UDC 004.35:004.5

Nazarenko Volodymyr
Ph.D., Computer Systems, Networks and Cybersecurity Department,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0000-0002-7433-2484
E-mail: *volodnz@nubip.edu.ua*

Ostroushko Bogdan
M.S., assistant Computer Systems, Networks and Cybersecurity Department,
National University of Life and Environmental Sciences of Ukraine
ORCID: https://orcid.org/0009-0003-0849-8990
E-mail: *b.ostroushko@nubip.edu.ua*

ENHANCING HUMAN-COMPUTER INTERACTION: THE ROLE OF INTELLIGENT USER INTERFACES IN EMBEDDED AND IOT SYSTEMS

Abstract. *The rapid evolution of innovative technologies has significantly transformed digital applications and human activities. Advances in hardware, digital software, and interactive user interfaces have made user experience (UX) design a critical factor in various industries, including healthcare, aerospace, construction, and military sectors. Human-Computer Interaction (HCI) is vital in enhancing system usability by ensuring intelligent interaction systems are user-centric and accessible. Integrating intelligent systems, the Internet of Things (IoT), and embedded technologies presents new challenges and opportunities in interface design. Additionally, as industries adopt intelligent automation and digital transformation strategies, the demand for adaptive, user-friendly interfaces grows. Understanding how users interact with these technologies is essential to optimizing efficiency and improving overall user satisfaction. This study explores the principles of HCI and interaction design, analyzing how digital and analog interfaces contribute to practical system usability. A qualitative approach was adopted, reviewing contemporary research on user interface (UI) design, interactive systems, and intelligent applications. The study also considers case studies from various industries where UX and interface optimization have significantly impacted system performance and user satisfaction. Data collection included literature analysis, expert interviews, and usability testing reports, ensuring a comprehensive understanding of the challenges and advancements in interaction design. The research methodology also incorporates a comparative analysis of traditional and emerging interface models to identify best practices in designing adaptive and intelligent systems. Findings indicate that well-designed interfaces enhance user engagement, efficiency, and system functionality. Factors influencing usability include adaptive design, accessibility considerations, and intuitive user experiences. Case studies reveal that industries incorporating advanced UX principles and innovative interaction technologies demonstrate improved operational efficiency and user satisfaction. Moreover, findings suggest that integrating AI-driven assistance, real-time feedback mechanisms, and multimodal interaction significantly improves user adaptability and system effectiveness. The study also highlights the importance of cognitive load reduction in interface design, emphasizing strategies such as predictive analytics and context-aware computing to enhance user interactions. The study highlights the Importance of HCI in designing user-friendly and efficient systems. It emphasizes the need for continuous innovation in UX and interface design, particularly in emerging fields such as IoT and embedded systems. The discussion underscores the role of AI, machine learning, and augmented reality in shaping the future of user interactions. Future research should focus on further integrating AI-driven personalization and adaptive interfaces to improve user experience. The findings suggest that interdisciplinary collaboration between engineers, designers, and cognitive scientists is essential to developing more effective and human-centered interaction systems. Addressing ethical considerations, such as data privacy and accessibility, will ensure equitable and inclusive design in future technological advancements.*

Keywords: *user interface design, user experience, smart systems, system interaction, computer engineering*.

Introduction. Innovative technologies have changed software and digital applications, rapidly transforming most modern human activities. The state-of-the-art advancements in hardware, contemporary material and production, digital software technologies, interactive digital devices, and user interfaces have become integral to the user experience interaction design. The areas of interaction design encompass most of the general user and professional applications across industries, such as construction, aerospace, healthcare, and military industries. Intelligent interaction systems are user-centric. They enable applications (systems) interfaces to be usable by humans in the case of either digital or analog systems, mechanical and purely virtual interaction devices.

Human-Computer Interaction (HCI)studies how to improve integration systems and their interfaces. Besides, it is an area of active research for the new methodologies to refine or redesign existing interaction processes and their underlying mechanics for enhanced user-system interaction experiences. Another significant benefit of the HCI field is task optimization and user-machine interaction efficiency. User-system interaction, in most cases, is a cyclical process that can be characterized as a series of continuous or step actions followed by feedback from the system. The iterative design pattern is not limited to input-output cycles; it influences the software, hardware, system design, and development processes.

Human-computer interaction systems are part of the extended software and hardware systems architecture process. However, it is a complex process that can and should be conducted separately before and during the system development pipeline. Interaction and feedback loops are central to user task processing and digital engineering system design management. A vital system designer’s task is to align system functionalities with user requirements and projected content of use. The feedback loop is a standalone component vital to improving user workflows. Feedback systems can serve as a supporting pillar to teach users about the systems’ nature, capabilities, limitations, and affordances, all of which ensure the long-term adaptability of interactive systems.

The importance of well-designed interaction and feedback loops in large interaction processing systems has been widely acknowledged. By utilizing these loops, the system designer can provide a user with an understandable interface, which in turn makes the whole system give a real-time response based on user input, which enables the user to make rational in-time decisions for a single task or help guide and support user support in case of numerous system states iterations. Interaction and feedback are parts of a larger complex system design engineering tasks. Such systems support input and output mechanisms, such as visual, auditory, haptic, and mixed-reality feedback mechanisms. Another task of the feedback subsystem is to offer users insights into system behaviors, prevent some (at least the most critical) errors, and teach the users to refine their system knowledge and make tasks faster and more seamless. The iterative interaction process as a whole promotes system usage efficiency. It reduces the cognitive load on user innovation, ensuring that digital interfaces and the system UI components are designed with the end-user in mind. Potential interactions are well aligned with their intended applications.

Purpose. This research aims to provide the current state of human-computer interaction research, focusing on interaction and feedback loops, breaking it into functional components, and presenting a robust model for interaction design context, interaction process, and the view on User Interaction – Feedback Systems and States.

