphics hardware to render 3D content directly in the browser.
- 3DVista [94]: Software for 360° Virtual Tours enables the creation of interactive environments through integrated elements such as clickable hotspots, navigation arrows, and objects that are activated based on predefined therapeutic goals and outcomes. The virtual tours include components designed to support the development of spatial awareness, orientation and logical thinking. Specific interactive features for children are included to facilitate navigation within environments and engage them through age-appropriate, game-based scenarios tailored to their individual needs.
- Cloud services for speech recognition and speech synthesis from MS Azure and Google and services for text generation from NLP Cloud [95], ChatGPT-4 of OpenAI [96], BgGPT of INSAIT [97].
- Express.js server [98]: A Node.js web application framework for developing the client–server model. Express.js is used to build the backend in ATLog, handling logic, communication and data processing.
- Visual programming interface, built on Google Blockly [99]: block-based programming interface allowing users to design graphical flows of blocks without programming skills.
- The internal memory, built on a Node.js object repository, stores and retrieves key-value pairs and acts as a temporary data storage for short-term operations.
- SQLite database [100]: persistent data storage for long-term data management to track and access the progress of the individual SLT interventions.

3.3. ATLog Platform: Design and Features

The platform architecture with network design solutions for the integration of SARs, VR devices and ConvAI in one platform is presented below. The integration of a block-based visual programming framework with Express server is presented, as well.

3.3.1. Platform Architecture

Figure 3 presents the architectural scheme of the ATLog platform, where a web-based interface facilitates the intuitive design of interactive scenarios, while Express server serves as the central backend component, managing communication with the frontend and among assistive technologies by processing incoming requests and responses. Express server processes incoming requests and responses through RESTful endpoints using standard HTTP methods and exchanges data in JSON format. However, it also includes some non-REST features, such as real-time streaming and command execution. The block-based graphical editor in the frontend removes the need for programming skills, enabling users to define a structured sequence of blocks (the scenario flow) that send requests to server endpoints in order to activate SAR skills, VR interactions, or NLP/NLG cloud services. Internal memory on the middleware enables data exchange among ATs, supporting cooperation between SARs, ConvAI and VR. For instance, text-generative AI (BgGPT) uses the physical presence of a Nao robot for QR code reading. The decoded text is saved in memory and later used for questioning BgGPT.

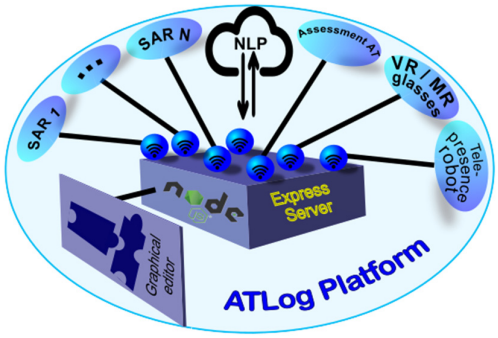


Figure 3. Architecture scheme of the ATLog platform.

3.3.2. Communication Protocols

- In Express server: standard HTTP protocols, along with child processes to access skills of SARs and VR scenarios. RESTful HTTP requests (GET, POST) are sent to cloud services for NLP and NLG. WebSocket protocol enables bidirectional communications in VR scenarios.
- In Blockly: HTTP GET and POST requests are used to send commands to Express server or external web resources.
- For file transfers: Files are securely transferred between the platform and the Nao robot via SCP (Secure Copy Protocol) over the LAN network, using SSH (Secure Shell) for encryption and authentication.
- For remote script execution on Nao and Furhat: TCP sockets are used for communication from the Express backend to NAOqi (on Nao) over IP, while for Furhat we use the Python SDK wrapper, which connects to the Furhat API via WebSocket.
- For remote script execution on Double3: python scripts are remotely executed using an SSH client via PuTTY. A developer mode and additional administrative rights are required to activate the camera and subscribe to events for Double3, such as reading QR codes (DRCamera.tags) using the command "camera.tagDetector.enable".
- For MQTT connections to Emo robot: Remote control and execution of scripts on the Emo robot by using the mqtt protocol within a PY 3.x virtual environment and paho-mqtt library. On the Raspberry Pi, a script runs as a communication bridge between the Raspberry Pi and the ATLog server.
- For database writing, the protocol is SQL, executed via the SQLite3 package for Node.js.

3.3.3. Network Solutions for the Integration of Block-Based Visual Programming with Express Server

Express server handles communication between Blockly graphical blocks and SARs or VR applications using HTTP, TCP, SSH, Websocket, SCP and SQL protocols, ensuring bidirectional and real-time responses. The server manages child processes and SSH connections in order to integrate the ATs in the platform. Dependencies that need to be installed comprise 'child_process' and 'ssh2' Node.js modules for process management and remote connections. The endpoints that use child processes execute server-side python- or JavaScripts for controlling robots and VR or calling NLP/NLG cloud services. This network solution ensures easy maintenance, since *only the scripts, which are used in the child processes, need updates when technology evolves*. Additionally, the ATLog interface uses HTTP requests to connect to external text-generative AI APIs, bypassing security limitations on unauthenticated devices like robots and VR glasses. The network solutions for integrating pre-programmed blocks with their corresponding endpoints in Express server are detailed in Section 4 and can be seen in Appendix A, Tables A1–A5.

- On the frontend (Blockly): ATLog sends HTTP GET and POST requests to the server to initiate the skills of ATs, retrieve data or send data to the internal ATLog repository. To ensure the correct execution of the next block in the scenario flow, specific handling of the backend response is mandatory for certain blocks.
- On the backend (Express server): ATLog uses standard TCP or HTTP protocols, along with child processes and several middleware, to access the skills of SARs and VR scenarios. It sends RESTful HTTP requests (GET, POST) to cloud services for NLP and NLG. Backend endpoints return responses to the frontend as needed.
- The VR endpoints on the backend run on a separate port, enabling bidirectional communication between the client and server using the socket.io library. Socket.io enhances the Websocket protocol by implementing organized communication spaces.
- VR streaming is provided through Meta's Meta Quest casting service [101], allowing the VR headset screen to be shared on an external monitor for observation by the therapist or other children.

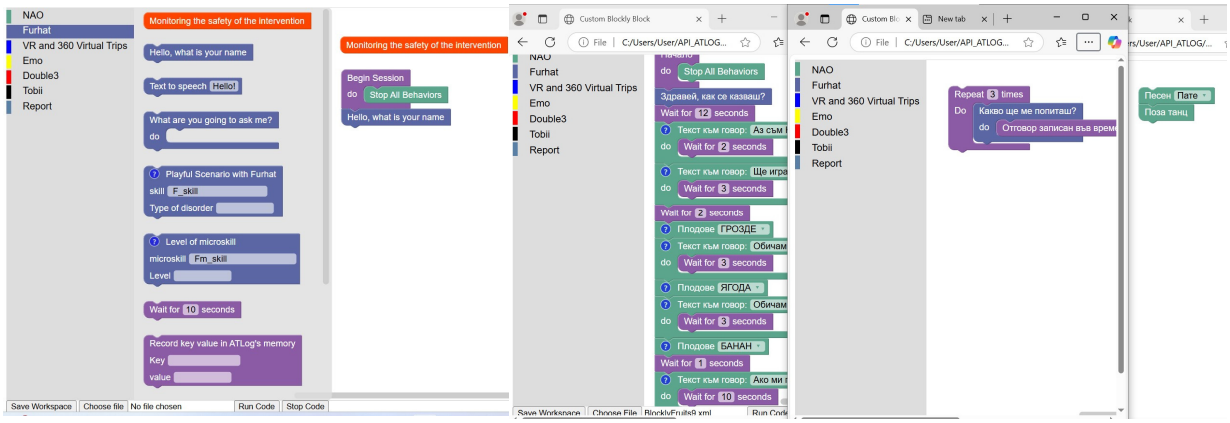
3.3.4. Cooperation of ATs

Cooperation between Nao, Furhat, Emo and Double3 robots enhances the SLT session by enabling multimodal interactions, user presence detection, emotional responses and telepractice features, which are either lacking or less developed in individual robots. For example, the cooperation between Nao and Furhat alters the questioning approach, with Nao using QR codes for nonverbal children, while Furhat provides an advanced verbal recognition system. Furhat can further enhance Nao, Emo and Double3 robots by enabling them to subscribe to FurhatOS events and receive notifications when Furhat receives a new oral question from the child. Through calls to Furhat Remote API, these robots can also listen, speak and attend to children from external IDE. Emo can contribute by replicating child head movements and expressing its current emotions, making sessions more engaging and fun with natural, human-like mimicking. Emo can read QR codes in Bulgarian, a language not natively supported by Nao. Nao reads QR codes in English, requiring translation via cloud services, which inserts latency. Double3 can contribute to each robot through remote interactions with them during telepractice. According to our experimental test, when the child's face is displayed via video telepresence on the Double3 tablet in front of Furhat's camera, the child is detected from Furhat as active user and a unique ID is assigned, which remains fixed as long as the Double3 tablet stays in view of Furhat. This automatically initiates Furhat's transition from an 'idle' to an 'active' state, following the steps defined in the designed interactive scenario for the current telepractice.

3.4. Design Process of Interactive Scenario

To streamline the design of interactive scenarios, the ATLog frontend consists of preprogrammed graphical blocks that can be combined based on the existing ATs and logopedic materials. The platform employs a subprocess-oriented approach, breaking down SAR skills into microskills and VR interactive scenarios into smaller components, allowing the therapist to choose and combine microskills from different ATs. The blocks are assembled in a flow using a drag-and-drop interface.

A snapshot of the ATLog graphical interface is shown in Figure 4a. On the left, different colors represent various categories of ATs that are integrated into the platform. Nao's blocks are in green color, while the Furhat category is in dark blue. By the security block in the Furhat category (the red one in Figure 4a), all running ATs in the platform can be stopped. Thus, during the session, the Furhat robot listens for keywords like "stop" or "break" and terminates all spawned child processes, stops robot behaviors, closes open databases and shuts down Express server. Alternatively, the platform can be stopped using the '*Stop Code*' button (Figure 4a). The purple blocks are related to the ATLog internal repository, NLP/NLG cloud services, or logic blocks, one of which is the loop block 'repeat_execution' presented on the middle window in Figure 4b. It has an input parameter for the number of iterations and wraps any inner blocks inside '*for loop*' block to execute them multiple times. Thus, therapists determine how many times to repeat the execution of one or more blocks. Several instances of the ATLog block-based graphical interface can run separately, as shown in Figure 4b. This allows the therapist to partition the design of the scenario into parts, and by opening them in different tabs in a browser, they have the option to repeat some of the parts or to switch immediately to a certain part.



(**a**) (**b**)

Figure 4. (**a**) ATLog block-based graphical interface for visual programing with opened category for Nao robot; (**b**) three instances of ATLog block-based graphical interface running separately.

Each block has a pop-up window, accessible by pressing the question mark, which provides explanations of the therapy objectives, treatment domain, type of disorder and language skill level. Most of the blocks are hierarchical (structured with a drop-down menu) in order for therapists to select from predefined options. For example, a drop-down menu for the skill "Animals" presents choices corresponding to the hierarchical behavior '*WORDS_anim*' uploaded onto a Nao robot, which links to 9 sub-behaviors for different animals. Similar hierarchical blocks are: '*Ling 6-Sound Test*' (A, M, I, U, SH, S); '*Transports*'; '*Colors*'; '*Musical Instruments*'; '*Emotions*'; and '*Store*'. These blocks await a QR code on a double-sided flashcard illustrating the content, which should be shown to Nao. If the child selects the correct card, Nao responds with approval phrases and proceeds to the next block. If the child selects the wrong card, Nao waits for the correct one.

Nao can be asked a question by dragging '*TextQuestionBlock*'. The context and question must be specified in the block's fields. The default context is: "Say the information in a child-friendly way". For nonverbal children, an alternative option is to use '*QRquestionBlock*' by showing a QR code on a double-sided flashcard illustrating the question content. To cope with interactions not natively supported by Nao Bulgarian language, we use cloud-based TTS and ASR engines to convert text into audio, which is then dynamically uploaded to the robot's memory. By '*Text_to_speechBlock*', Nao plays the input text using an audio file, while by '*PlaytextBlock*', Nao plays the audio answer converted from Bulgarian ChatGPT (BgGPT). BgGPT provides a response in the backend and saves the answer in the ATLog repository. The '*FurhatQuestionBlock*' can be dragged to operate in parallel, allowing the child to continuously interact with Furhat during the session using prompts like 'Furhat, tell this?', which encourages active listening and speaking.

Figure 4a illustrates a block with 'do' input. Such blocks differ in how they handle responses and manage response data, with 'do' input blocks (doCode) allowing for additional actions to be triggered based on received responses. For example, the blocks for TTS and Q&A send requests to NLP/NLG cloud services and must wait for responses, using 'doCode' input blocks to handle text-to-speech synthesis.

Two types of temporary memory can be used: the one initialized when starting the ATLog and the other being the Nao robot's native memory. Blocks in green read or write in the Nao internal memory, while blocks in purple read or write in the memory of ATLog. Each AT has in its category a '*RepositoryRead*' block, which retrieves a value from the ATLog repository using a specified key, and a '*RepositorySave*' block, which stores a value in the repository under a specified key. The persistent memory of the ATLog is a database and exists as a standalone category. Using the '*DB_save*' block, the therapist can save session observations and notes at the end of the therapy session. The information is anonymous, as each child is assigned a unique identity code. The records in the database can be exported in *csv* format.

ATLog frontend allows for the integration and execution of new skills through *"custom"* blocks. Anyone experienced in creating new behaviors in Choregraphe or programming via remote NAOqi can integrate new skills using two Nao custom blocks with input fields. The '*CustomBehavior*' block accepts the custom behavior name for behaviors designed in Choregraphe and uploaded to the NAO robot, while the '*CustomRemote*' block allows the execution of custom scripts remotely on NAOqi, enabling remote control of the NAO robot's capabilities. Similarly, for Furhat, new skills or microskills can be seamlessly integrated into the ATLog frontend using the '*CustomSkillBlock*'. It runs a custom skill prepared in Kotlin SDK and uploaded to Furhat robot.

More explanations for blocks in the different categories are provided in the Supplementary Materials Videos S1 and S2.

3.5. Ethics

The design and implementation of the ATLog platform raised ethical considerations for both developers and therapists, specifically regarding the assessment of risks linked with using AI systems. Although the European AI Act classifies conversational AI systems as having a low level of risk, ATLog adheres to its own Ethical Codex [102]. An ethical framework was established within the methodology to analyze potential risks associated with using text-generative AI, personal data and the protection of individuals' human rights. Throughout the development, ethical guidelines were followed to ensure data privacy and security through anonymization.

