TOM: A Development Platform For Wearable Intelligent Assistants

Nuwan Janaka
nuwanj@u.nus.edu
Smart Systems Institute, National University of Singapore
Synteraction Lab
Singapore

Shengdong Zhao*
shengdong.zhao@cityu.edu.hk
Synteraction Lab
School of Creative Media &
Department of Computer Science,
City University of Hong Kong
Hong Kong, China

David Hsu*
dyhsu@comp.nus.edu.sg
School of Computing, National University of Singapore
Smart Systems Institute, National University of Singapore
Singapore

Sherisse Tan Jing Wen
sherisse_tjw@u.nus.edu
School of Computing, National University of Singapore
Singapore

Chun Keat Koh
idmkck@nus.edu.sg
Smart Systems Institute, National University of Singapore
Singapore

ABSTRACT

Advanced wearable digital assistants can significantly enhance task performance, reduce user burden, and provide personalized guidance to improve users' abilities. However, developing these assistants presents several challenges. To address this, we introduce *TOM* (*The Other Me*), a conceptual architecture and open-source software platform (https://github.com/TOM-Platform) that supports the development of wearable intelligent assistants that are contextually aware of both the user and the environment. Collaboratively developed with researchers and developers, *TOM* meets their diverse requirements. *TOM* facilitates the creation of intelligent assistive AR applications for daily activities, supports the recording and analysis of user interactions, and provides assistance for various activities, as demonstrated in our preliminary evaluations.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; **Mobile devices**; **Mixed / augmented reality**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

context-aware system, wearable, AI assistance, smart glasses, HMD, interactions, augmented reality, AR, MR, XR

ACM Reference Format:
Nuwan Janaka, Shengdong Zhao, David Hsu, Sherisse Tan Jing Wen, and Chun Keat Koh. 2024. *TOM*: A Development Platform For Wearable Intelligent Assistants. In *Companion of the 2024 ACM International Joint Conference on Pervasive and Ubiquitous Computing Pervasive and Ubiquitous Computing (UbiComp Companion '24), October 5–9, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3675094.3678382

*Corresponding Authors.

UbiComp Companion '24, October 5–9, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-1058-2/24/10
https://doi.org/10.1145/3675094.3678382

1 INTRODUCTION

With recent advancements in Machine Learning (ML) and Artificial Intelligence (AI) technologies, intelligent digital assistants are becoming an integral part of daily life. These include traditional voice assistants like Siri or Google, and emerging wearable assistants like Humane Ai Pin [1] and Rabbit R1 [24]. Intelligent digital assistants can practically aid users in performing both familiar and new tasks, reduce task load and errors, and enhance task performance [11]. Moreover, these assistants can offer personalization, optimize support for individual needs, and broaden accessibility.

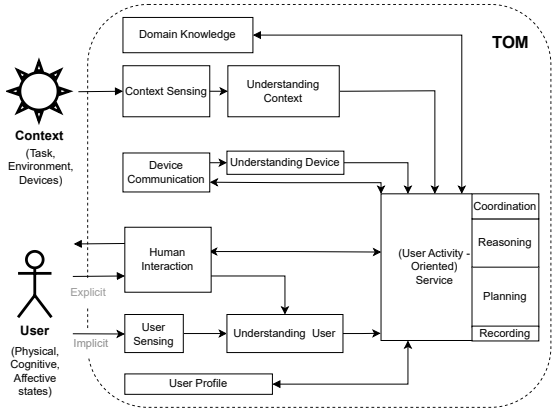


Figure 1: High-level conceptual architecture of *TOM*. Arrows indicate the communication channels, and arrow directions represent the data/interaction flow.

However, developing wearable intelligent assistants presents challenges for stakeholders such as users, developers, and researchers. Despite existing interaction paradigms such as Heads-Up Computing [33] aiming to realize such assistance in daily activities with a focus on users, there is a lack of understanding of the required system capabilities and development guidance. While Augmented and Mixed Reality (AR/MR) assistive systems that enhance user performance have been developed [6, 7, 29], most are tailored to specific tasks (e.g., ARGUS [11] for immersive analytics, Project Aria [15] for data collection) and lack adaptability for various daily activities.