Literature review. Undoubtedly, HCI and user interface design are prominent and novel areas for research. Many scientists and engineers are actively engaged in human-computer interaction research. The area of their research work ranges widely across industries and fields, including various areas such as 3D interaction, the Internet of Things, voice interface, ML, AI in interface design, mixed reality, and intelligent systems.

Borchers (2000) highlighted a novel at the time, a pattern-based approach to design interaction systems; he focused on methodologies that could improve interface (systems) usability in complex use-case environments such as music and education [1]. More modern but similar studies conducted by Löwgren in 2013 showed an in-depth breakdown of human-computer interaction issues, focusing on the interdisciplinary nature of the systems design process [2]. The author noted the role of interaction design methodology in shaping the user’s experience to make it seemingly intuitive. Both works are fundamental, highlighting the critical role of interactive design as a novel methodology. Interaction design can be considered not only as a technical but also as a creative component in the field of HCI, introducing the balance between technical and user-centered perspectives.

The integration of the latest technologies, such as 3D interfaces, virtual reality, voice interfaces, big data, and wireless communications, to name a few, has led to the expansion of modern interaction design and its applications. One of the earliest researchers in this area, Bowman, introduced in 2001 the principles of designing a 3D user interface for enchanting virtual environments and providing more interaction options through 3D graphics [3]. Expanding on Bowman’s work, LaViola 2008 showed the accessibility of spatial 3D interaction using video games as a medium, reducing the gap between digital interfaces' technological implementation complexity and making complex user interfaces available to an enormous scope of designers and users [4]. As a continuation of previous advancements in material science and computer systems technologies, Zhu worked on solving the interoperability issues between IoT and HCI [5]. In his work, he presented the 3D-printed triboelectric sensor, which had been printed for this task for gait analysis and virtual inspection. As a result, this enabled the interaction systems to combine a data-centered architecture with the intended functionality of the user interface system. All the research works above highlighted the impact and importance of 3D graphics and IoT technologies in HCI research and interaction systems design [6, 7].

Feedback and interaction loops have also been studied from the perspective of system tasks and resource optimization. Fischer (2022), in his research work, used the optimal feedback control method to simulate user-system interaction dynamics, while Ritter (2011) investigated the cognitive benefits of subliminal feedback loops for reducing the user's mental load [8, 9]. Machine learning advances have greatly benefited feedback loop design and conceptual interface user testing. Honeycutt, in 2020, examined the implementation of machine learning algorithms using human-in-loop feedback mechanisms and identified interaction process problems [10]. Among the errors and issues found, balancing system usability with user experience in the context of seemingly natural interaction processes was the most critical. At the same time, Palanque (2020) classified interaction faults in system feedback loops, highlighting the impact of software system (application) reliability on safety issues [11]. These works highlight the critical role of feedback in improving user experience, maintaining system reliability, and increasing trust.

Interdisciplinary approaches provide a great field of potential topics that can be used to solve existing HCI issues and look for novel, groundbreaking approaches [12]. Psychology, sociology, and neural sciences are interviewed in a close context with user experience design [13, 14]. Costabile (2007) presented a new methodology that classified visual interactive systems design models that combined user-application requirements with software system design [15]. Ju and Leifer (2008) advocated for a more intuitive interaction design that considers the user interaction loop approach to help minimize user-application interaction frustration [16]. Hollender (2010) combined cognitive load theory with HCI concepts to improve educational applications [17]. A new paper by Ramirez (2024) on advances in natural language processing further illustrates how AI-driven technologies can transform interaction paradigms [18]. Most scientific works emphasize the requirements for new interaction systems to be more intelligent and user-driven and consider human behavior's cognitive and mental aspects. Such studies by many authors illustrate a diverse approach to HCI research, embracing new methodologies, technologies, and interdisciplinary connections. The research synthesis provides the basis for the design of advanced, user-friendly interaction systems in an increasingly complex digital age.

Methods. A comprehensive overview of the primary data, components, and systems used in human-computer interaction (HCI) research is summarized in Table 1. HCI is facilitated through diverse mechanical, visual, and auditory devices. Major interaction systems comprise digital, electronic, and electrical components for input and output operations and data and signal processing modules. Each interaction type corresponds to a specific sample system, such as LCDs, touchscreen interfaces, audio speakers, and spatial sensors.

User, software, and hardware systems design incorporates data and programming functions. Interaction data types are categorized based on their interaction mode. Additionally, the most common user-interface interaction functions, derived from Table 1, are outlined, highlighting their ability to accept input data and provide result outputs. A brief classification of human-computer interaction underlying systems:

- Direct input (output) – screen, mechanical keyboard, sound, and voice recording devices;
- Context-aware systems – scanner device, spatial sensor, digital camera, mixed reality devices;

- Smart data-driven system – semi-automated, autonomous decision support system, predictive behavior, and state systems.

*Table 1– Human-computer interaction data, systems, and components**

HCI system	Components	Data
Manual mechanical	Device, Sensor, Signal processing	Analog Signal, Device data, spatial data
Hardware	Smart Sensors Monitoring and recording devices Embedded systems Industrial devices	Signals Numerical Logs Commands/Tasks
Communications and Data	Wired/Wireless Smart Grid Gateway Data and Protocols	IP addresses Data packets Segments Connection Status Message
Analog bw screen	physical input, output	Screen size, text, language
Sound	audio processor, speaker, recorder	wave data, timing, compression algo.
Visual digital screen	GPU, screen, renderer	geometry data, screen data, FX, 3D coordinates
Smart	computational unit, intelligent system	parameters, state, visualization

* prepared based on the author's work and public research data

Researchers and industry professionals use various tools to develop novel interaction systems. The most frequently employed hardware and software solutions range from - mixed reality glasses, sensor screens, mobile phones, tablet devices, mechanical interaction devices (joystick, gamepad, mouse, and keyboard), misc. Sensors and trackers devices for detection and recognition tasks, spatial motion controllers, and other digital or analog devices.