4. Implementation and Results

Following the proposed methodology in Section 3, this section provides the implementation of the ATLog platform. Technical details for developers on how AT devices and services interact within a client–server model can be seen in Appendix A (Tables A1–A5) and Appendix B (Table A6). This section also provides an example of how to design and execute an interactive scenario step-by-step using the platform. The final subsection presents the results of the ATLog platform usability assessment, including user-friendliness, functionality and technological acceptance as evaluated by the therapists.

4.1. Implementation of ATLog Platform

Figure 4 shows the implemented frontend, developed using the Blockly web-based visual programming framework. The frontend blocks control ATs by sending parameterized requests to the endpoints of a Node.js-based Express server, as detailed in Appendix A, Tables A1–A5. Table A6 in Appendix B provides an overview of Express server setup with the required dependencies, middleware and dynamic imports. The middleware section in Table A6 describes the processing of JSON and URL-encoded request bodies, along with specific configurations for serving static assets, including Three.js libraries for VR web-based integration and rendering 3D graphics in the browser using WebGL.

Server endpoints manage AT control by processing API requests from the frontend, executing pre-programmed scripts for certain ATs via child processes, or by using SSH connections and the MQTT protocol. Tables A1–A5 detail the relationships between block names; their functional descriptions; the required skill, microskill, or other files to be uploaded on the ATs; the input parameters; and the corresponding backend endpoints

Table A1 illustrates Nao blocks. When the user drags a Nao block in the Blockly workspace, it sends a request to the server endpoint, which executes either a behavior uploaded on Nao or 'child_process', which runs remote Nao API. Nao blocks activate specific endpoints (/nao1, /nao2, /nao3 or /naoUpload), forming a URL with input parameters. Such an example is http://serverIP:3000*/nao1?IP=${ipNao}&pyScript=${pyScript}*, where the *pyScript* parameter specifies the name of the PY 2.7 script to be executed on the Nao robot. Another example is http://serverIP:3000*/nao2?IP=${ipNao}&behname=${behname}*, where the *behname* parameter specifies the name of the uploaded behavior on the Nao robot to be executed. Some Nao blocks require the server endpoints to activate a virtual environment *(/venv)* to execute Python 3.x scripts, including those for NLP/NLG cloud services. The endpoint http://serverIP:3000*/nao3?IP=${ipNao}&filename=${filename}* specifies the script name to be executed. Some Nao blocks require input parameters such as port, Nao IP, password, file path on the server, file path on the robot and file name. One such endpoint is/*naoUpload*, which constructs a *pscp* command to upload the specified file from a local directory to the Nao robot via SSH.

Table A2 shows Furhat blocks that link endpoints using child_process to execute Python scripts that interface with Furhat remote APIs. The */furhatRem* endpoint processes a GET request with the input parameters *${ipFurhat}*, *${pyScript}* and *${text}* (if TTS is required). The server captures the script output from 'stdout' and returns the response to Blockly, ensuring correct continuous block execution.

Table A3 summaries frontend blocks for Emo and Double3. The/*emo1* endpoint establishes a MQTT communication bridge to the Raspberry Pi 5 of Emo for remote control through a paho-mqtt library for message exchange. Different scripts run on the Emo robot, such as those for QR code scanning or playing audio, in response to block requests. The/*emo2* endpoint opens another type of MQTT session to the robot with different topics, such as 'audios', 'emotions', etc. The endpoint/*double3* opens an SSH session to the robot,

requiring authentication, and runs shell commands and PY scripts in the remote robot environment, which interact with the Double3 Developer SDK, D3 API and events.

Table A4 summaries the general frontend blocks and their corresponding backend endpoints for functions not related to SARs and VR, such as '*RepeatBlock*', '*RepositoryReadBlock*', '*RepositorySaveBlock*', '*DB_saveBlock*', '*BG_GPT_Block*', '*QABlock*', and '*WaitBlock*'. A global repository, established for storing text in key–value pairs, is formatted in JSON (see established middleware in Table A6). It also transmits messages among robots and VR, enabling multi-robot cooperation scenarios where each robot's unique features are utilized. Since the Nao robot has a reliable and remote event subscription mechanism, all robots have a block in their category to declare events, raise events, subscribe to events and receive messages by callbacks, as well as to set and receive data to the internal Nao repository via the NAOqi framework. For instance, in order to ask for cooperation with the Nao robot, Furhat has to subscribe remotely to NAOqi Message Broker and QRevent in order to read the QR text that is in Nao memory. Robots can also be subscribed to Nao for audio playback cooperation through remote calls to NAOqi APIs or for TTS. This is useful when the robot does not have embedded TTS, such as Emo, or if TTS is not native for a specific language, such as Nao. Our solution is to use cloud service to obtain an audio file from the text and to play it on the robot, implemented by using gTTS (Google Text-to-Speech), a Python library, and a CLI tool that interfaces with Google Translate's text-to-speech API, generating spoken MP3 data and saving it to a file for playback on the robot. The synthesized audio file is uploaded to the robot using the PuTTY pscp network protocol. The Furhat robot comes with credentials for speaking voices like PollyVoice and Azure voices; however, the Bulgarian language is not supported with PollyVoice. We use personal Azure credentials; however, if they are unavailable, gTTS can also be used through Furhat's remote APIs, with the synthesized audio file uploaded to the web as Furhat plays files from a URL.

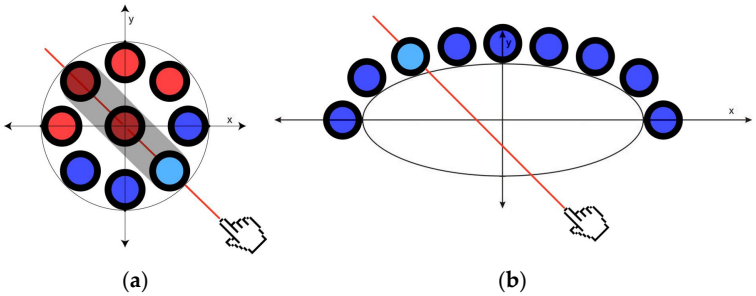
Table A5 summaries the VR frontend blocks. The '*Setup_VR_Session_Block*' sets up the WebSocket communication and assigns unique session IDs for both the therapist and the child. The '*Setup_VR_Scene_Block*' initializes a VR scene in Three.js, configuring the camera, VR controllers, renderer, and WebXR support and loading 3D models, while delivering static files such as frontend code, images, and styles from the "public" folder and dynamically generating HTML pages. The '*StartVRbutton_block*' activates the VR button in immersive mode within the browser. The '*360_Virtual_Tour*' block has a drop-down menu and fetches an interactive 360° virtual tour on a touchscreen, allowing children to explore familiar real-world settings (e.g., supermarket, house, museum) hosted on the ATLog project website. In the same way, the '*360_Virtual_Game*' block provides interactive 360° virtual games on a touchscreen, where children search for and identify objects from a preassigned list.

To streamline the design of blocks that must always be connected, a Blockly event function is utilized. This function is triggered automatically whenever the first block (block1) is moved or created in the Blockly workspace, providing information about the event type (e.g., MOVE or CREATE), the dragged block and its ID. The sequence is ensured by verifying that block1 has block2 connected as its previous block. If block2 is missing, the function automatically generates and links block2 for the correct sequence. An example of such a block is '*Song*', connected in sequence with '*PoseDance*' (shown in the right window of Figure 4b).

4.2. Technical Challenges

- A virtual environment for different python versions must be activated through the 'venv' endpoint because the child processes for text translation, text generation or speech synthesis rely on PY 3.x or higher.

- We faced challenges with continuous block execution in Blockly. While child processes were still running, the run code in the JavaScript of the block had exited, causing the next block to start early. The solution was to use an *async* function with *await new Promise(resolve)*, which either waits for a server response indicating the child process has exited with code 0, or continuously checks the status of the child process by sending requests to a */monitor* endpoint. The second approach works better for blocks with more complex run code. When Nao blocks need to wait for Furhat child processes to finish, an *async* function continuously checks the status of the python process by sending requests to the */monitor/pyFurhat* endpoint. It waits for a response from the server, which indicates whether the Python process is still running or has been exited. Once the process is no longer running, the function exits the loop, and Nao blocks can start. Similarly, */monitor/pyNao1* and */monitor/pyNao2* endpoints allow for client-side functions to determine the child's status when Furhat blocks wait for Nao child processes to exit.
- Another challenge was that by default, Express sets a request body size limit of 100 KB when using the *express.json()* middleware. To handle larger payloads, such as saving responses from chatGPT in KEY=answer, the limit had to be increased to at least 500 KB by configuring the middleware to *express.json({limit: '500 kb'})*.
- We faced challenges with delay in the API response time of BgGPT. To address this, we experimented with response streaming using Server-Sent Events (SSEs) over an HTTP connection. We enhanced performance by delivering tokens in real time through an Express endpoint, which requests a streamed response from the BgGPT API and pipes the received chunks directly to the SARs TTS services.
- From the pilot experiment, where we tested interactive scenarios for stuttering using Furhat, we found that the embedded Furhat's speech recognition based on ML occasionally misinterpreted stuttered speech, sometimes recognizing it as fluent, while other times as '*NOMATCH*' (not fluent). The current solution relies on the SLT practitioners verbally instructing Furhat by saying "fluent" or "not fluent" to classify the stuttered speech correctly.
- We faced challenges with object selection and tracking the direction of the child's hand in VR Bingo (Figure 2b). In the previous DBR iteration, the objects (fruits) were placed on a round table (Figure 5a) and the children had difficulty changing the perspective in order to select the objects that were behind another one. The solution was to expand the round table along the *X*-axis in an ellipse and rearrange the objects to prevent overlap from the initial perspective (Figure 5b). Additionally, the invisible boxes representing the "hitbox" were initially too small and were adjusted to match the height of the objects (fruits), ensuring easier selection. The white ray used to track the direction of the child's hand was replaced with a longer and thicker red cylinder, significantly improving tracking accuracy.



(**a**) (**b**)

Figure 5. Object selection challenges: (**a**) fruits placed on a round table; (**b**) fruits arranged in an ellipse layout. The red fruits (circles) are not reacable; blue means reacable objects.

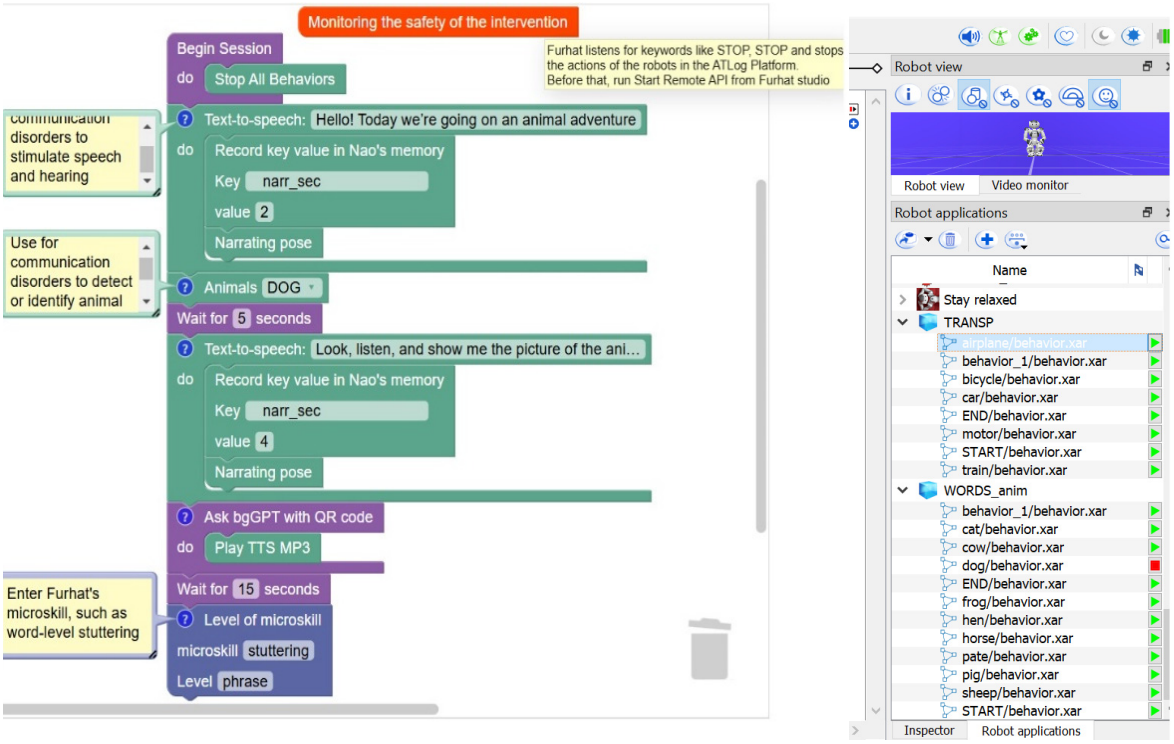
4.3. Example Scenario, Development Steps and Runtime

Figure 6 presents an example scenario, written by an SLT practitioner, which needs to be designed using the ATLog frontend. Figure 7a illustrates the scenario flow of blocks. The block responsible for the first three rows (1–3) is the Furhat block for greeting the child, listening to its name and saving it in the internal repository. Then, Nao starts guiding the playful exercises by asking oral questions about looking, listening and showing the correct flashcard (rows 4–7). For speaking the text in row 4, the used block is the Nao 'TTS block'. The block automatically connected two blocks in the 'DO' input: (1) a narrating block and (2) a block specifying the required seconds for narrating, which are recorded in Nao's memory. A cloud-based speech recognition and TTS from Google engine were used, with a dedicated Nao block responsible for playing the converted response in audio format. This audio was dynamically uploaded to the robot's internal memory. The block responsible for rows 8–10 was '*Ask BgGPT with QR code*'. It waited for the child to show the QR code of the flashcard to the robot's eyes. BgGPT processed the request in the backend and saved the response in the repository. The default context was: 'Say in a child-friendly way". The block 'Ask BgGPT with QR code' automatically connected 'playtext_block,' which played the audio file containing the answer from BgGPT. For raw 12, the Furhat 'Ask BgGPT' block had to be dragged and placed in the 'repeat_execution' block with input parameters for three iterations (as shown in Figure 4b). For row 14, the Furhat 'microskill' block should be dragged, with 'stuttering' as the input parameter and 'phrase' as the selected level. During the interactions, a child can always ask "Furhat, tell this", prompting BgGPT to generate natural language responses, thus motivating the child to actively listen and speak.