Although the Platform for Situated Intelligence (\psi) [5, 8] enables accelerated research and development in traditional interactive systems, it lacks support for wearable, user-centered applications [33] that facilitate task assistance while minimizing interference by understanding user and context. While the recently introduced SIGMA (Situated Interactive Guidance, Monitoring, and Assistance) [9], which extends \psi, enables a mature system for (linear) procedural task guidance using MR, it lacks user modeling, is catered to specific smart glasses, and does not support general-purpose daily tasks by default. Emerging wearable intelligent assistants such as Rabbit R1 [24], which support specific activities (e.g., booking taxis, querying objects), do not provide easy development or research support (e.g., analyzing/visualizing data) and have limited user interactions and understanding.

To tackle these challenges, we introduce *TOM* (*The Other Me*), a software platform developed by identifying the needs of users, researchers, and developers. *TOM* facilitates the creation and analysis of wearable assistive applications, integrates new devices, enables understanding of context and users, and supports multimodal interactions with AR/MR devices and ML/AI technologies. Through developing several proof-of-concept services (e.g., running coach assistance, querying assistance, and memory assistance), we showcase the utility of *TOM* in supporting different daily activities and highlight the necessary future improvements.

The contributions of this paper are twofold: 1) Identifying essential capabilities for a wearable intelligent assistive system and proposing a conceptual architecture; 2) Creating the *TOM* system platform to facilitate the development of assistive services for diverse activities and demonstrating its utility.

2 *TOM: THE OTHER ME*

2.1 System Capabilities

We have identified necessary system capabilities for users, researchers, and developers based on literature reviews, interviews with AR/MR/AI researchers and developers, and tests of assistive Human-AI interfaces with users.

Just-in-time Assistance for Users. Users should be able to **interact ($\mathbf{C1}_a$)** with the system (i.e., provide input and receive feedback) naturally and optimally to obtain the desired assistance [17, 33]. Such assistance should be delivered just in time to match the user's current needs or proactively when users have limited knowledge of system capabilities [3, 25, 31], with minimal interference in the user's ongoing activities while accommodating the user's cognitive capabilities [4, 21]. To achieve this, the system should **understand the user ($\mathbf{C1}_b$) and context ($\mathbf{C1}_c$)** to provide the most appropriate feedback to support the user's ongoing activities [13, 33]. This understanding aids in modeling the human and the world to minimize awareness mismatch between user expectations and system feedback, and maintaining profiles [27].

Data Recording and Analysis for Researchers. To understand user interactions with such a system and to design optimal interactions, researchers need to **record ($\mathbf{C2}_a$), visualize ($\mathbf{C2}_b$), and analyze ($\mathbf{C2}_c$)** the data and develop models [11, 15, 22]. This involves collecting data to support real-time and retrospective observations, training models to predict optimal feedback, and analyzing their performance, and understanding the underlying reasons for user and system behaviors [11, 15, 23].

Ease of Development for Developers. Considering the variety of activities users may engage in, the system should enable developers to create different assistive features easily. This requires that developers can **integrate new devices ($\mathbf{C3}_a$)** easily (e.g., sensors to understand new contexts or actuators to provide optimal feedback), **deploy new assistance and models ($\mathbf{C3}_b$)** (e.g., to predict optimal feedback), and **access and control current data ($\mathbf{C3}_c$)** (e.g., from existing devices or models).

2.2 Conceptual Architecture

To support the above requirements, we consider three main entities: *user* (i.e., the individual receiving assistance), *context* (i.e., the user's perceptual space and associated tasks), and the *system*, *TOM*, as illustrated in Figure 1, following the high-level context sources [16].