HCI and interaction design methodologies are grounded in structured workflows. The product or application design process begins with business and product planning, proceeds through design and technology development, and culminates in user-facing outputs. These outputs consist of structured visual and textual elements accessible through menus and navigation panels.

HCI involves a three-phase process, as illustrated in Figure 1. The initial phase begins with a user-defined goal, representing the objective achieved through interactive devices, including digital, analog, mechanical, or miscellaneous types. User input is processed by a software system (e.g., a computer or microcomputer) based on the interaction medium and input type. The underlying components, actions, and conceptual functions associated with each interaction phase are summarized in the lower section of Figure 1, categorized by their respective interaction phases.

Figure 2 provides a comprehensive framework for understanding an IoT ecosystem's key components and interconnections. It outlines the layered structure of IoT systems, beginning with the foundational technologies such as sensors, actuators, communication protocols, and cloud infrastructure that form the backbone of data collection, transmission, and storage. These technologies enable seamless data exchange between physical devices and digital platforms. Above this technological foundation lies the services layer, which includes real-time analytics, machine learning algorithms, and application programming interfaces (APIs) that process, interpret, and derive actionable insights from IoT-generated data.

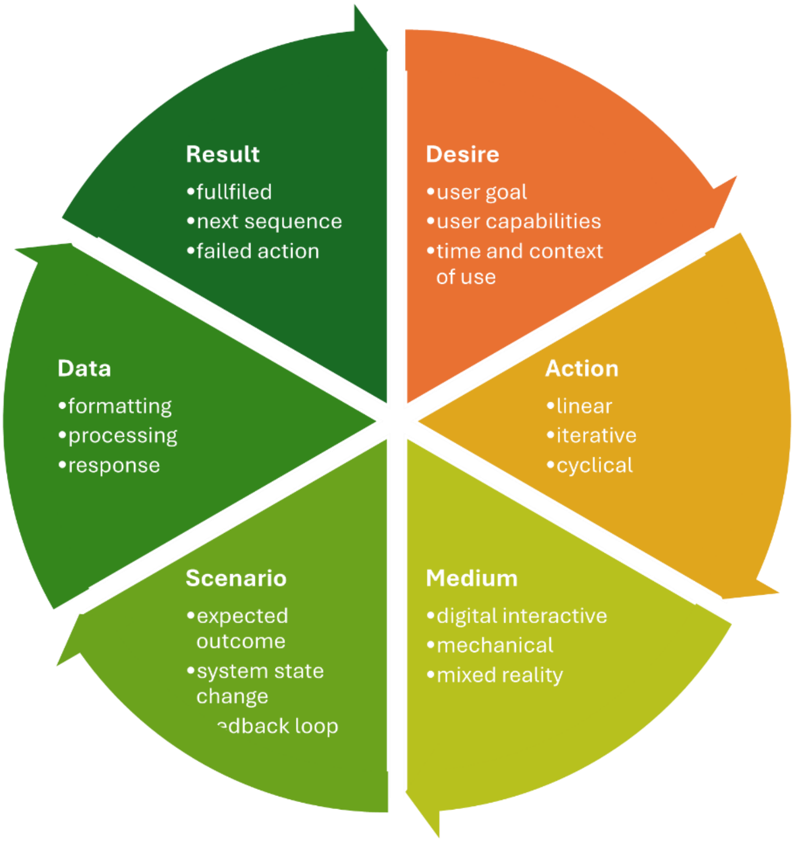

Figure 1 – Human-Computer Interaction System Process

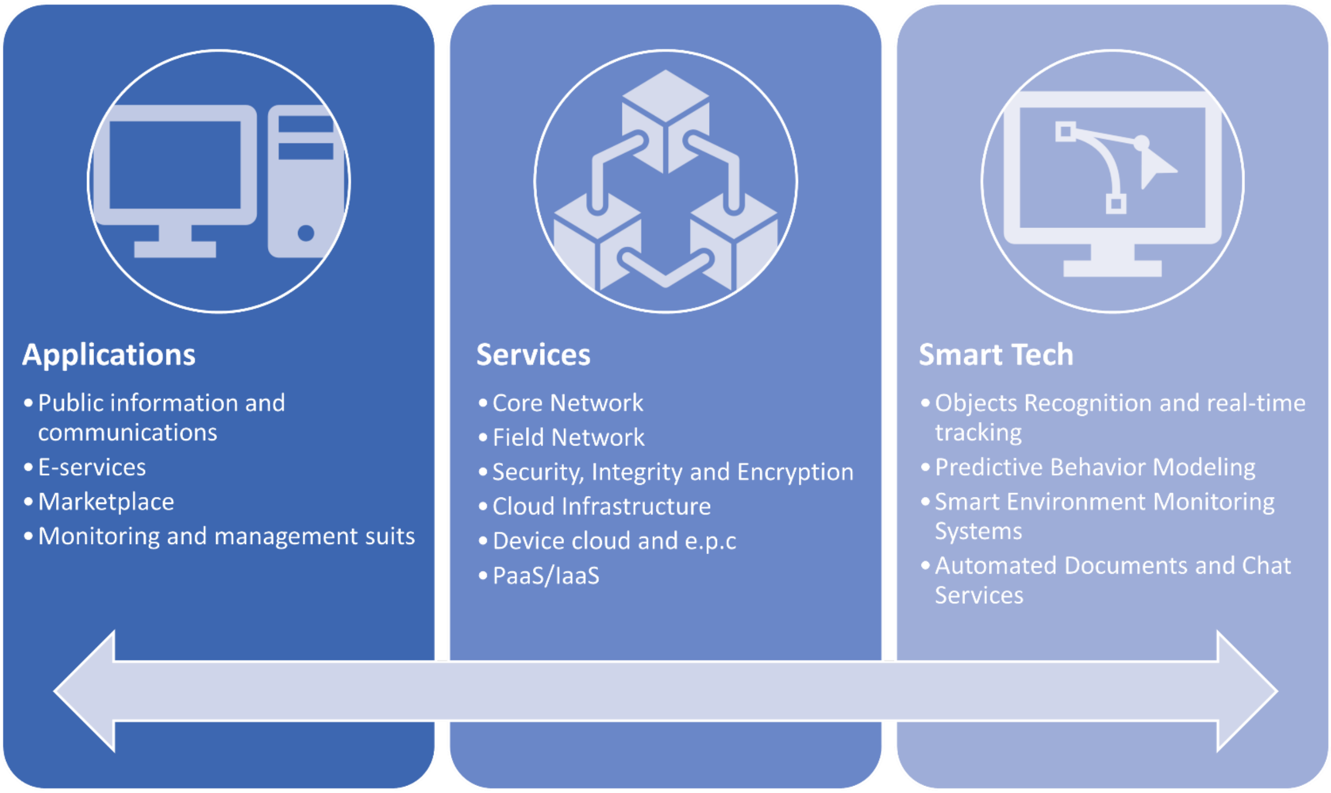

Figure 2 – Applications, services, and technologies for IoT system design and development

These services facilitate functionalities such as predictive maintenance, anomaly detection, and automation, which are essential for scalable IoT deployments. The applications layer, positioned at the top, highlights the diverse use cases of IoT across industries, including smart cities, healthcare, agriculture, manufacturing, and logistics. Each application leverages specific services and technologies to address industry-specific challenges, such as optimizing resource usage in smart homes, enabling remote health monitoring in