Scenario #1 Theme: "Animals";
1. Furhat: "Hello! I'm Furhat. What's your name?";
2. Child: "My name is <name>";
3. Furhat: "Nice to meet you, <name>!";
4. Nao: "I'm Nao. Today we're going on an adventure! Are you ready?";
5. Child: "Yes!";
6. Nao: "We're going to play with different animals. Look, listen, and show me the picture of this animal (dog is selected)";
(Therapist positions two cards of a dog and a cat with QR code in front of the child.)
7. Child: Child shows the dog to Nao eyes.;
8. Nao: "Well done! If you show me any of these pictures, I'll tell you more."
9. Child: Shows the QR code of the cat to Nao eyes;
10. Nao: Reads QR, asks BgGPT and tells about the cat in a childish style;
11. Pause=10sec;
12. repeat (iteration times =3){
Furhat: "What more will you ask me?" Waits for a verbal question…
Child: Asks a question.;
Furhat: Furhat Action: Ask Q→Get response from BgGPT→Tell the answer: A
} until iteration==0 ↻
13. Pause=10sec;
14. Furhat: "Let's play something else! See animal picture on my tablet and tell what they eat?"; (microskill 'stuttering' with level 'phrase' are selected).
15. Nao: "Well done! You know the animals";

Figure 6. Example scenario #1 theme: "Animals".

Figure 7b illustrates the running behavior uploaded on Nao for 'WORDS_anim,' with the sub-behavior (microskill) 'dog', which was triggered by the 'Animals' block in the frontend. Videos S3 and S4 in the Supplementary Materials provide a detailed explanation specifically demonstrating the development steps and runtime of the scenario presented in Figure 6.



(**a**) (**b**)

Figure 7. Scenario flow and activation (**a**) Developed in the ATLog frontend scenario #1. Skill block: "Animals", microskill 'DOG'. (**b**) Uploaded to robot Nao skill for 'WORDS_anim', the running 'dog' child behavior triggered by the corresponding Blockly block.

In addition to the example scenario presented in Figure 7, therapists using ATLog can design a wide range of interactive therapy activities that integrate speech, listening, visual cues, and AI-based dialog, as well as supporting phonological training, vocabulary development and social communication skills training in 2D or 3D VR environments. These activities include personalized interactions based on the child's name, object recognition tasks using QR codes, question–answer dialogs driven by NLP/NLG cloud services, and exercises such as "stuttering" or "phrase repetition". Activities can also trigger microskills within scenarios using various block types—such as sequential, parallel, and asynchronous blocks—that support diverse response-handling strategies. '*DoCode*' input blocks further allow the execution of specific actions based on received responses, such as playing the audio from ASR, narrating instructions, or prompting questions. The platform enables blocks repetition, setting time durations and combining multimodal inputs (e.g., visual, verbal and tactile).

4.4. Results from the TAM and the SUS

The results from the online study, conducted to qualitatively assess the potential of using the ATLog platform by SLT professionals, along with an overview of the questionnaires, are presented in four figures and five tables.

The consistency and accuracy of all the questionnaires used were tested using Cronbach's Alpha. The researchers employed this method to determine the minimum reliability value of each construct, which is considered reliable if the Cronbach's Alpha value exceeds 0.60. The results in Table 1 show that the Cronbach's Alpha value of 0.798 confirms the validity of the SUS questionnaire. In terms of the Design Structure evaluation in the ATLog platform, the Cronbach's Alpha value is 0.849, indicating very strong reliability. The TAM questionnaire shows a very good consistency, with Cronbach's Alpha values of

0.906 and 0.922, respectively, for the PU and PEU subscales. These results reveal that all the questionnaires used for the study were consistent and trustworthy.

The mean score of the study was used to assess the acceptability of the SUS, with a score of 70 generally considered as the threshold for acceptable usability. The SUS survey data revealed an average score of 78, indicating that participants collectively perceive the ATLog platform as meeting acceptable usability standards. When interpreted using an adjective rating scale, this score corresponds to a "Good" level of usability. Furthermore, the results from the Design Structure evaluation within the ATLog platform yielded an average score of 86, indicating its features are acceptable.

The calculated SUS scores, along with the scores from the second part of the questionnaire, were assigned to an absolute grading scale as defined in [81]. In this scale, "A" corresponds to scores ranging from 90 to 100, "B"—to scores between 80 and 89, "C"—to scores between 70 and 79, "D"—to scores between 60 and 69, and "F"—to scores below 60. The results for the SUS are illustrated in Figure 8. There were four SLT experts (15%) who evaluated the usability of the ATLog platform with a score between 52.5 and 57.5 (F). Eleven % of the participants gave Grade D. Most of the experts (74%) evaluated the usability of ATLog platform with scores over 70—Grades C (18.5%), B (25.9%) and A (29.6%).

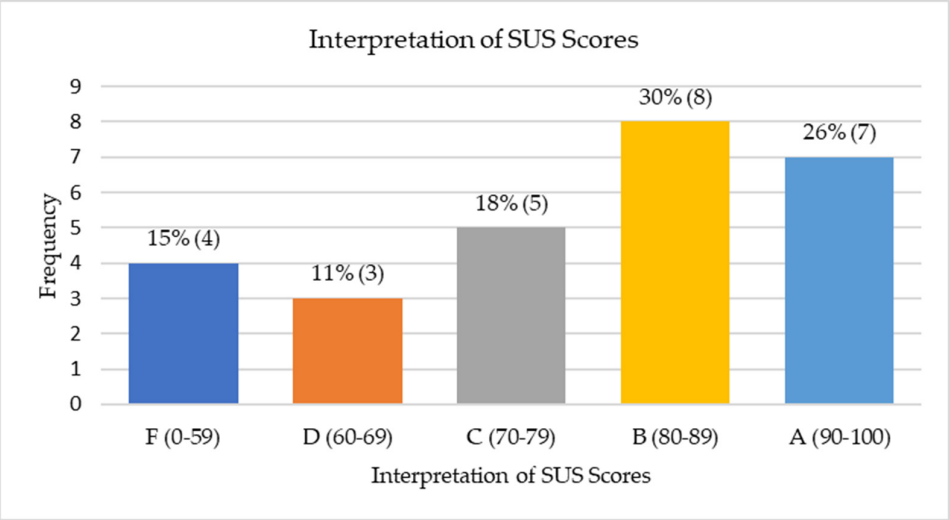


Figure 8. Absolute grading scale for the SUS scores.

The results for the Design Structure evaluation of the ATLog questionnaire are presented in Figure 9. There was only one expert who gave Grade F for the Design Structure evaluation of the ATLog platform; two experts rated it with Grade D and three gave it Grade C. Scores ranging from 80 to 89 were given by 25.9% of the therapists. Scores ranging from 90 to 100, corresponding to Grade A, accounted for 51.9%, indicating that the majority of experts gave the highest score for the platform.

As shown in Figure 10, the PU subscale scores of the TAM concerning the ATLog platform show different evaluations. A small proportion of professionals (11%; n = 3) rated the platform's usefulness as Grade D, while two (8%) assigned it Grade C. Ten professionals (37%) evaluated it more favorably as Grade B. The highest level of perceived usefulness—Grade A (scores between 90 and 100)—was reported by 44% of the therapists, indicating a strong positive assessment by nearly half of the respondents.

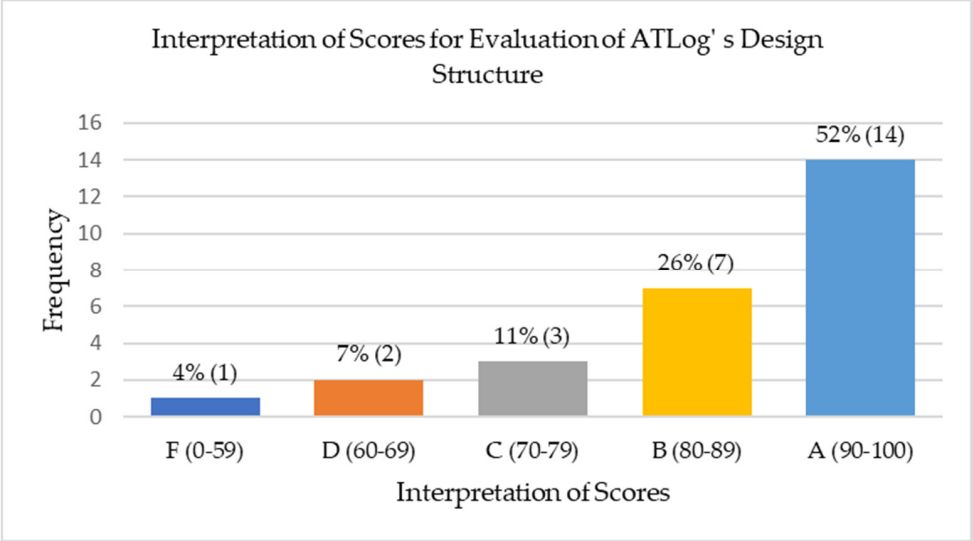


Figure 9. Absolute grading scale for the Design Structure evaluation of the ATLog platform scores.

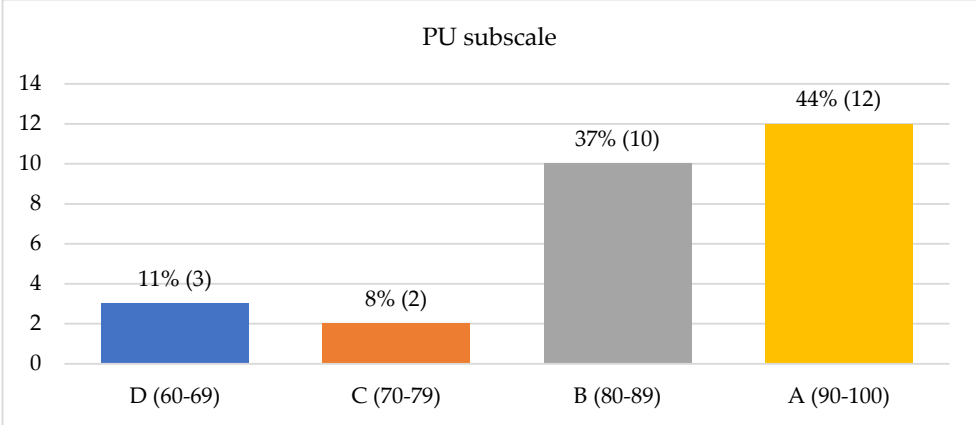


Figure 10. Absolute grading scale for the perceived usefulness subscale scores of the TAM.

The findings related to the perceived ease of use subscale, as illustrated in Figure 11, reveal a generally positive evaluation of the ATLog platform. Three professionals (11%) assigned it Grade D, indicating lower ease of use, while two (8%) rated it as Grade C. Additionally, five experts (18%) provided it Grade B. The majority (63%) rated the platform within the 90–100 score range, corresponding to Grade A. This finding indicates a broad agreement among speech and language therapists concerning the ATLog platform's high level of perceived ease of use.

In the Likert scale assessment, a score above 3.00 is classified as positive (agree), while a score below 3.00 is considered negative (disagree). The SUS method adopts a balanced approach by incorporating five positive and five negative questions to assess both the strengths and weaknesses of a product, thereby minimizing response bias and enhancing the validity of the research findings. This methodological approach ensures a more objective and reliable evaluation of product usability by mitigating the risk of misinterpretation and inaccuracies in data analysis [80].

The average scores of each statement of the SUS survey are presented in Table 2. Based on the results, all odd-numbered statements (1, 3, 5, 7, 9) received scores above 3 (from 3.6 to 4.4), indicating general agreement with these items. These positive responses indicate that the ATLog platform is perceived as easy to use, well-integrated, user-friendly, and efficient by the SLT practitioners. In contrast, all even-numbered statements (2, 4, 6, 8,

10) received scores below 3 (from 1.4 to 2.8), indicating disagreement among respondents. These responses suggest a low perceived level of complexity and a minimal need for technical support when using the platform. Also, the ATLog platform is regarded as easy to learn, consistent and user-friendly.

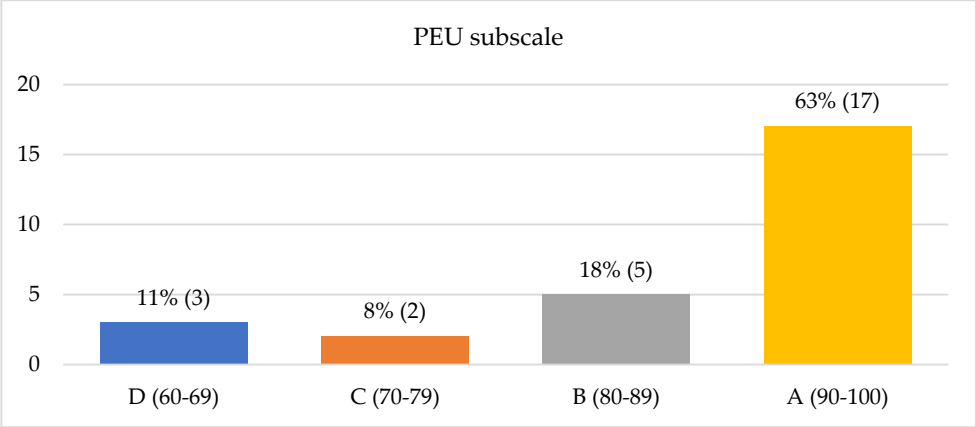


Figure 11. Absolute grading scale for the perceived ease of use subscale scores of the TAM.

The results according to the average scores of each statement of the questionnaire about the Design Structure evaluation of ATLog platform are accessible in Table 3. All participants consistently agree with the positive aspects of the platform. The average scores were generally from 4.2 to 4.9, which suggests that professionals find the ATLog design user-friendly, well-organized and engaging. They particularly like features such as the child's ability to take initiative, the therapist's control over actions in the ATLog platform, and the personalization options for creating interactive scenarios. The platform is also seen as being adequately structured for managing scenarios and as informative and providing sufficient details for both therapeutic applications and interactions between the therapist, the ATs, and children and adolescents with CDs. These highly positive responses suggest that the design features of the ATLog platform are well-perceived and meet the needs of SLT practitioners, offering a high level of usability, individualization, control and functionality. While most responses were strongly positive, the statement for question 17 received a score of 3.7. As mentioned above [80], when the score is greater than 3, the statement is positive, indicating that professionals perceive the text-generative AI's reaction time to be acceptable. Although this remains in the positive range, it suggests a mild concern regarding response time. Therapists are generally accepting of the delay, but this result highlights an area for potential improvement, especially in contexts where real-time interaction is critical.