Separating the *user* from the *context* enables us to develop user interaction models [33]. These models sense and understand the user ($\mathbf{C1}_b$, e.g., cognitive states, affective states, physical states [16]) to provide personalized feedback. Thus, *TOM* maintains user profiles to cater to individual preferences and capabilities.

Given that daily activities, such as cooking, typically involve both digital (e.g., viewing a recipe) and physical tasks (e.g., selecting the proper portion), *TOM* offers system-level support to connect the digital world with the physical world by understanding the context ($\mathbf{C1}_c$, e.g., physical environment) and utilizing pervasive augmented reality [16]. This involves a multi-modal and multifaceted understanding of the environment (e.g., understanding the ongoing activities, associated objects, and relationships) as well as understanding the devices that facilitate interactions (e.g., device resource availability).

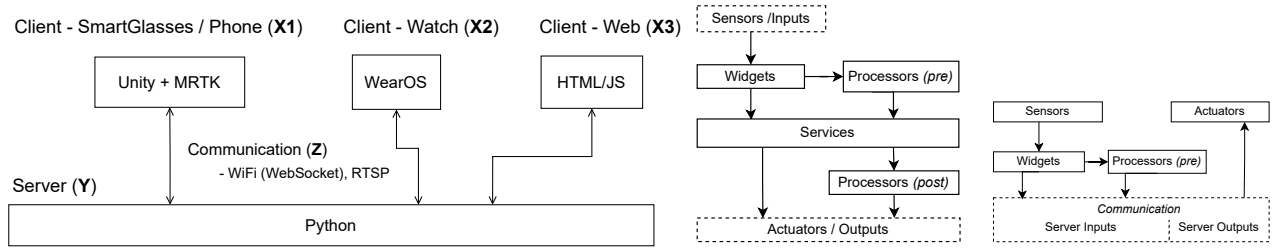
In terms of input, *TOM* supports the user's explicit multi-modal inputs ($\mathbf{C1}_a$, such as voice and gesture) as well as implicit inputs ($\mathbf{C1}_a$, like gaze and physiological data), in addition to processing multi-modal context information.

After understanding the context (e.g., ongoing activity) and user (e.g., intention), *TOM* activates a context-aware service, employing domain knowledge to generate real-time proactive suggestions through reasoning and planning. These suggestions are conveyed to users as multi-sensory feedback ($\mathbf{C1}_a$), tailored to their cognitive capacity, including visual, auditory, and/or haptic modalities. The feedback is dynamically updated based on the user's actions; for instance, if the user does not follow a given suggestion, *TOM* formulates the next appropriate suggestion, considering the user's current status and context, facilitating a closed-loop control system.

When multiple *TOM* users are involved in a collaborative activity (e.g., group discussion on an artifact), each *TOM* system enables multi-agent coordination to complete the collaborative activity optimally.

3 SYSTEM ARCHITECTURE

We develop the following system/physical architecture based on the requirements and envisioned use cases.



(a) System (physical) architecture. (b) Server architecture. (c) Client architecture.

Figure 2: System architecture of *TOM*. Arrow directions represent the data flow. (a) Client-server architecture with multiple clients is used considering the limited computational resources of wearable devices [11] (b) Server architecture with multiple layers to support the separation of concerns, distributed communication, extendability, and resource discovery [13] (c) Client architecture with multiple layers.

3.1 Devices and Technologies

Following a user-centric approach [33], *TOM* uses wearable devices that align with human input-output channels (e.g., eyes, hands), such as Optical See-Through Head-Mounted Displays (OHMD, Augmented Reality Smart Glasses, e.g., HoloLens2, Nreal Light) and ring-mouses, to support multi-modal interactions ($\mathbf{C1}_a$). It also includes everyday wearable devices like smartwatches (e.g., Samsung Galaxy/WearOS, Fitbit) to understand users, smartphones (e.g., Android) to provide familiar interactions, and web browsers for visualizations.