healthcare, or enhancing supply chain efficiency in logistics. By categorizing the IoT ecosystem into applications, services, and technologies, the diagram provides a systematic approach to IoT system design, illustrating the dependencies and flow of data between layers while emphasizing the flexibility and scalability of these systems in real-world scenarios. This layered model also underscores the importance of interoperability and security considerations at every level, ensuring robust, efficient, and reliable IoT solutions.

Results. The human - user is at the core of software, applications, and system interaction design. The interaction process is initiated and sustained by the user's desire to achieve one or more specific goals (system tasks). Interaction systems serve as tools that facilitate the attainment of these objectives. Figure 3 demonstrates the user-centered interaction loop. The emphasis is placed on the feedback loop, which serves as an integral of the interaction process. This iterative processing system comprises the user interface components, input and output processing functions, and the transmission of commands to the primary system processor.

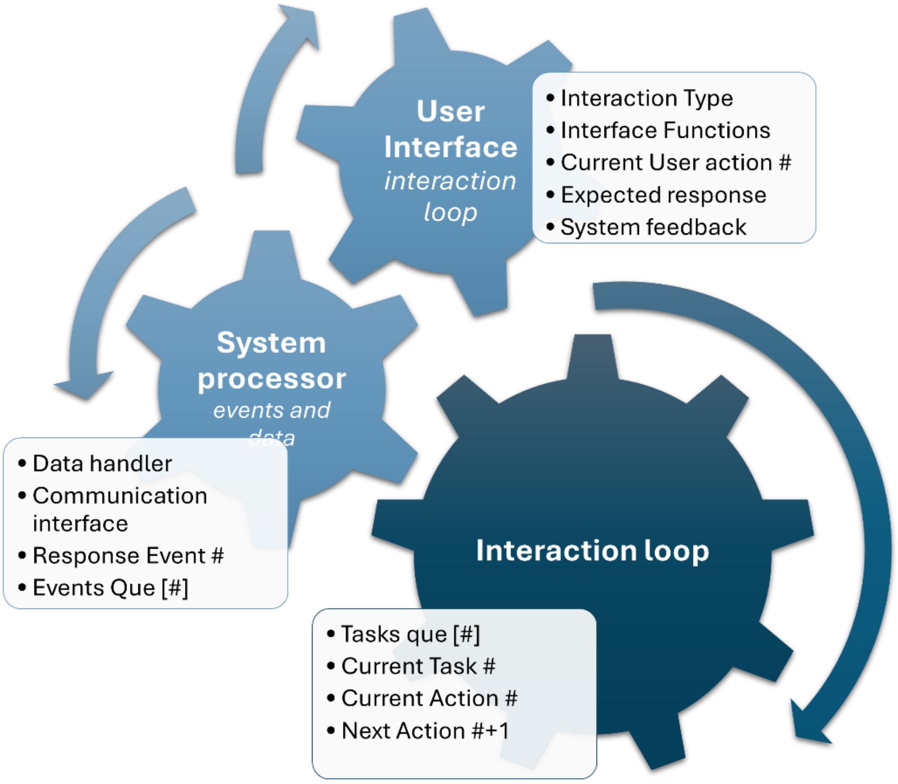

Figure 3 – Human application interaction system

Interaction processes do not occur in isolation but are embedded within real-world environments, a consideration that is particularly critical in industrial, medical, and mobile applications (professional usage). Figure 4 highlights the contextual factors (context of use for short) that should be considered for system design. Context of use influences how the user perceives and interacts with the application (system). An interactive iterative system must account for the environmental context, including the time, location, and conditions under which the device or application is utilized. Task-specific requirements and interaction activities mold user behavior, introducing system-level constraints and user-interaction limitations. Differentiating (extensive classification and research) user types and system usage purposes is crucial in the early stages of new software application design. Industry-specific environments and professional usage scenarios often prioritize tasks to the end user's goals, which leads to sequential changes in user-system interaction flow. Figure 4 provides an in-depth visualization of these factors.