The results for the mean scores of the TAM are presented in Table 4. All statements in the TAM questionnaire for both PU and PEU subscales received mean scores above 5.9, which indicates that the professionals mostly find the ATLog platform useful and easy to use. The statement "*Using the ATLog platform in my work would allow me to complete my therapeutic tasks faster*" was rated with the lowest score among all items (5.9), but it was still rated with "Strongly agree" and shows nuance in the SLT practitioners' opinions. The highest score (6.5) was about the statement "*It would be easy for me to learn to work with the ATLog platform*". Standard deviations ranged from 0.7 to 1.0, indicating consistent responses and a strong positive perception of the ATLog platform's usefulness and ease of use.

A correlation analysis was conducted to explore the relationships between the SUS, the Design Structure evaluation of the ATLog platform, and the TAM (represented by PU and PEU subscales). A summary of the results about Spearman's rho correlation analysis is presented in Table 5. The findings show a Spearman's rho correlation coefficient of 0.450, which indicates a moderate positive correlation between the two questionnaires, the SUS

and Design Structure evaluation of the ATLog platform. This suggests that as values in the SUS increase, values in the Design Structure evaluation of the ATLog survey tend to increase as well. The p-value of 0.018 is less than 0.05, meaning that the correlation is statistically significant ($p < 0.05$). This indicates that professionals who perceived the platform as well-structured were more likely to report higher system usability. However, the Design Structure evaluation of the ATLog platform did not significantly correlate with PU (r = 0.095) or PEU (r = −0.022), suggesting limited direct impact on them. The SUS exhibited weak, non-significant positive correlations with both PU (r = 0.262) and PEU (r = 0.243), implying that while perceptions of usefulness and ease of use may contribute to usability perceptions, the relationships were not statistically robust in this dataset.

Table 5. Correlation analysis for the questionnaires (Spearman's rho).

Correlations	SUS	Design Structure of ATLog Platform	PU	PEU
SUS	1			
Design Structure of ATLog platform	0.450 *	1		
PU	0.262	0.095	1	
PEU	0.243	−0.022	0.879 **	1

*. Correlation is significant at the 0.05 level (2-tailed); **. Correlation is significant at the 0.01 level (2-tailed).

A strong and statistically significant correlation was found between PEU and PU (r = 0.879, $p < 0.01$), supporting the TAM hypothesis that ease of use strongly influences the perceived usefulness of the presented ATLog platform.

5. Discussion

By reducing technical barriers and aligning design with therapeutic needs, we aim to foster innovative practices among SLT practitioners and increase their engagement with high-tech ATs through a user-friendly visual programming interface and a seamless integration of ATs. ATLog utilizes RESTful APIs to ensure standardized and secure communication with ATs, making the system accessible without requiring advanced technical expertise. The platform is built to be flexible, allowing therapists easily customize it to fit each child's therapy needs. By decomposing complex SAR and VR functionalities into therapy-specific microskills, represented by pre-configured graphical blocks, therapists can intuitively design personalized therapy scenarios aligned with targeted goals and the language level of each child, without engaging in software coding.

The technical details provided for developers enable easy replication and extension of the ATLog platform design.

5.1. Technological Innovation of ATLog Platform

ATLog offers customizable resources tailored to specific therapeutic goals, treatment domains and language skill levels. The platform is designed with a subprocess-oriented approach, where SAR skills and VR scenarios are broken down into microskills represented as pre-programmed graphical blocks. The interoperability and scalability of ATLog's open-access framework and expandable design provide opportunities for integration with additional ATs, extending its functionality. The integrated mobile telepresence robots will facilitate remote speech therapy interventions.

5.2. Human–Computer Interaction (HCI)

Through ATLog's frontend, speech therapists can independently select and combine ATs based on their needs. The platform's visual programming interface allows therapists

to assemble predefined blocks for robotic behaviors, 3D scenes, interactive elements and cloud-based text generation in multiple languages. HCI using ATLog, with integrated NLP and NLG in SARs/VRs, enhances human-like natural language interactions and helps encourage intensive listening and speaking in children.

5.3. Social and Economic Impact

The proposed approach of integrating high-tech ATs in ATLog allows for the seamless integration of other ATs, appropriate to different language skill levels and communication needs. ATLog enhances accessibility by enabling therapists to design structured and personalized SLT scenarios with ATs. The design is intuitive and empowers those without advanced technical and programming expertise or a large budget. Through dynamic combinations of pre-programed blocks for specific AT devices and resources, ATLog facilitates the design of individualized therapy interventions tailored to each child's needs. Since ATLog is an open-access, universal software platform, it minimizes the need for costly custom solutions, making ATs more affordable. Additionally, it will support professionals working with children with CDs across different settings, such as resource centers, schools and healthcare institutions.

5.4. User Behavior Analysis

5.4.1. SUS Q1–10

The SUS helps assess usability by measuring how easy and efficient SLT practitioners find the ATLog platform. The results in Table 1 show that the SUS questionnaire has a Cronbach's Alpha of 0.798, indicating strong reliability. This score demonstrates that the questionnaire is highly consistent in measuring the usability of the ATLog platform. When comparing this value with other studies that apply the SUS, we find that reliability scores range from 0.83 to 0.97 [79]. Our score (0.798) is slightly lower but still within the strong range, which aligns well with typical values found in the literature.

An SUS average score of 70 is considered the threshold for acceptable usability [81]. In our study, the ATLog platform received an average SUS score of 78, indicating that the participants perceived it as having good usability. When mapped onto the adjective rating scale [81], this score corresponds to a "Good" usability level, reflecting the professionals' positive experience with the platform. When comparing this result to other studies, the ATLog platform outperforms the average SUS score for Internet platforms. According to research in [79], where the authors analyzed 77 surveys in the Internet platforms category, the mean SUS score was 66.25 (SD = 12.42). This score is below the 70 point usability acceptability threshold, indicating that many Internet platforms may present usability challenges or require improvements to meet user expectations. Furthermore, the score for the Design Structure evaluation of ATLog platform was 86, categorizing it as "Excellent". This suggests that while the overall platform usability is well-received, the specific actions and functionalities therapists engage with are perceived as even more intuitive and efficient.

Figure 8 reveals that while most participants rated ATLog positively, some variation in individual ratings was observed. Specifically, 74.07% of professionals rated the SUS at 70 or higher, with grades of C (18.5%), B (25.9%), or A (29.6%). This indicates that the majority of users found the platform's usability to be acceptable, good, or excellent. Eleven percent gave a grade of D, indicating an SUS score between 60 and 69, suggesting the need for improvement, while 15% of professionals rated usability in the range [52.5–57.5], corresponding to "F". Most professionals rated the ATLog Design Structure evaluation positively, with 89% giving an SUS grade of A, B, or C, indicating high intuitiveness and efficiency. Only 11% gave a D or F score, suggesting some usability challenges to be explored.

To analyze the average scores for each statement in the two parts of the questionnaire, the SUS and Design Structure evaluation of the ATLog platform, we followed the approach used in [103]: statements with scores above 3.00 were classified as positive (agree), while those below 3.00 were considered negative (disagree). Scores very close to 3.00 were challenging to classify as distinctly positive or negative, as they lie near the neutral point of the Likert scale. An example of a borderline assessment is Q10 in Table 2. A score of 2.8 was the average for Q10. This indicates a slight tendency toward disagreement but not a definitive or strong view. This means that SLT practitioners do not strongly agree with the notion that they need to learn a lot before using the ATLog Platform, but they may still feel that some learning is required. This result suggests that professionals believe they do not need to learn a great amount of information before using the ATLog platform. While the score is closer to the neutral point, it still indicates that therapists feel the system is somewhat accessible or easy to understand without requiring extensive learning. However, the score of 2.8 implies that some may feel the need for some basic knowledge before using the ATLog platform effectively. If we compare the outliner Q10 average score with the one for Q7 (4.4), which is the opposite statement (positive), there is a big difference. It shows that the participants believe that the ATLog platform is easy for most people to learn quickly. The difference between the Q7 value and 3 is greater than the difference between the Q10 value and the neutral 3, where 1.4 (Q7) > 0.2 (Q10), with the difference between the two values being 1.2. The difference between the two values highlights a clear contrast in the therapists' perceptions. The score for Q7 shows strong agreement that the ATLog Platform is easy to learn. This result supports the conclusion that the ATLog has a user-friendly design, requiring minimal introduction of information for new users. However, the presence of some slight hesitation in Q10 suggests that offering additional resources, such as videos or training materials, could further enhance user confidence and ensure a smoother learning process for all professionals.

5.4.2. Q11–20

The second part of the questionnaire provided a more detailed analysis of user behavior, focusing on how therapists interact with the ATLog platform and their perception of its functionality and user-friendliness. In comparison, the reliability test results are even stronger, with a Cronbach's Alpha value of 0.849. This is an indication of excellent consistency and reliability. This score places the reliability of the ATLog survey in the very high range. The strong reliability scores in both sections of the questionnaire used in the current study (SUS and Design Structure evaluation in the survey) are well suited for measuring usability and the experience of SLT practitioners. However, it is important to note that even higher reliability could be achieved. It would be valuable to study a larger and more diverse group of professionals working with children and adolescents with CD and using the platform to assist technology-based interventions. These advancements could help enhance the results.

Research by Bangor et al. established a correlation between the participants' age and their scores on the SUS survey [81]. Contrary to the established relationship, our study did not observe such an association. These findings are supported by the results obtained in a systematic review of 30 scientific studies evaluating the usability of educational technology, where [103] found that there was no statistically significant correlation between the age of the respondents and the SUS scores. Future studies could explore this relationship further by including a larger number of SLT practitioners.

Regarding the open-ended question, the feedback from the experts highlights the ATLog platform's' potential and usefulness. A total of 41% of participants did not provide any comments in this section. Four participants described the platform as interesting

and beneficial, especially for professionals and children with CDs. Additionally, 22% of the responses emphasized the need for broader promotion and awareness among more professionals. Interest in training sessions was expressed by 11% of participants. One participant raised concerns about delays in robot responses affecting the session flow. Another one suggested simplifying scenario creation to involve only one robot. This can be achieved in the ATLog frontend when only one SAR category is utilized.

5.4.3. TAM Q1–12

The results from the TAM assessment demonstrate that speech and language therapists perceive the ATLog platform as both highly useful and easy to use (Table 4). This aligns with the TAM's core premise that perceived usefulness and ease of use are critical for successful technology adoption in professional contexts. A particularly strong result was seen for the item "*It would be easy for me to learn to work with the ATLog platform*", which received the highest mean score (M = 6.5), reflecting high user confidence in learning and using the system. This finding suggests that ATLog is intuitive, user-friendly, and easy to use and learn, allowing therapists to start using it quickly with minimal training.

While overall acceptance was high, some nuances emerged. The statement "*Using the ATLog platform in my work would allow me to complete my therapeutic tasks faster*" received the lowest mean score (M = 5.9; SD = 1.0), which, although still positive, suggests some hesitation about the platform's impact on time efficiency. This may reflect limited real-world experience with the platform or uncertainty about integrating its diverse features—such as combining ATs or uploading custom data—into everyday practice. The slightly higher standard deviation also points to varied individual perceptions regarding time savings, underscoring the need for hands-on training or the demonstration of practical efficiencies.

TAM subscales revealed particularly high reliability, with Cronbach's Alpha values of 0.906 for PU and 0.922 for PEU. These results not only exceed the conventional threshold of 0.70 for acceptable reliability [104] but fall within the range considered excellent, indicating that the TAM constructs were measured with a high degree of internal coherence. These results confirm that the measurement tools used in this study were both statistically reliable and psychometrically valid, enabling confident interpretation of the observed relationships between design structure, usability, and user acceptance. The high reliability of the measures strengthens the overall validity of the study's conclusions and confirms the appropriateness of using these standardized tools in the evaluation of user experience within digital platforms like ATLog.

The distribution of responses further reveals interesting patterns, as shown in Figures 10 and 11. For PU, 81% of participants gave higher ratings (Grades B or A), with 44% awarding the top rating (A). This suggests a strong overall validation of the ATLog platform's ability to support therapeutic tasks, despite some variability. For PEU, responses were even more favorable and consistent—63% of SLT practitioners rated the platform in the top range (90–100, Grade A), and only 11% offered lower ratings (Grade D), indicating few concerns about usability.

In summary, the evaluation of ATLog via the TAM provides robust evidence of its high perceived usefulness and ease of use among speech and language therapists. While the platform is generally seen as supportive and accessible, future implementations should emphasize hands-on training and efficiency in practice to address any remaining uncertainties and promote sustained adoption in therapeutic settings.

5.4.4. Correlation Analysis Between the SUS, the Design Structure Evaluation of the ATLog Platform, and the TAM

The Design Structure evaluation of the ATLog platform demonstrated a moderate and statistically significant correlation with the SUS, as shown in Table 5. This finding suggests that a well-organized and intuitively structured interface plays a critical role in enhancing users' overall usability perceptions. However, the Design Structure evaluation did not significantly correlate with PU or PEU, indicating that while structural clarity improves usability perceptions, it may not directly influence users' judgments of a system's utility or simplicity.

Interestingly, both PU and PEU showed weak and non-significant correlations with SUS, suggesting that while these subscales are central to the TAM, they may not directly contribute to how usable a system is perceived to be when measured by the SUS. This might reflect how the SUS primarily captures users' overall satisfaction with the ATLog platform's functionality and interface, whereas PU and PEU are more related to anticipated performance outcomes and cognitive effort, respectively.

The most robust finding was the very strong and highly significant correlation between PEU and PU ($r = 0.879$, $p < 0.01$). This is consistent with the original TAM framework [25], which suggests that systems perceived as easy to use are more likely to be seen as useful. This reinforces the idea that ease of interaction is a major driver of perceived effectiveness, especially in digital platforms like ATLog.

Together, these findings suggest a partial validation of the TAM in the context of usability and system design. While ease of use remains a critical determinant of perceived usefulness, system usability (as measured by the SUS) is more directly affected by the quality of the ATLog platform's Design Structure evaluation than by perceived usefulness or ease of use alone.