To understand users, devices, and the environment, and to facilitate reasoning, planning, and coordination, *TOM* employs AI [26] technologies. These include scene understanding, speech recognition, object recognition/tracking, natural language processing, and large language/multimodal models (LLM, LMM). For user feedback, *TOM* utilizes AR/MR technologies (e.g., OHMD). *TOM* uses databases (e.g., PostgreSQL, Milvus) for data recording ($\mathbf{C2}_a$), training ($\mathbf{C3}_b$), and visualization ($\mathbf{C2}_b$). Communication between devices and with external APIs (e.g., ChatGPT) is handled using data communication protocols (e.g., WebSocket, WebRTC, REST API) and wireless mediums (e.g., WiFi, BLE).

3.2 Client-Server Architecture

Given the limited computational resources of wearable devices [11], *TOM* is implemented as a client-server architecture, as illustrated in Figure 2. The server hosts services for processing and orchestrating data, supports ML/AI inferences, and provides real-time feedback to clients. Clients, such as OHMDs and smartwatches, send sensor data to the server and display feedback to the user. This separation also allows *TOM* to be device-agnostic, supporting various OHMDs through the same server ($\mathbf{C3}_a$).

Server Architecture. Designed for flexibility and simplicity, the server acts as a one-stop platform for deploying various services optimized for different activities and switching between them as needed. Adapting the architecture of the Context Toolkit [13], the *TOM* server (Figure 2b) is implemented with independent components under three distinct layers: Widgets (i.e., components that listen for sensors and receive input data), Processors (i.e., stateless components that process and transform input data from Widgets or output data from Services), and Services ($\mathbf{C3}_b$, i.e., stateful components that process data from Widgets and/or Processors to produce desired outcomes)[1]. These layers are interconnected with Sensors/Inputs and Actuators/Outputs linked to Clients. This setup supports the separation of concerns, distributed communication, context storage, and resource discovery. A specialized service, the Context Service, determines the most suitable Service based on current input data (e.g., through explicit user interactions or automatically determined by context data), switching services to support ongoing user activities.

Client Architecture. As shown in Figure 2c, Clients mirror the Server's architecture. However, instead of Services, they interact with the Server to stream sensor data and receive real-time feedback for actuators. Time-critical processing can also be implemented in clients (i.e., on-device) to overcome potential latency issues between the client and server.

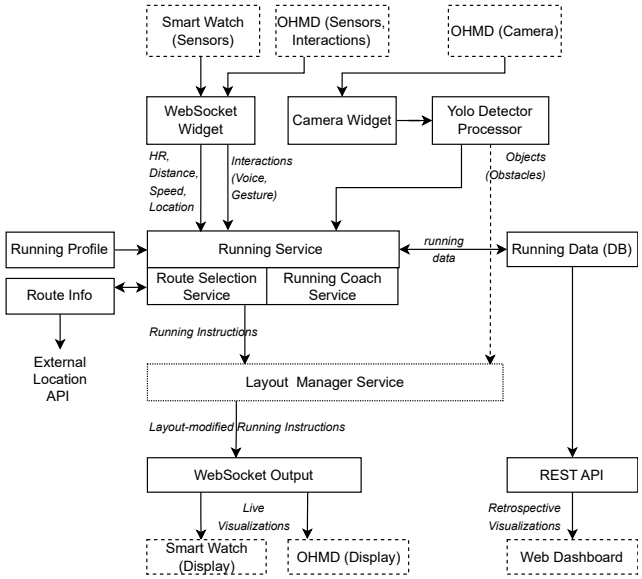
Data Flow and Communication. Data or messages, tagged with source and time, are transferred between different layers (e.g., Input -> Widget -> Processor -> Service -> Output, Figure 2b, Appendix A-Figure5) via message channels controlled by configuration files ($\mathbf{C3}_c$, e.g., Figure3b). This arrangement allows for the reuse of various components across multiple Services and supports distributed communication, thus reducing development efforts. Moreover, each component can store the data it handles in a local database for post-analysis ($\mathbf{C2}_c$, e.g., visualization, aggregation) or ML model training ($\mathbf{C3}_b$). The WebSocket protocol is employed for real-time bidirectional communication between Clients and the Server. REST APIs are used for communication with external APIs, while WebRTC and the Real-Time Streaming Protocol (RTSP) facilitate the streaming of real-time video data.