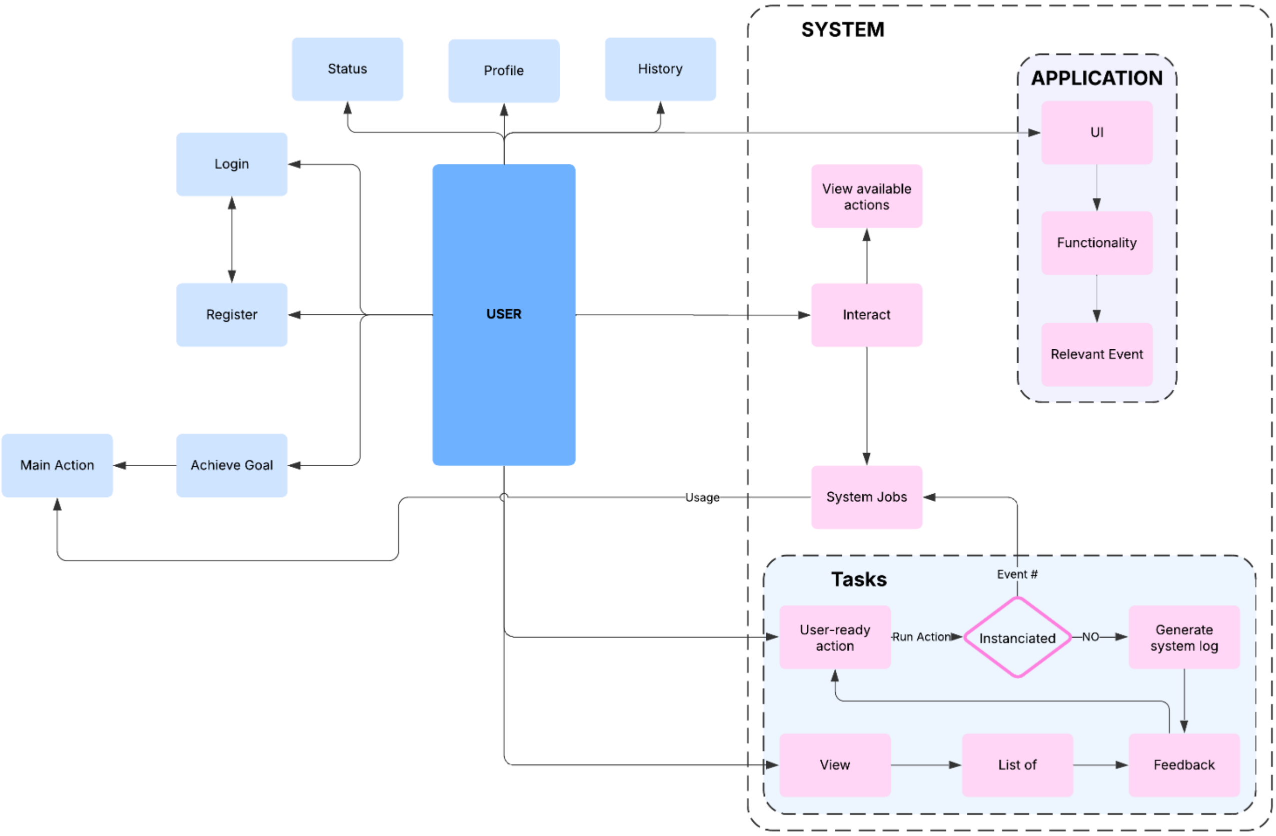

Figure 4 – Human application interaction system

The most crucial phases of human-system interaction, interaction, and feedback loops are presented in Figure 5. During this phase, two primary actors — the user and the computational system engage in an interaction cycle, with the system being a medium and processor within this context. Green highlighted rectangles represent Central System States, located at the diagram's center. The interaction process begins with a user action (task/job). Examples of system action can be inputting information or initiating task threads. The event queue and event manager record and process each sequential user action. As the user interacts with the application, he awaits a system response (feedback). In the meantime, the computational system actively participates in the loop by generating appropriate system-level responses and communication messages. This interaction process is part of a broader architecture encompassing server/client applications and database services, as illustrated in the lower section of Figure 5.

The six core modules (interaction phrases) of the interaction process are outlined in Table 2. Each interaction loop begins with system initialization (boot/loading) and concludes with system shutdown.

Once the system is operational, it activates the appropriate user interface or input-output system, enabling user interaction. The next-in-line phase involves the user-system interaction and feedback loop, which operate concurrently. Feedback and interaction can consist of an arbitrary number of steps determined dynamically by the user and the system. During each interaction cycle, the user can proceed to the next step, revert to the initial step, or terminate the system entirely. The loop concludes when the user ends the interaction; at this point, the system saves its state and powers down. Alternatively, the system may transition to a background operational state, monitoring for new inputs or events.