6. Conclusions

Following the four-phase Design-Based Research process proposed by Reeves, we identified and addressed the practical challenges faced by speech and language therapists in accepting and adopting high-tech ATs, such as Socially Assistive Robots, VR and conversational AI. Through iterative development, in collaboration with and based on feedback from therapists, we designed and refined ATLog—a visual programming platform that enables the integration of ATs and the intuitive creation of personalized, interactive and playful learning scenarios. As a result, the *Design-Based Principle* identified is as follows: *"Streamlining the integration of high-tech ATs (SARs, VR, ConvAI) in SLT can be achieved through an ergonomic software platform, such as ATLog. It lowers technical barriers through visual programming tools, such as a Google Blockly-based frontend, and a lightweight Express server that acts as the backend hub, managing all communications between the frontend and the ATs. The platform enables scenario personalization and supports the modular design of therapy activities through predefined graphical blocks aligned with specific therapeutic goals and skill levels"*.

To foster innovative thinking in therapists, the resulting design principle highlights the need to lower technical barriers through therapy-centered modularity and customizable tools. By breaking down complex AT functionalities into modular microskills aligned with specific therapy goals and language levels, ATLog empowers therapists to streamline the integration of ATs within SLT practice and to design and run interactive, playful scenarios without requiring advanced technical skills.

The identified principle is validated by the results of the TAM and SUS surveys, which suggest that the platform is well-designed, intuitive and effective. Most of the SLT practitioners (74%) evaluated the SUS of the ATLog platform with scores over 70, indicating high acceptance of its usability and functionality, assigning grades C (18.5%), B (25.9%) and A (29.6%). Over half (52%) of the experts rated the additional questions focused on ATLog's structure and functionalities in the A range (90–100), while 26% rated them in the B range (80–89), showing strong acceptance of the platform for creating and running personalized interactive scenarios with ATs. According to the TAM results, experts gave high grades for both perceived usefulness (44% in the A range) and perceived ease of use (63% in the A range). However, based on the open questions, there are still areas for improvement, particularly in enhancing integration and compatibility with other ATs, refining the user interface, and expanding educational resources with more step-by-step guides and interactive tutorials for beginners. The comparative strength of the ease-of-use ratings over perceived usefulness may indicate that while the platform is user-friendly, its practical value in clinical contexts might still be under evaluation or may be context-dependent.

6.1. Limitations

Despite promising outcomes, this study has several limitations that may affect the generalizability of the findings. The iterative evaluations were conducted within a limited range of SLT contexts and therapy types, and broader validation is needed to assess the platform's long-term impact and adoption. While ATLog simplifies technical complexity, its effectiveness in improving therapy outcomes requires further empirical investigation through large-scale, longitudinal studies involving children with CDs. Additionally, the questionnaires used in the study were performed only by stakeholders who participate in the training session, which limited the diversity of perspectives captured. Finally, the research was conducted within a national context, primarily focusing on Bulgaria, which may restrict the transferability of the results to other regions due to cultural, economic and systemic differences in how ATs are adopted globally.

6.2. Future Directions

While the current study focused on therapists' perspectives on designing structured playful scenarios in the physical, digital and virtual world using the ATLog platform, future research will evaluate the platform's impact on therapy outcomes by assessing how ATLog supports skill development in children and adolescents with CDs. A six-month experimental study involving 20 children and adolescents is currently ongoing in real-world therapy settings. By the end of the year, we aim to evaluate the intervention's effectiveness in enhancing skill acquisition through increased engagement and motivation by ATs, as well as to generate new insights for refining the design of the ATLog platform. Future development will also include the integration of other ATs, such as Oculus VR glasses, further refinement of the Emo robot for emotional learning, automatic phonological and linguistic assessment of disordered speech, and ATLog telepractice capabilities.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/technologies13070306/s1, Video S1: ATLog_lesson1, Video S2: ATLog_lesson2, Video S3: ATLog_lesson3, Video S4: ATLog_lesson4.

Author Contributions: Conceptualization, A.L., G.D., T.T. and A.A.; methodology, A.L., P.T., A.A., M.S., V.S.-P. and G.P.; software, A.L., A.K., K.R., G.D., T.T., D.V., T.S. and P.T.; validation, A.L., G.D., T.S., P.T., K.R., A.K. and D.V.; formal analysis, M.S., A.A., V.S.-P. and G.P.; data curation, A.A. and P.T.; writing—original draft preparation, A.L., P.T., A.A. and T.S.; writing—review and editing, V.S.-P., G.D., D.V., T.T., A.A. and K.R.; visualization, A.L., T.T., T.S., A.A. and P.T.; project administration, A.L., G.D. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research findings are supported by the Bulgarian National Scientific Research Fund, Project No KP-06-H67/1.

Institutional Review Board Statement: The ethical procedures inherent to these surveys were approved by the ethical committee of the Institute of Robotics Bulgarian Academy of Sciences (ECSR; Decision No. 7/21.07.2023, par. 4 of the Scientific Council of IR-BAS). This procedure was formalized in the protocols No. 7/05.02.2025 and No. 11/06.06.2025, issued by the Ethics Committee for Scientific Research, regarding a request submitted by Anna Lekova.

Informed Consent Statement: Written informed consent was obtained from all subjects involved in this survey.

Data Availability Statement: We provide data supporting the reported results at https://osf.io/3z65m/ (last access on 11 July 2025).

Acknowledgments: We acknowledge the technical support by the European Regional Development Fund under the Operational Program "Scientific Research, Innovation and Digitization for Smart Transformation 2021–2027", Project CoC "Smart Mechatronics, Eco- and Energy Saving Systems and Technologies", BG16RFPR002-1.014-0005.

Conflicts of Interest: The authors declare no conflicts of interest related to this study.

Abbreviations

The following abbreviations are used in this article:

AAC	Alternative and Augmented Communication
ATs	Assistive Technologies
CDs	Communication Disorders
ConvAI	Conversational AI
DBR	Design-Based Research
GPT	Generative Pre-trained Transformer
IDE	Integrated Development Environment
LLMs	Large Language Models
LRMs	Large Reasoning Models
NLG	Natural Language Generation
NLP	Natural Language Processing
PEU	Perceived Ease of Use
PU	Perceived Usefulness
PY	Python
Q&A	AI questions and answers
RQ	Research Question
SARs	Socially Assistive Robots
SCP	Secure Copy Protocol
SLT	Speech and Language Therapy
SUS	System Usability Scale
TAM	Technology Acceptance Model
VR	Virtual Reality

Appendix A

Table A1. Nao robot frontend blocks with corresponding backend endpoints.

Frontend Block Name	Description	Microskill/File Uploaded on the Robot	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
StartBlock	Initializes the session. ***Connected endpoints in doCode of the block***: *StopAll_BehaviorsBlock* and *AutonomousModeBlock*.	No	N/A	N/A	N/A
Autonomous ModeBlock	Enables or disables autonomous behavior on Nao robot.	No	ipNao, STATE (FieldCheckbox: TRUE, FALSE)	/nao1	http://serverIP:3000/nao1?IP=${ipNao}&state=${STATE}&pyScript=auton
StopAll_BehaviorsBlock	Stops ongoing behaviors on Nao robot.	No	ipNao	/nao1	http://serverIP:3000/nao1?IP=${ipNao}&pyScript=stopBehav2
VolumeBlock	By entering an integer from 0 to 100, the Nao robot's volume level can be adjusted.	No	ipNao, VOLUME	/nao1	http://serverIP:3000/nao1?IP=${ipNao}&volume=${VOLUME}&pyScript=volume
HelloBlock	Nao animation—greeting while in a seated position.	Yes	ipNao	/nao2	http://ipServer:3000/${endpoint}?IP=${ipNao}&BN=User/hello-6f0483/behavior_1
Read_qr_code Block	Initiates QR code reading in front of Nao's eyes. Reads from Nao's memory. Saves in ATLog repository with a key = QR. ***Connected endpoints in doCode of the block:*** *NaoMemoryRead* and *RepositorySaveBlock*.	Yes	ipNao KEY	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=User/qr-19f134/behavior_1 ***Connected endpoints:*** *http://serverIP:3000/monitor/pyNao2*
LingSounds Block	Nao says the Ling sound, defined by the input parameter, and waits to recognize the QR code shown to it.	Yes	ipNao, LING_SOUND: A, I, U, M, S, Sh	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=ling-2b453a/${LING_SOUND}
TransportationBlock	Nao says the name and play sounds for a transportation tool and waits to recognize the QR code shown to it.	Yes	ipNao, TRANSPORT: Bicycle, Car, Train, Motor, Airplane, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=transports-c868b7/${TRANSPORT}
AnimalsBlock	Nao says the animal's name, plays its sound, and waits to recognize the QR code shown to it.	Yes	ipNao, ANIMAL: Cat, Dog, Cow, Sheep, Frog, Horse, Monkey, Pate, Pig, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=animals-486c82/${ANIMAL}
FruitsBlock	Nao says the fruit's name and waits to recognize the QR code shown to it.	Yes	ipNao, FRUIT: strawberry, bannana, apple, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN= fruits-f0887f/${FRUIT}

Table A1. *Cont.*

Frontend Block Name	Description	Microskill/File Uploaded on the Robot	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
Vegetable Block	Nao says the vegetables's name and waits to recognize the QR code shown to it.	Yes	ipNao, VEGETABLE: cucumber, cabbage,	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=vegetables-7e6223/${VEGETABLE}
ShapesBlock	Nao says the shape's name and waits to recognize the QR code shown to it.	Yes	ipNao, SHAPE: Triangle, Rectangle, Circle, Square, Star, Moon, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=shapes-cc46c2/${SHAPE}
ColorsBlock	Nao says the color's name and waits to recognize the QR code shown to it.	Yes	ipNao, COLOR: Green, Orange, Red, Blue, Pink, White, Yellow, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=colors-dd3988/${COLOR}
MusicalInstr Block	Nao plays music with animation for an instrument (defined by the input parameter).	Yes	ipNao, INSTRUMENT: Trumpet, Guitar, Maracas, Piano, Drums, Violin, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=instruments-78257e/${INSTRUMENT}
EmotionsBlock	Nao performs an animation for the emotion defined by the input parameter.	Yes	ipNao, EMOTION: sad, happy, scared, tired, neutral	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=emotions-f6ef56/${EMOTION}
FairytaleBlock	Nao narrates a fairy tale, defined by the input parameter, and its mp3 is uploaded on Nao folder: home/nao/fairs/. ***Connected in sequence blocks:*** NarratingBlock.	Yes	ipNao, FAIRY_TALE_NAME: MuriBird, CatMum, 3cats, smallCat, 3catsMum, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=story-afb24d/${ FAIRY_TALE_NAME }
StoryBlock	Nao tells a story with sequential in time scenes by three or more flashcards (the story is defined by input parameter).	Yes	ipNao, STORY_NAME: tomato, cake, sleep, cat, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=[story_nina-97b2d5,story_emi-e0be5f,story_cat-7f9f13, story_mimisleep-afcea3]
SongBlock	Nao sings a song (defined by the input parameter), and its mp3 is uploaded on Nao folder: home/nao/songs/. ***Connected in sequence blocks:*** DanceBlock.	Yes	ipNao, SONG_NAME: song1, song2, song3, song4, song5, etc.	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&filepath=${filepath}&filename=${SONG_NAME}&pyScript=play_mp3
MarketPlay Block	Nao simulates market-related interactions with the child using 3D objects embedded with barcodes (real or toys), such as toothbrush, cash desk, payment card, etc.	Yes	ipNao, MarketItem: Juice, Salad, Bathroom, Teeth, etc.	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=market-6f6441/${MarketItem}

Table A1. *Cont.*

Frontend Block Name	Description	Microskill/File Uploaded on the Robot	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
TouchSensor Block	Nao detects a touch on its tactile sensors and plays the corresponding mp3 for the touched body_part, uploaded on the robot in folder home/nao/body/.	Yes	ipNao, SENSOR: Head_Front, RArm, Larm, LFoot/Bumper RFoot/Bumper	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&touch=${SENSOR}&pyScript =touchBodyPart
Head_tactile_ sensorBlock	Activates the tactile sensor on Nao head to stop any currently running microskill. ***Connected in sequence block:*** *StopAll_behaviours.*	No	ipNao, SENSOR Head_Front, Head_Middle Head_Rear	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&bodyPart&{SENSOR}&pyScript =touchHead
NaoMemory ReadBlock	Reads a value from Nao's memory key.	No	ipNao, KEY	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&key=${KEY}&pyScript=ALmemory_get
NaoMemory SaveBlock	Writes a {key: value} to Nao's memory.	No	ipNao, KEY, VALUE	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&key=${KEY}&value=${VALUE}& pyScript=ALmemory_set
DeclareEvent Block	Declares an event name in Nao's memory.	No	ipNao, EVENT	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&name=${ EVENT }&pyScript=ALmemory_decl
RaiseEvent Block	Raises a declared event in Nao's memory.	No	ipNao, EVENT	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&name=${ EVENT }&pyScript=ALmemory_raise
PlayAudio Block	Plays audio files uploaded to the robot in Nao folder home/nao/audio/.	Yes	ipNao, path, file_name	/nao1	http://serverIP:3000/nao1?IP=${robot_ip}&filepath=${path}&filename=${file_name}& pyScript=play_mp3
UploadFile Block	Uploads a file on Nao robot in folder:/home/nao/tts/with name: tts.mp3.	No	ipNao, passwordNao, file_path, file_name	/naoUpload	http://serverIP:3000/naoUpload?IP=${ipNao}&password=${passwordNao}${file_path}$ {file_name}
Text_to_speechBlock	Nao plays the input text as audio file. Text is converted to speech by GoogleTTS or MSAzureTTS. The generated tts.mp3 file is uploaded to NAO folder:/home/nao/tts/.	No	ipNao, passwordNao text filename: tts.mp3, filepath on Nao:/home/nao/tts/	/nao3	http://serverIP:3000/${endpoint}?IP=${ipNao}&filepath=${path}&filename=${file_name} ?TEXT=${text} ***Connected endpoints:*** *http://serverIP:3000/venv;http://serverIP:3000/monitor/pyNao1* http://serverIP:3000/naoUpload?*IP=${ipNao}*&password=${passwordNao}${file_path}$ {file_name} *http://serverIP:3000/nao1?IP=${robot_ip}&filepath=${path}&filename=${file_name}&pyScript= play_mp3* http://serverIP:3000/${endpoint}?IP=${ipNao}&filepath=${path}&filename=${file_name} ?TEXT=${answer)}
PlaytextBlock	Nao plays the current answer in the ATLog repository as audio file. Answer is converted to tts.mp3 and uploaded to Nao directory/home/nao/tts/.	No	ipNao, passwordNao file_name, file_path, KEY=answer	/nao3	***Connected endpoints:*** http://serverIP:3000/repository/answer *http://serverIP:3000/venv;http://serverIP:3000/monitor/pyNao1* http://serverIP:3000/naoUpload?*IP=${ipNao}*&password=${passwordNao}${file_path}$ {file_name} *http://serverIP:3000/nao1?IP=${robot_ip}&filepath=${path}&filename=${file_name}&pyScript= play_mp3*