3.3 Implementation

The Server (Figure 2a-Y) is implemented in Python (3.11), chosen for its extensive user base and support of numerous ML/AI libraries with multiprocessing capabilities. The built-in library, SQLAlchemy, also supports Data Storage.

The OHMD or mobile phone clients (Figure 2a-X1) are developed using Unity3D (2021.3) and MRTK[2] (2.8) to provide AR/MR content. This setup accommodates various devices (e.g., HoloLens2

[1] *TOM* does not incorporate Interpreters and Aggregators, unlike the Context Toolkit; their functions are managed by either Processors or Services in *TOM*, minimizing stateful components to enhance testability.
[2] Mixed Reality Toolkit: https://github.com/Microsoft/MixedRealityToolkit-Unity



(a)

```
input:
  - name: "camera"
    entrypoint: "camera_widget.CameraWidget.start"
    exitpoint: ""
    next:
      - "processing:yolov8"
  - name: "websocket"
    entrypoint: "websocket_widget.WebsocketWidget.start"
    exitpoint: ""
    next:
      - "service:running"
processing:
  - name: "yolov8"
    entrypoint: "Yolov8.detector.Yolov8Detector.run"
    exitpoint: ""
    next:
      - "service:running"
service:
  - name: "running"
    entrypoint: "running_service.running_service.RunningService.run"
    exitpoint: ""
    next:
      - "output:websocket"
output:
  - name: "websocket"
    entrypoint: "websocket_output.WebsocketOutput.send"
    exitpoint: ""
```

(b)

Figure 3: Running assistance implemented in *TOM*. (a) System components that enable the running assistance, C3$_b$. Dashed-line boxes indicate implemented Client components, solid lines represent implemented Server components, and dotted lines denote Server components under development. (b) The configuration file that controls the data flow, C3$_c$. Data is received in one or more components in the Input Layer (e.g., 'camera' component) and is sent to the next component as specified in the *next* key (e.g., 'yolov8' component in the Processing Layer). This process occurs similarly for all components regardless of the layer, with the *entry point* dictating the method in each component that receives the data from the previous component. The *exit point* then dictates the method for each component, which is called when they should be stopped (e.g., when the context switch indicates the component is no longer required).

Table 1: Technologies used within the current *TOM*. For the latest supported technologies, refer to the TOM-Platform

Capability	Type	Server	Client			
		Python Server (Laptop)	**HoloLens 2**	**XReal Light**	**Web Client**	**Android Phone**
Context Understanding	Visual	GoogleCloudVision, Yolov8	-	-	-	MLKit
	Audio	MediaPipe	-	-	-	-
	Spatial	GoogleCloudVision	SLAM (inbuilt)	SLAM (inbuilt)	-	-
User Understanding	Physiological	Fitbit API	-	-	-	WearOS Watch
Interactions	Visual	Bing Image Search API	-	-	Graphs, Charts	-
	Audio	Speech-to-Text (Whisper), Text-to-Speech	Speech-to-Text (Azura), Text-to-Speech	Text-to-Speech	-	Speech-to-Text (Android), Text-to-Speech
	Other	-	Hand Tracking (inbuilt), Gaze Tracking (inbuilt)	Hand Tracking (inbuilt)	Visualizations	Touch
Communication	APIs	REST	-	-	-	-
	Data Stream	Websocket	Websocket	Websocket, TCPSocket	Websocket, TCPSocket	TCPSocket
	Video Stream	WebRTC, RTSP	-	WebRTC	WebRTC	WebRTC
Assistance	Queries, Memory	ChatGPT, Claude, Local LLM (TinyLlama) Encoding (Clip, ImageBind) Vector Database (Milvus)	-	-	-	-

with UWP, Nreal with Android) and enables mixed reality capabilities with OpenXR support. Other clients, such as the Smartwatch (Figure 2a-X2), designed to sense the user, are implemented using WearOS due to its widespread use. Additionally, a web client (Figure 2a-X3) is used for visualization, utilizing HTML/VueJS and NodeJS.