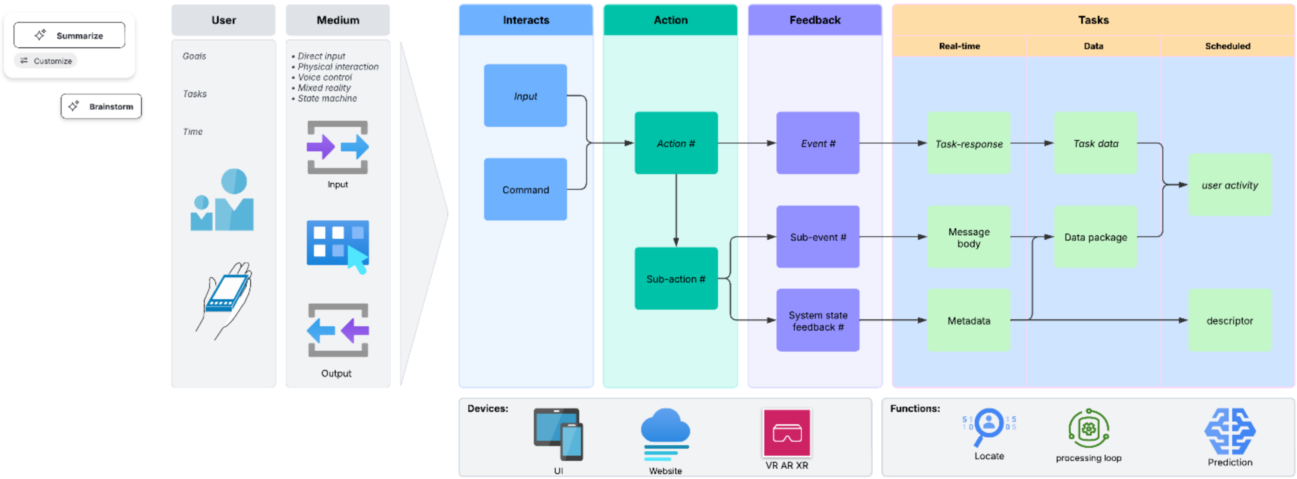

Figure 5 – Interaction process software system model

*Table 2 – Interaction and feedback system modules**

System	Components	Data
Initial - loading state	User, device, power control	Saved and system initial parameters
Main user interface	GUI, manipulation device, interaction system	Text, graphical data, input data
Interaction cycle	Tasks manager, Event manager	activities, attributes, action-request
System response	Communication and post-processing	system state and comms data
Support system	Action evaluation, Feedback control, Prediction	metadata, control parameters
Final - shutdown state	Data and System state manager	saved user and device data

* prepared based on the author's work and public research data

Figure 6 provides an in-depth representation of the key architectural elements and their roles in facilitating interactive IoT system development. Middleware is a critical intermediary layer, bridging the gap between IoT devices, networks, and application services by providing a standardized platform for managing data, devices, and user interactions. Core components of this middleware include device management modules, which handle registration, authentication, and configuration of IoT devices, ensuring secure and reliable operation within the system. Data processing and storage modules are pivotal in collecting, filtering, and storing vast amounts of sensor data for subsequent analysis, employing techniques such as real-time data streaming and machine learning to derive actionable insights. Communication protocols like MQTT, CoAP, and HTTP enable efficient data exchange between heterogeneous IoT devices and the middleware platform, ensuring interoperability across diverse technologies [8].

Interactive system development is further supported by application programming interfaces (APIs) and software development kits (SDKs), which allow developers to integrate custom user interfaces, implement real-time feedback loops, and design user-centric interactions. These APIs enable seamless communication between the middleware and higher-level services or applications, ensuring modular and scalable system design. Event-driven architectures within middleware enhance system responsiveness by triggering predefined actions based on real-time sensor inputs, essential for interactive applications such as smart home systems or IoT-driven healthcare monitoring [6]. Furthermore, security features like encryption, role-based access control, and anomaly detection safeguard user data and system integrity, addressing key challenges in human–computer interaction systems [7].