Table A1. *Cont.*

Frontend Block Name	Description	Microskill/File Uploaded on the Robot	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
QRquestion Block	Nao waits for a QR code to be shown, decodes it to generate a text-based question, sends the question to bgGPT, and stores the received answer in the repository. It then plays the answer as an audio file. ***Connected endpoints in doCode of the block: PlaytextBlock.***	No	ipNao, Question, Context, KEY =answer, VALUE	/nao1 /nao2 /nao3 /bgGPT /repository */monitor/pyNao1* */monitor/pyNao2*	***Connected endpoints:*** *http://serverIP:3000/nao2*?IP=${ipNao}&*BN=qr-19f134* *http://serverIP:3000/monitor/pyNao2* *http://serverIP:3000/nao1?IP=*${ipNao}*&key=QR&pyScript=ALmemory_get2* *http://serverIP:3000/venv; ttp://serverIP:3000/monitor/pyNao1* http://serverIP:3000/bgGPT?question=${question}&context=${context} *http://serverIP:3000/repository?key=${KEY}&value=${VALUE*
TextQuestion Block	Retrieves the questions and context entered in the input fields, sends them to bgGPT, stores the generated answer in the repository, and Nao plays the current response. ***Connected endpoints in doCode of the block:*** **PlaytextBlock.**	No	ipNao, Question, Context, KEY: answer, VALUE	/nao1 /nao3 /bgGPT /repository	***Connected endpoints:*** *http://serverIP:3000/venv* http://serverIP:3000/bgGPT?question=${question}&context=${context} *http://serverIP:3000/monitor/pyNao1* *http://serverIP:3000/repository?key=${KEY}&value=${VALUE*
Narrating Block	Nao animation—for narrating pose. **Connected in sequence blocks**: *NaoMemorySave* with {key= narr_sec, value= VALUE}.	Yes	ipNao VALUE (Seconds for narrating)	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=User/slow_narrating-0de14c/behavior_1
Thinking Block	Nao animation—thinking pose in a seated position.	Yes	ipNao	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao&BN=User/thinking-b54157/behavior_1
Narrating_sit Block	Nao animation—pose for storytelling in a seated position.	Yes	ipNao	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=User/narr_sit-a1f157/behavior_1
Dance Block	Nao animation—performing dance movements.	Yes	ipNao	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&BN=User/dancing-e1h145/behavior_1
CustomBehavior Block	Runs a custom skill prepared in Choregraphe and uploaded to Nao.	Yes	ipNao, BEHAVIOR_NAME (Unique Behavior ID on Nao)	/nao2	http://serverIP:3000/${endpoint}?IP=${ipNao}&?id=${BEHAVIOR_NAME}
CustomRemote Block	Runs a custom python script by remote NAOqi.	No	ipNao, PY_SCRIPT (Unique py script name)	/nao1	http://serverIP:3000/${endpoint}?IP=${ipNao}&${endpoint}id=${PY_SCRIPT}

Table A2. Furhat robot frontend blocks with corresponding backend endpoints.

Frontend Block Name	Description	Microskill/File Uploaded on the Robot	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
Monitoring SafetyBlock	Furhat listens for keywords like 'STOP', and stops the actions of the integrated ATs in the current session.	No	ipFurhat	/furhatRem	http://serverIP:3000/furhatRem?IP=${ipFurhat}&scriptName=fur_stop.py ***Connected endpoints:*** *http://serverIP:3000/venv* *http://serverIP:3000/furhat?IP=${ipFurhat}&?id=RemoteAPI* *http://serverIP:3000/stop-server*
CustomSkill Block	Runs a custom skill prepared in KotlinSSDK and uploaded to Furhat.	Yes	ipFurhat, SKILL_NAME	/furhat	http://serverIP:3000/${endpoint}?IP=${ipFurhat}&?id=${SKILL_NAME}
FurhatTTS Block	Specifies the text to vocalize (TTS).	No	ipFurhat, Text to Vocalize	/furhatRem	http://serverIP:3000/${endpoint}?IP=${ipFurhat}&scriptName=fur_tts.py&text=${text} ***Connected endpoints:*** *http://serverIP:3000/venv* *http://serverIP:3000/*${endpoint}*?IP=${ipFurhat}&?id=RemoteAPI*
FurhatAsk NameBlock	Asks the user their name and saves the response in the internal repository.	No	ipFurhat, KEY: name	/furhatRem	http://serverIP:3000/furhatRem?IP=${ipFurhat}&scriptName=fur_name.py **Connected endpoints:** http://serverIP:3000/venv; http://serverIP:3000/monitor/pyFurhat http://serverIP:3000/${endpoint}?IP=${ipFurhat}&?id=RemoteAPI http://serverIP:3000/repository?key=${KEY} http://serverIP:3000/${endpoint}?IP=${ipFurhat}&scriptName=furhat_rem_textFur.py'
FurhatSpeak AnswerBlock	Furhat speaks the answer stored in the internal repository.	No	ipFurhat, KEY: answer	/furhatRem	**Connected endpoints:** http://serverIP:3000/venv http://serverIP:3000/${endpoint}?IP=${ipFurhat}&?id=RemoteAPI http://serverIP:3000/repository?key=${KEY} http://serverIP:3000/${endpoint}?IP=${ipFurhat}&scriptName=furhat_rem_question.py'
FurhatQuestionBlock	Listens to a question, **fetches the question to bgGPT** and stores the answer in the repository. ***Connected endpoints in doCode of the block: FurhatSpeakAnswerBlock.***	No	ipFurhat, Question, Context, KEY: answer, VALUE	/furhatRem	***Connected endpoints:*** *http://serverIP:3000/venv;* http://serverIP:3000/monitor/pyFurhat *http://serverIP:3000/*${endpoint}*?IP=${ipFurhat}&?id=RemoteAPI* http://serverIP:3000/bgGPT?question=${question}&context=${context} *http://serverIP:3000/repository?key=${KEY}&value=${VALUE}*
FurhatRepeat Block	Listens to a spoken phrase with attempts to repeat it, unless it returns a 'NOMATCH' response.	No	ipFurhat, text_ intent	/furhatRem	http://serverIP:3000/furhatRem?IP=${ipFurhat}&scriptName=fur_understand.py ***Connected endpoints:*** *http://serverIP:3000/venv* *http://serverIP:3000/*${endpoint}*?IP=${ipFurhat}&scriptName=fur_tts.py&text=${text}*

Table A3. Frontend blocks for robots EMO and Double3 with corresponding backend endpoints.

Frontend Block Name	Description	Microskill/File Uploaded on the Robot	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
Start_EMO_mqqt_ session1	Establishes MQTT session1 (via Python paho-mqtt bridge).	remote_paho_ mqtt.py	ipEmo, Port, Username, Password (PUP)	/emo1	http://serverIP:3000//${endpoint}?IP=${ipEmo}&PUP=${PUP} ***Connected endpoints:*** *http://serverIP:3000/venv*
Start_EMO_mqtt_ session2	Establishes an MQTT session2 (via MQTT net library).	Yes	ipEmo, Port, Username, (PU), TOPIC, file_name	/emo2	http://serverIP:3000//${endpoint}?IP=${ipEmo}&PU=${PU}&topic=${TOPIC}&file=${file_name}
ReadQRcode	Initiates QR code reading in front of EMO. Saves in ATLog repository with a key = QR. ***Connected in sequence blocks:*** *RepositorySaveBlock.*	Yes	ipEmo, Port, Username, Password (PUP), KEY	/emo1	http://serverIP:3000/${endpoint}?IP=${ipEmo}&PUP=${PUP}&script=*qr_reader_atlog.py* ***Connected endpoints:*** *http://serverIP:3000/venv*
UploadFile Block	Uploads a file after serialization to Emo robot with topic = 'upload'.		ipEmo, Port, Username, (PU), TOPIC, data	/emoUpload	http://serverIP:3000/${endpoint}?*IP=${ipEmo}*&PU=${PU}&topic=${TOPIC}& serialization=${data}***Connected endpoints:*** *http://serverIP:3000/emo2*
EmoPlay AudioBlock	Emo plays audio file uploaded on Emo folder/home/emo/audio/ topic = 'audios'.	Yes	ipEmo, Port, Username, (PU), TOPIC, filename filename	/emoPlay	http://serverIP:3000/${endpoint}?IP=${ipEmo}&PU=${PU}&topic=${TOPIC}&filename=${name} ***Connected endpoints:*** *http://serverIP:3000/emo2*
EmoChat TextBlock	Emo plays the current answer from BgGPT in the internal repository as audio file. Answer is converted in tts.mp3 and uploaded to Emo folder/home/emo/tts/ topic is 'answer'.	Yes	ipEmo, Port, Username, (PU), TOPIC, filename filename/home/emo/tts/, KEY	/emoChat	http://serverIP:3000/${endpoint}?IP=${ipEmo}&PU=${PU}&topic=${TOPIC}&filename=${file_name}?TEXT=${answer)} ***Connected endpoints:*** http://serverIP:3000/repository/answer *http://serverIP:3000/venv* *http://serverIP:3000/emoUpload; http://serverIP:3000/emoPlay*
EmoQRquestionBlock	Waiting for a QR code, decode it to prepare a question, **fetch the question to bgGPT** and stores the answer in the repository. Emo plays the current answer. ***Connected endpoints in doCode of the block: EmoPlayAudioBlock***	No	ipEmo, Question, Context, KEY: answer, VALUE	/qrEmo	http://serverIP:3000/${endpoint}?IP=${ipEmo}&PUP=${PUP} ***Connected endpoints:*** *http://serverIP:3000*/emo1 *http://serverIP:3000/venv* *http://serverIP:3000/emo1*?IP=${ipEmo}*&pyScript=qr_reader_atlog.py* http://serverIP:3000/bgGPT?question=${question}&context=${context} *http://serverIP:3000/repository?key=${KEY}&value=${VALUE}* *http://serverIP:3000/emo2/emoUpload* *http://serverIP:3000/emo2/emoPlay*
Start_D3_session	Opens an SSH session to a Double3 (D3) robot and runs PY scripts that interact with the Double3 Developer SDK and D3 API.	Yes	ipD3, Port, Username, Password (PUP), script_name	/double3	http://serverIP:3000/double3?IP=${ipD3}&PUP=${PUP}&script=alive.py
D3QRBlock	Opens an SSH session to D3 and runs PY script that reads QR codes via D3 robot.	Yes	ipD3, Port, Username, Password (PUP), script_name	/double3	http://serverIP:3000/double3?IP=${ipD3}&PUP=${PUP}&script=readQR.py
D3 ControlBlock	Opens an SSH session to a D3 and run PY script with parameters.	Yes	ipD3, Port, Username, Password (PUP), script_name, PARAM	/double3	http://serverIP:3000/double3?IP=${ipD3}&PUP=${PUP}&script=${script_name} &param=${PARAM}

Table A4. General frontend blocks with corresponding backend endpoints.

Frontend Block Name	Description	Microskill/File Uploaded on the Robot	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
Stop Code (button)	Sends a request to shut down the server by terminating the all active child processes and closes the database connections	No	No	/stop-server	http://serverIP:3000/stop-server
Repository ReadBlock	Reads a value in ATLog's repository with a specified key	N/A	KEY	/repository	http://serverIP:3000//${endpoint}?&key=${ KEY}
Repository SaveBlock	Saves a value in ATLog's repository with the specified key	N/A	KEY, VALUE	/repository2	http://serverIP:3000//${endpoint}?&key=${ KEY}&value=${VALUE}
DB_save Block	Saves session observations or assessments in ATLog's database	No	postData: ID, VALUE (childID, observations, date and time recorded automatically)	/save-child-notes	http://serverIP:3000/${endpoint} with JSON: { method: 'POST', headers: { 'Content-Type': 'application/json; charset=utf-8' }, body: JSON.stringify(${JSON.stringify(postData)}) }
BG_GPT Block	Sends questions and context entered in the input fields to the BgGPT APIs (Bulgarian ChatGPT)	No	QUESTION, CONTEXT	/bgGPT	http://serverIP:3000/bgGPT?question=${QUESTION}&context=${CONTEXT}
QABlock	Sends questions and context to NLPcloud	No	QUESTION, CONTEXT	/QA	http://serverIP:3000/QA?question=${QUESTION}&context=${CONTEXT}
WaitBlock	Makes the robot wait for a specified number of seconds	No	SECONDS	N/A	**Blockly local run** await new Promise(resolve => setTimeout(resolve, ${ SECONDS }));
RepeatBlock	Wraps (one or more) blocks in a loop to execute it multiple times based on a variable number of iterations. *In doCode of the block*: any block(s)	No	ITERATIONS		**Blockly local run** loop to repeat execution ${ITERATIONS} times ***Connected endpoints*** = some_of ([*http://serverIP:3000/monitor/pyNao1,* *http://serverIP:3000/monitor/pyNao2,* *http://serverIP:3000/monitor/pyFurhat]}*

Table A5. VR frontend blocks with corresponding backend endpoints.

Frontend Block Name	Description	Microskill/File Uploaded on the Server	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
Setup_VR_Session_Block	Sets up Express server, WebSocket communication, assigns unique session IDs for the therapists and the child.	Socket.io, JavaScript ES6 Modules	VRsceneID	/startVR	http://serverIP:3005/${VRsceneID}/startVR
Setup_VR_Scene_Block	Initializes a VR scene in Three.js with camera, VR controllers, renderer, WebXR support and 3D model loading. Delivers static files (frontend code, images, styles) from the "public" folder. Dynamically generates HTML pages.	JavaScript ES6 Modules	sessionID, BINGO SIZE: 3x3,4x4,5x5, SCENARIO: animals, fruits, vegetables, transports.	/sessions:id/setup	http://serverIP:3005/sessions/${sessionID}/setup with JSON:{method: 'POST', headers: { 'Content-Type': 'application/json; }, body: JSON.stringify (${JSON.stringify({ "bingoSize": BINGO_SIZE, "scenario": SCENARIO })}) }
StartVRbutton_Block	Starts VR button in the immersive mode of the browser.	Socket.io, Three.js, WebXR, JavaScript ES6 Modules	N/A	/startVRbutton	http://serverIP:3005/startVRbutton

Table A5. *Cont.*

Frontend Block Name	Description	Microskill/File Uploaded on the Server	Input Parameters	Backend Endpoint	Fetching Endpoint(s) with Parameters
360_Virtual_Tour	An interactive 360° virtual tour on a touchscreen where children explore common places, as well as educational spaces. The tours are hosted on the ATLog project website.	N/A	SPACE: house, hypermarket, museum	N/A	Blockly local runtime for fetch request https://atlog.ir.bas.bg/images/tours/${SPACE}/index.htm
360_Virtual_Game	An interactive 360° virtual game on a touchscreen, where children search for and identify objects from a preassigned list. The games are hosted on the ATLog project website.	N/A	SCENARIO: animals, fruits, vegetables, transports	N/A	Blockly local runtime for fetch request https://atlog.ir.bas.bg/images/tours/${SCENARIO}/index.htm

Appendix B

Table A6. Express server setup.