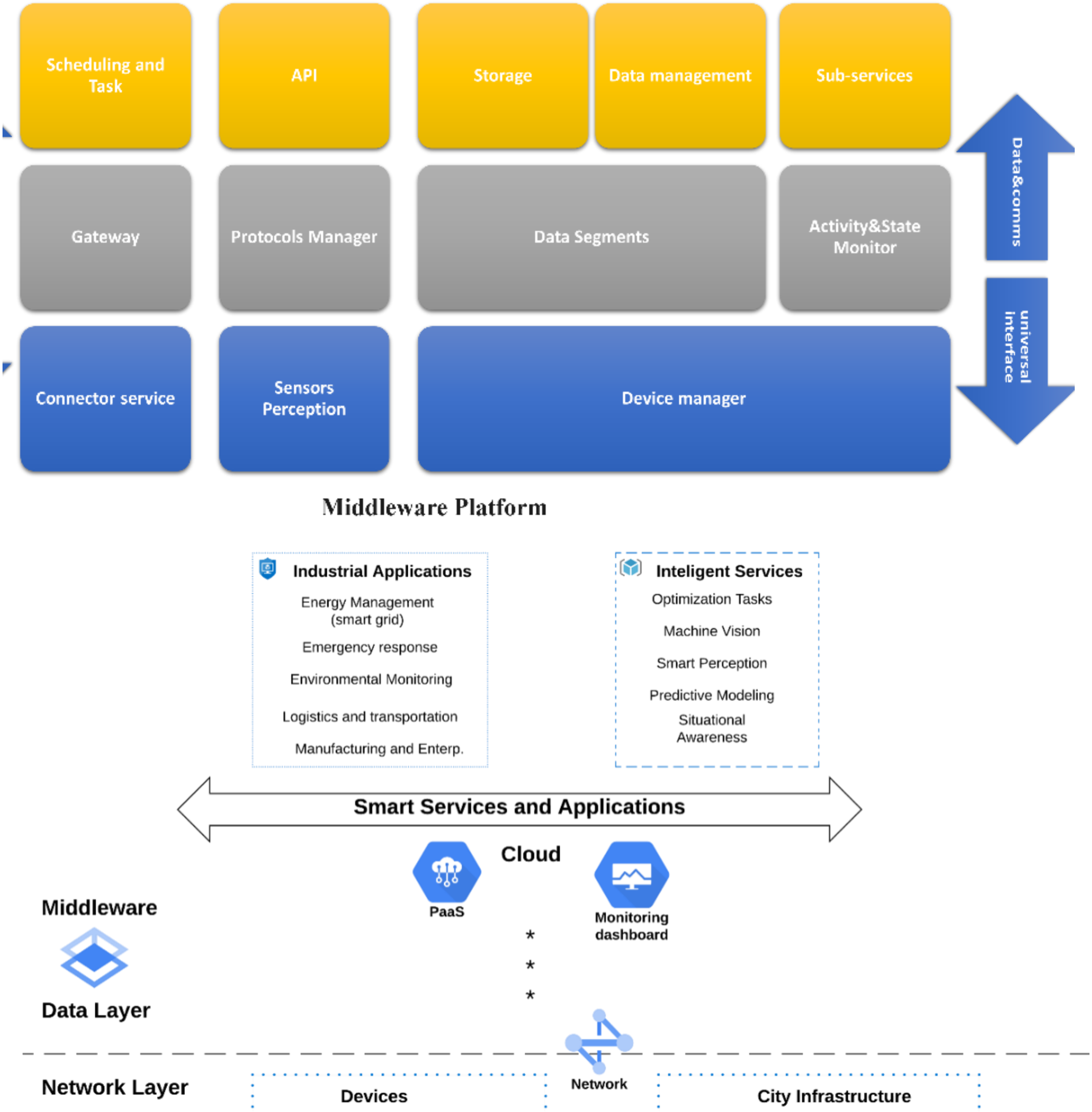

Figure 6 – Middleware IoT system components used for interactive system development

The diagram underscores the middleware's role in supporting hybrid intelligence by integrating human-in-the-loop feedback mechanisms, which refine system performance through iterative user input and interaction [14]. This is particularly relevant in systems where cognitive load or user trust plays a significant role, such as in interactive learning platforms or industrial IoT systems [17]. Overall, the middleware IoT system components provide a robust, flexible, and secure framework for developing interactive systems, enabling real-time, context-aware, and user-friendly solutions across various applications.

Table 3 illustrates the multilayered architecture of an interactive middleware system, highlighting the hierarchical structure and detailing the essential components and subsystems required for IoT systems. The design is divided into three primary layers: the hardware endpoint devices, the central core application management services, and distributed services and applications, each serving distinct but interconnected roles in the middleware framework.

*Table 3 – Interactive middleware system design layers**

Conceptual system layer	Underlying subsystems	Main layer components
Hardware endpoint devices	Sensor and Actuator Router/Network Device Singal Converter Driver and Firmware Kernel Basic notification/indication	Power supply Device memory Hardware Data Communication Protocol Connectivity Low-level code Local address
Primary core application management services	Device Configurator Network Configurator Device Authentication (secure) Communication Optimization Protocols Event manager Events System Adaptation System	(local) Network Data segments Comms Protocols Device (cluster) address Message input/output Control code Listeners
Distributed services and applications	Data Connectors Data en(de)-encryption Input/output transfer Visualization/Graphics Application Communication Virtual Machine Main Server Additional Servers	API connectivity Communication protocol Data packages Database(s) Server-level code OS Network layers

* prepared based on the author's work and public research data

This foundational layer comprises the physical devices and essential hardware components that interact with the external environment. Key elements include sensors and actuators, which are responsible for data collection and execution of commands, and routers or network devices that facilitate connectivity. Supporting components such as signal converters, drivers, firmware, and the kernel enable the hardware to operate efficiently and communicate effectively with higher layers. Additional features like power supply systems, device memory, and communication protocols ensure reliable operation. Low-level functionalities, such as basic notification/indication mechanisms and local address assignments, ensure seamless data transmission and device operation. Together, these elements form the hardware foundation upon which the middleware's higher layers rely.

This intermediate layer acts as the brain of the middleware system, managing devices and networks while ensuring secure and efficient communication. Core components include the device and network configurators, which oversee the setup and optimization of devices and communication networks. Secure communication protocols and device authentication mechanisms provide robust security against unauthorized access. Additionally, the optimization protocols ensure efficient resource allocation, and event management systems enable responsive, context-aware functionalities. This layer also incorporates event systems and adaptation mechanisms to handle dynamic changes in the environment or system configuration. These services are vital for creating scalable and adaptive interactive systems capable of real-time operation.

The topmost layer of the middleware system supports higher-level functionalities, including data handling, application management, and user interface integration. Key components such as data connectors and encryption/decryption mechanisms enable secure and efficient data transfer across the system. This layer also includes input/output transfer modules, visualization, and application communication, providing end-users with actionable insights and a user-friendly interface. Advanced components such as virtual machines, main and additional servers, and API connectivity facilitate scalability and interoperability. Integrating communication protocols, databases, and server-level code ensures seamless interaction between distributed devices and applications. Including network layers and operating systems in this layer highlights its comprehensive role in supporting distributed and scalable IoT ecosystems.

This layered design framework ensures that interactive middleware systems are modular, secure, and adaptive, supporting a wide range of applications and enabling seamless integration of IoT devices into real-world systems.

Discussion. The novel state-of-the-art user interface systems (UIs) can be classified as multilayered and complex systems. Their design and development extend far beyond traditional single-button or keyboard-based interactions. Many user-interaction devices and output mediums, such as mobile phone screens, VR glasses, gamepads, and other controllers, are available and actively used. Their range of usage varies from day-to-day tasks to specific industry or job-role-specific tasks. It should be noted that the range of input-output systems is more comprehensive than the provided examples and goes beyond the listed scenarios (use cases). This research shows that user interactions encompass numerous stages and steps with diverse potential interaction activities.