Dependencies that Need to be Installed by Npm	Dynamic Imports Used for:
const express = require('express');	Express framework to create a web server.
const app = express();	Creating an instance of the Express application.
const {spawn } = require('child_process');	Running child processes (e.g., executing PY scripts).
const {Client } = require('ssh2');	SSH2 client module to establish SSH connections.
const {exec } = require('child_process');	Exec to run shell commands asynchronously.
const fetch = require('node-fetch');	Node-fetch for making HTTP requests.
const repository = {};	Initializing an empty object to store key-value pairs.
const Database = require('better-sqlite3');	SQLite database interactions (better-sqlite3).
const fs = require('fs');	File System (fs) module to handle file operations.
const path = require('path');	Path module to manipulate file paths.
const http = require('http');	HTTP module to create an HTTP server for VR session.
const socketIo = require('socket.io');	Socket.IO library for real-time communication in VR.
const server = http.createServer(app);	Create an HTTP server using the Express app.
const io = socketIo(server);	Initialize Socket.IO by passing the HTTP server.
const os = require('os');	OS module to interact with the operating system.
Middleware (built into Express)	**Establishing a middleware to:**
app.use(express.json());	Parse JSON request bodies.
app.use(express.urlencoded({extended: true }));	Parse URL-encoded request bodies.
app.use((req, res, next) => {res.setHeader('Content-Type', 'application/json; charset=utf-8'); next(); });	Set the response header to JSON format with UTF-8 encoding.
app.use(express.static('public'));	Deliver static files from the public directory.
app.use(morgan('dev'));	Log HTTP requests in "dev" format.
const corsOptions = {	
origin: function (origin, callback) {. . .};	Handle requests only for LAN devices, based on the request's origin or IP address.
app.use(cors(corsOptions));	
app.use(express.static(__dirname + '/public'));	Listen for HTTP requests with a path starting with/public. Deliver static files (e.g., HTML, CSS, JS).
app.use(express.urlencoded({ extended: true }));	Parse URL-encoded request bodies (e.g., form data).
paths = ["/geometries", "/textures", "/sounds", "/styles"];	
paths.forEach((path) => {	Deliver static files from directories corresponding to the paths:/geometries;/textures;/sounds;/styles.
app.use(path, express.static(__dirname + path));});	
app.use('/build/',express.static(path.join(__dirname, 'node_modules/three/build')));	Deliver Three.js build files from the build directory under the "/build" route.
app.use('/jsm/',express.static(path.join(__dirname, 'node_modules/three/examples/jsm')));	Deliver Three.js modules from the "jsm" directory under the "/jsm" route.

References

1. Reeves, T. Design Research from a Technology Perspective. In *Educational Design Research*; Routledge: London, UK, 2006; pp. 64–78.
2. Kouroupa, A.; Laws, K.R.; Irvine, K.; Mengoni, S.E.; Baird, A.; Sharma, S. The Use of Social Robots with Children and Young People on the Autism Spectrum: A Systematic Review and Meta-Analysis. *PLoS ONE* **2022**, *17*, e0269800. [CrossRef]
3. Mills, J.; Duffy, O. Speech and Language Therapists' Perspectives of Virtual Reality as a Clinical Tool for Autism: Cross-Sectional Survey. *JMIR Rehabil. Assist. Technol.* **2025**, *12*, e63235. [CrossRef]
4. Peng, L.; Nuchged, B.; Gao, Y. Spoken Language Intelligence of Large Language Models for Language Learning. *arXiv* **2025**, arXiv:2308.14536.
5. Austin, J.; Benas, K.; Caicedo, S.; Imiolek, E.; Piekutowski, A.; Ghanim, I. Perceptions of Artificial Intelligence and ChatGPT by Speech-Language Pathologists and Students. *Am. J. Speech-Lang. Pathol.* **2025**, *34*, 174–200. [CrossRef] [PubMed]
6. Leinweber, J.; Alber, B.; Barthel, M.; Whillier, A.S.; Wittmar, S.; Borgetto, B.; Starke, A. Technology Use in Speech and Language Therapy: Digital Participation Succeeds through Acceptance and Use of Technology. *Front. Commun.* **2023**, *8*, 1176827. [CrossRef]
7. Szabó, B.; Dirks, S.; Scherger, A.-L. Apps and Digital Resources in Speech and Language Therapy—Which Factors Influence Therapists' Acceptance? In *Universal Access in Human–Computer Interaction. Novel Design Approaches and Technologies. HCII 2022*; Antona, M., Stephanidis, C., Eds.; Springer: Cham, Switzerland, 2022; Volume 13308, pp. 379–391. [CrossRef]
8. Hastall, M.R.; Dockweiler, C.; Mühlhaus, J. Achieving End User Acceptance: Building Blocks for an Evidence-Based User-Centered Framework for Health Technology Development and Assessment. In *Universal Access in Human–Computer Interaction. Human and Technological Environments*; Antona, M., Stephanidis, C., Eds.; Springer: Cham, Switzerland, 2017; pp. 13–25. [CrossRef]
9. Spitale, M.; Silleresi, S.; Garzotto, F.; Mataric, M. Using Socially Assistive Robots in Speech-Language Therapy for Children with Language Impairments. *Int. J. Soc. Robot.* **2023**, *15*, 1525–1542. [CrossRef]
10. Rupp, R.; Wirz, M. Implementation of Robots into Rehabilitation Programs: Meeting the Requirements and Expectations of Professional and End Users. In *Neurorehabilitation Technology*, 2nd ed.; Reinkensmeyer, D.J., Juneal-Crespo, L., Dietz, V., Eds.; Springer: Cham, Switzerland, 2022; pp. 263–288.
11. Lampropoulos, G.; Keramopoulos, E.; Diamantaras, K.; Evangelidis, G. Augmented Reality and Virtual Reality in Education: Public Perspectives, Sentiments, Attitudes, and Discourses. *Educ. Sci.* **2022**, *12*, 798. [CrossRef]
12. Suh, H.; Dangol, A.; Meadan, H.; Miller, C.A.; Kientz, J.A. Opportunities and Challenges for AI-Based Support for Speech-Language Pathologists. In Proceedings of the 3rd Annual Symposium on Human—Computer Interaction for Work (CHIWORK 2024), New York, NY, USA, 9–11 June 2024; pp. 1–14. [CrossRef]
13. Du, Y.; Juefei-Xu, F. Generative AI for Therapy? Opportunities and Barriers for ChatGPT in Speech-Language Therapy. *Unpubl. Work 2023.* Available online: https://openreview.net/forum?id=cRZSr6Tpr1S (accessed on 20 June 2025).
14. Fu, B.; Hadid, A.; Damer, N. Generative AI in the Context of Assistive Technologies: Trends, Limitations and Future Directions. *Image Vis. Comput.* **2025**, *154*, 105347. [CrossRef]
15. Santos, L.; Annunziata, S.; Geminiani, A.; Ivani, A.; Giubergia, A.; Garofalo, D.; Caglio, A.; Brazzoli, E.; Lipari, R.; Carrozza, M.C.; et al. Applications of Robotics for Autism Spectrum Disorder: A Scoping Review. *Rev. J. Autism Dev. Disord.* **2023**. [CrossRef]
16. Sáez-López, J.M.; del Olmo-Muñoz, J.; González-Calero, J.A.; Cózar-Gutiérrez, R. Exploring the Effect of Training in Visual Block Programming for Preservice Teachers. *Multimodal Technol. Interact.* **2020**, *4*, 65. [CrossRef]
17. Rees Lewis, D.; Carlson, S.; Riesbeck, C.; Lu, K.; Gerber, E.; Easterday, M. The Logic of Effective Iteration in Design-Based Research. In Proceedings of the 14th International Conference of the Learning Sciences: The Interdisciplinarity of the Learning Sciences (ICLS 2020), Nashville, TN, USA, 19–23 June 2020; Gresalfi, M., Horn, I.S., Eds.; International Society of the Learning Sciences: Bloomington, IN, USA; Volume 2, pp. 1149–1156.
18. ATLog Project. Available online: https://atlog.ir.bas.bg/en (accessed on 20 June 2025).
19. Lekova, A.; Tsvetkova, P.; Andreeva, A.; Simonska, M.; Kremenska, A. System Software Architecture for Advancing Human-Robot Interaction by Cloud Services and Multi-Robot Cooperation. *Int. J. Inf. Technol. Secur.* **2024**, *16*, 65–76. [CrossRef]
20. Andreeva, A.; Lekova, A.; Tsvetkova, P.; Simonska, M. Expanding the Capabilities of Robot NAO to Enable Human-Like Communication with Children with Speech and Language Disorders. In Proceedings of the 25th International Conference on Computer Systems and Technologies (CompSysTech 2024), Sofia, Bulgaria, 28–29 June 2024; pp. 63–68. [CrossRef]
21. Lekova, A.; Tsvetkova, P.; Andreeva, A. System Software Architecture for Enhancing Human-Robot Interaction by Conversational AI. In Proceedings of the 2023 International Conference on Information Technologies (InfoTech), Varna, Bulgaria, 19–20 October 2023; pp. 1–6. [CrossRef]
22. Lekova, A.; Vitanova, D. Design-Based Research for Streamlining the Integration of Text-Generative AI into Socially-Assistive Robots. In Proceedings of the 2024 International Conference "ROBOTICS & MECHATRONICS", Sofia, Bulgaria, 29–30 October 2024.

23. Kolev, M.; Trenchev, I.; Traykov, M.; Mavreski, R.; Ivanov, I. The Impact of Virtual and Augmented Reality on the Development of Motor Skills and Coordination in Children with Special Educational Needs. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering: Computer Science and Education in Computer Science*; Springer Nature Switzerland: Cham, Switzerland, 2023; Volume 514, pp. 171–181. [CrossRef]
24. Brooke, J. SUS—A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*; CRC Press: Boca Raton, FL, USA, 1996; pp. 189–194.
25. Davis, F. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.* **1989**, *13*, 319–340. [CrossRef]
26. Cheah, W.; Jusoh, N.; Aung, M.; Ab Ghani, A.; Rebuan, H.M.A. Mobile technology in medicine: Development and validation of an adapted system usability scale (SUS) questionnaire and modified technology acceptance model (TAM) to evaluate user experience and acceptability of a mobile application in MRI safety screening. *Indian J. Radiol. Imaging* **2023**, *33*, 36–45. [CrossRef] [PubMed]
27. Hariri, W. Unlocking the Potential of ChatGPT: A Comprehensive Exploration of Its Applications, Advantages, Limitations, and Future Directions in Natural Language Processing. *arXiv* **2023**, arXiv:2304.02017.
28. Vaezipour, A.; Aldridge, D.; Koenig, S.; Theodoros, D.; Russell, T. "It's Really Exciting to Think Where It Could Go": A Mixed-Method Investigation of Clinician Acceptance, Barriers and Enablers of Virtual Reality Technology in Communication Rehabilitation. *Disabil. Rehabil.* **2022**, *44*, 3946–3958. [CrossRef]
29. Hashim, H.U.; Md Yunus, M.; Norman, H. Augmented Reality Mobile Application for Children with Autism: Stakeholders' Acceptance and Thoughts. *Arab World Engl. J.* **2021**, *12*, 130–146. [CrossRef]
30. Rasheva-Yordanova, K.; Kostadinova, I.; Georgieva-Tsaneva, G.; Andreeva, A.; Tsvetkova, P.; Lekova, A.; Stancheva-Popkostandinova, V.; Dimitrov, G. A Comprehensive Review and Analysis of Virtual Reality Scenarios in Speech and Language Therapy. *TEM J.* **2025**, *14*, 1895–1907. [CrossRef]
31. Vanderborght, B.; Simut, R.; Saldien, J.; Pop, C.; Rusu, A.S.; Pintea, S.; Lefeber, D.; David, D.O. Using the Social Robot Probo as a Social Story Telling Agent for Children with ASD. *Interact. Stud.* **2012**, *13*, 348–372. [CrossRef]
32. LuxAI. QTrobot for Schools & Therapy Centers. Available online: https://luxai.com/robot-for-teaching-children-with-autism-at-home/ (accessed on 20 June 2025).
33. Furhat Robotics. Furhat AI. Available online: https://www.furhatrobotics.com/furhat-ai (accessed on 20 June 2025).
34. PAL Robotics. ARI. Available online: https://pal-robotics.com/robot/ari/ (accessed on 20 June 2025).
35. RoboKind Milo Robot. Available online: https://www.robokind.com/?hsCtaTracking=2433ccbc-0af8-4ade-ae0b-4642cb0ecba4%7Cb0814dc8-76e1-447d-aba7-101e89845fb7 (accessed on 20 June 2025).
36. She, T.; Ren, F. Enhance the Language Ability of Humanoid Robot NAO through Deep Learning to Interact with Autistic Children. *Electronics* **2021**, *10*, 2393. [CrossRef]
37. Cherakara, N.; Varghese, F.; Shabana, S.; Nelson, N.; Karukayil, A.; Kulothungan, R.; Farhan, M.; Nesset, B.; Moujahid, M.; Dinkar, T.; et al. FurChat: An Embodied Conversational Agent Using LLMs, Combining Open and Closed-Domain Dialogue with Facial Expressions. In Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2023), Prague, Czech Republic, 11–13 September 2023; pp. 588–592. [CrossRef]
38. Belda-Medina, J.; Calvo-Ferrer, J.R. Using Chatbots as AI Conversational Partners in Language Learning. *Appl. Sci.* **2022**, *12*, 8427. [CrossRef]
39. ChatGPT Prompting for SLPs. Available online: https://eatspeakthink.com/chatgpt-for-speech-therapy-2025/ (accessed on 20 June 2025).
40. Bertacchini, F.; Demarco, F.; Scuro, C.; Pantano, P.; Bilotta, E. A Social Robot Connected with ChatGPT to Improve Cognitive Functioning in ASD Subjects. *Front. Psychol.* **2023**, *14*, 1232177. [CrossRef] [PubMed]
41. Salhi, I.; Qbadou, M.; Gouraguine, S.; Mansouri, K.; Lytridis, C.; Kaburlasos, V. Towards Robot-Assisted Therapy for Children with Autism—The Ontological Knowledge Models and Reinforcement Learning-Based Algorithms. *Front. Robot. AI* **2022**, *9*, 713964. [CrossRef] [PubMed]
42. Naneva, S.; Sarda Gou, M.; Webb, T.L.; Prescott, T.J. A Systematic Review of Attitudes, Anxiety, Acceptance, and Trust Towards Social Robots. *Int. J. Soc. Robot.* **2020**, *12*, 1179–1201. [CrossRef]
43. Liu, Z.; Li, H.; Chen, A.; Zhang, R.; Lee, Y.C. Understanding Public Perceptions of AI Conversational Agents. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI 2024), Honolulu, HI, USA, 11–16 May 2024; pp. 1–17. [CrossRef]
44. Gerlich, M. Perceptions and Acceptance of Artificial Intelligence: A Multi-Dimensional Study. *Soc. Sci.* **2023**, *12*, 502. [CrossRef]
45. Severson, R.; Peter, J.; Kanda, T.; Kaufman, J.; Scassellati, B. Social Robots and Children's Development: Promises and Implications. In *Handbook of Children and Screens*; Christakis, D.A., Hale, L., Eds.; Springer: Cham, Switzerland, 2024; pp. 627–633. [CrossRef]

46. Youssef, K.; Said, S.; Alkork, S.; Beyrouthy, T. A Survey on Recent Advances in Social Robotics. *Robotics* **2022**, *11*, 75. [CrossRef]
47. Georgieva-Tsaneva, G.; Andreeva, A.; Tsvetkova, P.; Lekova, A.; Simonska, M.; Stancheva-Popkostadinova, V.; Dimitrov, G.; Rasheva-Yordanova, K.; Kostadinova, I. Exploring the Potential of Social Robots for Speech and Language Therapy: A Review and Analysis of Interactive Scenarios. *Machines* **2023**, *11*, 693. [CrossRef]
48. Estévez, D.; Terrón-López, M.-J.; Velasco-Quintana, P.J.; Rodríguez-Jiménez, R.-M.; Álvarez-Manzano, V. A Case Study of a Robot-Assisted Speech Therapy for Children with Language Disorders. *Sustainability* **2021**, *13*, 2771. [CrossRef]
49. Ioannou, A.; Andreva, A. Play and Learn with an Intelligent Robot: Enhancing the Therapy of Hearing-Impaired Children. In *Human–Computer Interaction—INTERACT 2019*; Lamas, D., Loizides, F., Nacke, L., Petrie, H., Winckler, M., Zaphiris, P., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11747, pp. 436–452. [CrossRef]
50. Lee, H.; Hyun, E. The Intelligent Robot Contents for Children with Speech-Language Disorder. *J. Educ. Technol. Soc.* **2015**, *18*, 100–113.
51. Lekova, A.; Andreeva, A.; Simonska, M.; Tanev, T.; Kostova, S. A System for Speech and Language Therapy with a Potential to Work in the IoT. In Proceedings of the 2022 International Conference on Computer Systems and Technologies (CompSysTech 2022), Ruse, Bulgaria, 16–17 June 2022; pp. 119–124. [CrossRef]
52. QTrobot for Special Needs Education. Available online: https://luxai.com/assistive-tech-robot-for-special-needs-education/ (accessed on 20 June 2025).
53. Vukliš, D.; Krasnik, R.; Mikov, A.; Zvekić Svorcan, J.; Janković, T.; Kovačević, M. Parental Attitudes Towards the Use of Humanoid Robots in Pediatric (Re)Habilitation. *Med. Pregl.* **2019**, *72*, 302–306. [CrossRef]
54. Szymona, B.; Maciejewski, M.; Karpiński, R.; Jonak, K.; Radzikowska-Büchner, E.; Niderla, K.; Prokopiak, A. Robot-Assisted Autism Therapy (RAAT): Criteria and Types of Experiments Using Anthropomorphic and Zoomorphic Robots. *Sensors* **2021**, *21*, 3720. [CrossRef]
55. Nicolae, G.; Vlădeanu, C.; Saru, L.-M.; Burileanu, C.; Grozăvescu, R.; Crăciun, G.; Drugă, S.; Hatiș, M. Programming the NAO Humanoid Robot for Behavioral Therapy in Romania. *Rom. J. Child Adolesc. Psychiatry* **2019**, *7*, 23–30.
56. Gupta, G.; Chandra, S.; Dautenhahn, K.; Loucks, T. Stuttering Treatment Approaches from the Past Two Decades: Comprehensive Survey and Review. *J. Stud. Res.* **2022**, *11*. [CrossRef]
57. Chandra, S.; Gupta, G.; Loucks, T.; Dautenhahn, K. Opportunities for Social Robots in the Stuttering Clinic: A Review and Proposed Scenarios. *Paladyn J. Behav. Robot.* **2022**, *13*, 23–44. [CrossRef]
58. Bonarini, A.; Clasadonte, F.; Garzotto, F.; Gelsomini, M.; Romero, M.E. Playful Interaction with Teo, a Mobile Robot for Children with Neurodevelopmental Disorders. In Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2016), Vila Real, Portugal, 1–3 December 2016; pp. 223–231. [CrossRef]
59. Kose, H.; Yorganci, R. Tale of a Robot: Humanoid Robot Assisted Sign Language Tutoring. In Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia, 26–28 October 2011; pp. 105–111. [CrossRef]
60. Fung, K.Y.; Lee, L.H.; Sin, K.F.; Song, S.; Qu, H. Humanoid Robot-Empowered Language Learning Based on Self-Determination Theory. *Educ. Inf. Technol.* **2024**, *29*, 18927–18957. [CrossRef]
61. Cappadona, I.; Ielo, A.; La Fauci, M.; Tresoldi, M.; Settimo, C.; De Cola, M.C.; Muratore, R.; De Domenico, C.; Di Cara, M.; Corallo, F.; et al. Feasibility and Effectiveness of Speech Intervention Implemented with a Virtual Reality System in Children with Developmental Language Disorders: A Pilot Randomized Control Trial. *Children* **2023**, *10*, 1336. [CrossRef]
62. Purpura, G.; Di Giusto, V.; Zorzi, C.F.; Figliano, G.; Randazzo, M.; Volpicelli, V.; Blonda, R.; Brazzoli, E.; Reina, T.; Rezzonico, S.; et al. Use of Virtual Reality in School-Aged Children with Developmental Coordination Disorder: A Novel Approach. *Sensors* **2024**, *24*, 5578. [CrossRef]
63. Maresca, G.; Corallo, F.; De Cola, M.C.; Formica, C.; Giliberto, S.; Rao, G.; Crupi, M.F.; Quartarone, A.; Pidalà, A. Effectiveness of the Use of Virtual Reality Rehabilitation in Children with Dyslexia: Follow-Up after One Year. *Brain Sci.* **2024**, *14*, 655. [CrossRef] [PubMed]
64. Mangani, G.; Barzacchi, V.; Bombonato, C.; Barsotti, J.; Beani, E.; Menici, V.; Ragoni, C.; Sgandurra, G.; Del Lucchese, B. Feasibility of a Virtual Reality System in Speech Therapy: From Assessment to Tele-Rehabilitation in Children with Cerebral Palsy. *Children* **2024**, *11*, 1327. [CrossRef] [PubMed]
65. Macdonald, C. Improving Virtual Reality Exposure Therapy with Open Access and Overexposure: A Single 30-Minute Session of Overexposure Therapy Reduces Public Speaking Anxiety. *Front. Virtual Real.* **2024**, *5*, 1506938. [CrossRef]
66. Pergantis, P.; Bamicha, V.; Doulou, A.; Christou, A.I.; Bardis, N.; Skianis, C.; Drigas, A. Assistive and Emerging Technologies to Detect and Reduce Neurophysiological Stress and Anxiety in Children and Adolescents with Autism and Sensory Processing Disorders: A Systematic Review. *Technologies* **2025**, *13*, 144. [CrossRef]

67. Tobii Dynavox Global: Assistive Technology for Communication. Available online: https://www.tobiidynavox.com/ (accessed on 20 June 2025).
68. Klavina, A.; Pérez-Fuster, P.; Daems, J.; Lyhne, C.N.; Dervishi, E.; Pajalic, Z.; Øderud, T.; Fuglerud, K.S.; Markovska-Simoska, S.; Przybyla, T.; et al. The Use of Assistive Technology to Promote Practical Skills in Persons with Autism Spectrum Disorder and Intellectual Disabilities: A Systematic Review. *Digit. Health* **2024**, *10*, 20552076241281260. [CrossRef]
69. Ask NAO. Available online: https://www.asknao-tablet.com/en/home/ (accessed on 20 June 2025).
70. Furhat Blockly. Available online: https://docs.furhat.io/blockly/ (accessed on 20 June 2025).
71. Vittascience Interface for NAO v6. Available online: https://en.vittascience.com/nao/?mode=mixed&console=bottom&toolbox=vittascience (accessed on 20 June 2025).
72. LEKA. Available online: https://leka.io (accessed on 20 June 2025).
73. Platform for VR Public Speaking. Available online: https://www.virtualrealitypublicspeaking.com/platform (accessed on 20 June 2025).
74. ThingLink. Available online: https://www.thinglink.com/ (accessed on 20 June 2025).
75. Kurai, R.; Hiraki, T.; Hiroi, Y.; Hirao, Y.; Perusquía-Hernández, M.; Uchiyama, H.; Kiyokawa, K. MagicItem: Dynamic Behavior Design of Virtual Objects with Large Language Models in a Commercial Metaverse Platform. *IEEE Access* **2025**, *13*, 19132–19143. [CrossRef]
76. Therapy withVR. Available online: https://therapy.withvr.app/ (accessed on 20 June 2025).
77. Grassi, L.; Recchiuto, C.T.; Sgorbissa, A. Sustainable Cloud Services for Verbal Interaction with Embodied Agents. *Intell. Serv. Robot.* **2023**, *16*, 599–618. [CrossRef]
78. LuxAI. Complete Guide to Build a Conversational Social Robot QTrobot with ChatGPT. Available online: https://luxai.com/blog/complete-guide-to-build-conversational-social-robot-qtrobot-chatgpt/ (accessed on 20 June 2025).
79. Lewis, J.R. The system usability scale: Past, present, and future. *Int. J. Hum. Comput. Interact.* **2018**, *34*, 577–590. [CrossRef]
80. Mulia, A.P.; Piri, P.R.; Tho, C. Usability analysis of text generation by ChatGPT OpenAI using system usability scale method. *Procedia Comput. Sci.* **2023**, *227*, 381–388. [CrossRef]
81. Bangor, A.; Kortum, P.T.; Miller, J.T. An empirical evaluation of the system usability scale. *Int. J. Hum. Comput. Interact.* **2008**, *24*, 574–594. [CrossRef]
82. Lewis, J.R. Comparison of Four TAM Item Formats: Effect of Response Option Labels and Order. *J. Usability Stud.* **2019**, *14*, 224–236.
83. Aldebaran. NAO Robot. Available online: https://aldebaran.com/en/nao6/ (accessed on 20 June 2025).
84. Furhat Robotics. The Furhat Robot. Available online: https://www.furhatrobotics.com/furhat-robot (accessed on 20 June 2025).
85. Tanev, T.K.; Lekova, A. Implementation of Actors' Emotional Talent into Social Robots Through Capture of Human Head's Motion and Basic Expression. *Int. J. Soc. Robot.* **2022**, *14*, 1749–1766. [CrossRef]
86. Double Robotics. Double3 Robot for Telepresence. Available online: https://www.doublerobotics.com/double3.html (accessed on 20 June 2025).
87. Aldebaran. Choregraphe Software Overview. Available online: http://doc.aldebaran.com/2-8/software/choregraphe/choregraphe_overview.html (accessed on 20 June 2025).
88. Aldebaran. NAOqi API and SDK 2.8. Available online: http://doc.aldebaran.com/2-8/news/2.8/naoqi_api_rn2.8.html?highlight=naoqi (accessed on 20 June 2025).
89. Furhat Robotics. Furhat SDK Documentation. Available online: https://docs.furhat.io/getting_started/ (accessed on 20 June 2025).
90. Furhat Robotics. Furhat Remote API. Available online: https://docs.furhat.io/remote-api/ (accessed on 20 June 2025).
91. Double Robotics. Double3 Developer SDK. Available online: https://github.com/doublerobotics/d3-sdk (accessed on 20 June 2025).
92. W3C. WebXR Device API. Available online: https://www.w3.org/TR/webxr/ (accessed on 20 June 2025).
93. Dirksen, J. *Learning Three.js—The JavaScript 3D Library for WebGL*, 2nd ed.; Packt Publishing: Birmingham, UK, 2015; ISBN 978-1-78439-221-5.
94. 3DVista Virtual Tour Software. Available online: https://www.3dvista.com/en/ (accessed on 20 June 2025).
95. NLP Cloud API Platform. Available online: https://nlpcloud.com/ (accessed on 20 June 2025).
96. OpenAI. ChatGPT-4 Research Overview. Available online: https://openai.com/index/gpt-4-research/ (accessed on 20 June 2025).
97. INSAIT. BgGPT Language Model. Available online: https://bggpt.ai/ (accessed on 20 June 2025).
98. Express.js. Node.js Web Application Framework. Available online: https://expressjs.com/ (accessed on 20 June 2025).
99. Google Developers. Blockly for Developers. Available online: https://developers.google.com/blockly (accessed on 20 June 2025).
100. Node.js. SQLite Module Documentation. Available online: https://nodejs.org/api/sqlite.html (accessed on 20 June 2025).
101. Meta. Casting from Meta Quest 3 to Computer. Available online: https://www.oculus.com/casting/ (accessed on 20 June 2025).
102. ATLog Ethical Codex. Available online: https://atlog.ir.bas.bg/en/results/ethical-codex (accessed on 20 June 2025).

103. Vlachogianni, P.; Tselios, N. Perceived Usability Evaluation of Educational Technology Using the Post-Study System Usability Questionnaire (PSSUQ): A Systematic Review. *Sustainability* **2023**, *15*, 12954. [CrossRef]
104. Lah, U.; Lewis, J.R.; Šumak, B. Perceived usability and the modified technology acceptance model. *Int. J. Hum. Comput. Interact.* **2020**, *36*, 1216–1230. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.