As detailed in this study, iterative feedback loops are a critical part of human-computer interaction (HCI) systems and one of the user-application core tasks. The HCI is mostly cyclical (iterative) and is centered on tasks and activities initiated by the user (or, in some cases, systems, if such behavior had been programmed in advance or is part of smart autonomy system design) and subsequently processed by the system. The context of use plays a pivotal role in the interaction cycle, offering interaction-system designers valuable insights into the user environment. It helps answer critical design-time questions on when and how the user will utilize the system in real-time within the scope of the intended scenario.

As outlined in this work, the interaction process consists of several steps, components, actors, and systems. Each application constitutes unique interaction and feedback loops. Despite these variations, the implementation of interaction systems can be standardized through system abstractions and algorithmic approaches, such as feedback loop algorithms. This systematic approach enhances the design and usability of interaction systems across diverse applications and environments.

In conclusion, the proposed engineering design framework for smart and intelligent IoT interaction systems provides a structured and adaptable approach to developing user-centric, scalable, and secure solutions for diverse IoT applications. The framework addresses critical challenges in IoT system development, including interoperability, data security, and real-time adaptability, by integrating layered middleware components, advanced user interface design principles, and robust human-computer interaction mechanisms. The detailed breakdown of hardware endpoint devices, core application management services, and distributed services highlights the interdependencies within the system, ensuring seamless operation across diverse environments. Furthermore, including hybrid intelligence through human-in-the-loop mechanisms enhances system responsiveness and user trust. This research contributes to the field by offering a comprehensive blueprint for IoT systems that effectively combine embedded systems, innovative technologies, and interactive functionalities to meet the evolving demands of modern applications, such as healthcare, smart cities, and industrial automation.

References

1. Borchers, J. O. (2000). A pattern approach to interaction design. Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS' 00), 369–378. https://doi.org/10.1145/347642.347795
2. Löwgren, J. (n.d.). Interaction design - brief intro. Interaction Design Foundation. Retrieved November 30, 2024, from https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/interaction-design-brief-intro

3. Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2001). An introduction to 3-D user interface design. Presence, 10(1), 96–108. https://doi.org/10.1162/105474601750182342
4. LaViola, J. J., Jr. (2008). Bringing VR and spatial 3D interaction to the masses through video games. IEEE Computer Graphics and Applications, 28(5), 10–15. https://doi.org/10.1109/MCG.2008.92
5. Zhu, Y., Sun, F., Jia, C., Huang, C., Wang, K., Li, Y., Chou, L., & Mao, Y. (2022). A 3D printing triboelectric sensor for gait analysis and virtual control based on human-computer interaction and the Internet of Things. Sustainability, 14(17), 10875. https://doi.org/10.3390/su141710875
6. Nazari Shirehjini, A. A., & Semsar, A. (2017). Human interaction with IoT-based smart environments. Multimedia Tools and Applications, 76(11), 13343–13365. https://doi.org/10.1007/s11042-016-3697-3
7. Wang, K. (2021). Human-computer interaction design of intelligent vehicle-mounted products based on the internet of things. Mobile Information Systems, 2021, 6795440. https://doi.org/10.1155/2021/6795440
8. Fischer, F., Fleig, A., Klar, M., & Müller, J. (2022). Optimal feedback control for modeling human–computer interaction. ACM Transactions on Computer-Human Interaction, 29(1), Article 1. https://doi.org/10.1145/3524122
9. Ritter, W. (2011). Benefits of subliminal feedback loops in human-computer interaction. Advances in Human-Computer Interaction, 2011, 346492. https://doi.org/10.1155/2011/346492
10. Honeycutt, D. R., Nourani, M., & Ragan, E. D. (2020). Soliciting human-in-the-loop user feedback for interactive machine learning reduces user trust and impressions of model accuracy. Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, 8(1). https://doi.org/10.1609/hcomp.v8i1.7464
11. Palanque, P., Cockburn, A., & Gutwin, C. (2020). A classification of faults covering the human-computer interaction loop. In A. Casimiro, F. Ortmeier, F. Bitsch, & P. Ferreira (Eds.), Computer Safety, Reliability, and Security (pp. 434–448). Springer International Publishing. https://doi.org/10.1007/978-3-030-54549-9_29
12. Nazarenko, V. (2024). Interaction and feedback loops in user interface for digital engineering design. Information Technologies in Economics and Environmental Management, 0, 46–54.
13. Savidis, A., & Stephanidis, C. (2004). Unified user interface design: Designing universally accessible interactions. Interacting with Computers, 16(2), 243–270. https://doi.org/10.1016/j.intcom.2003.12.003
14. Wiethof, C., & Bittner, E. A. (2021). Hybrid intelligence - Combining the human in the loop with the computer in the loop: A systematic literature review. In ICIS 2021 Proceedings.
15. Costabile, M. F., Fogli, D., Mussio, P., & Piccinno, A. (2007). Visual interactive systems for end-user development: A model-based design methodology. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 37(6), 1029–1046. https://doi.org/10.1109/TSMCA.2007.904776
16. Ju, W., & Leifer, L. (2008). The design of implicit interactions: Making interactive systems less obnoxious. Design Issues, 24(3), 72–84. https://doi.org/10.1162/desi.2008.24.3.72
17. Hollender, N., Hofmann, C., Deneke, M., & Schmitz, B. (2010). Integrating cognitive load theory and concepts of human–computer interaction. Computers in Human Behavior, 26(6), 1278–1288. https://doi.org/10.1016/j.chb.2010.05.031
18. Ramírez, J. G. C. (2024). Natural language processing advancements: Breaking barriers in human-computer interaction. Journal of Artificial Intelligence General Science, 3(1), 31–39. https://doi.org/10.60087/jaigs.v3i1.63
19. Nazarenko, V. A., & Ostroushko, B. P. Smart city management system utilizaing micro-services and IoT-based systems. Energy and Automation, (1), 29-38. http://dx.doi.org/10.31548/energiya1(71).2024.029.