Table 1 provides an overview of the technologies that support the *TOM*'s **current** capabilities, and Table 2 illustrates the various client data supporting the system's capabilities. These data sources include Ring Mouse Controllers connected to OHMDs, Gesture and Gaze Detection through OHMDs, and User Input via the Touch Screen on Android Phones.

For additional details, please refer to https://github.com/TOM-Platform and Appendix A.

3.4 Preliminary Use Cases

We invited three teams to implement proof-of-concept services of their choice to support daily activities as case studies [20] to understand how developers, users, and researchers benefit from the *TOM*. The details and feedback are shown in Figure 4, which

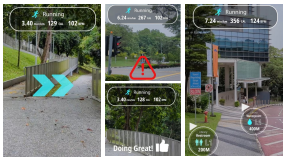
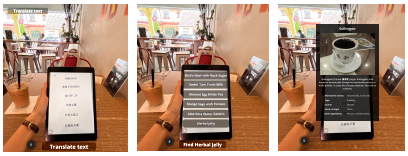
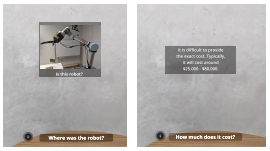
	Running Assistance	**Translation and Querying Assistance**	**Memory Assistance**
Goal	Provide real-time coaching during daily running and supports training for competitions.	Answer user queries in real-time during activities such as shopping, dining, and learning.	Assist users in recalling details such as forgotten locations, names, and items.
Assistance	• Training guidance (speed, directions ...) • Proactive feedback on potential dangers and motivational encouragements • Details the locations of water points for hydration • Summary, Retrospection 	• Translation support for multiple languages to facilitate communication. • Querying information on both physical objects and digital data. • Displays images relevant to user queries to enhance understanding. 	• Remembering and recalling both visual and auditory information. • Facilitates querying of detailed information 
Team	1 Developer, 2 Researcher	2 Developers, 2 Researcher	1 Developer, 1 Researcher
System Physical Architecture	**Clients** • HoloLens2 (Unity3D, MRTK) • Samsung Watch 5 (WearOS) • Web Dashboard (Vue JS) **Server** • Laptop (Python)	**Clients** • XReal Light (Unity3D, MRTK) • HoloLens2 (Unity3D, MRTK) **Servers** • Laptop (Python) • Samsung Galaxy S23 (Unity3D)	**Client** • HoloLens2 (Unity3D, MRTK) **Server** • Laptop (Python)
APIs, Libraries	**Context Understanding** • [Server] GoogleCloudVision, YoloV8 • [Client] Camera feed, Location **User Understanding** • [Client] Physiological (Heart Rate...), Physical info (speed ...) **User Interactions** • [Client] Visual + Auditory + Haptic feedback • [Client] Speech Input, Gesture Input **Communication** • [External] REST, PostgreSQL	**Context Understanding** • [Server] GoogleCloudVision, YoloV8, Claude, MediaPipe • [Client] Camera feed, Head motion, ML Kit **User Interactions** • [Client] Visual + Auditory feedback • [Client] Speech Input, Gesture Input, Gaze Input **Communication** • [External] ChatGPT, Bing Image	**Context Understanding** • [Server] OpenAI CLIP, Meta ImageBind, OpenAI Whisper, MediaPipe • [Client] Camera feed, Audio **User Interactions** • [Client] Visual + Auditory feedback • [Client] Speech Input, Gesture Input **Communication** • [External] Local LLM (TinyLlama), Claude, Milvus (Vector DB) for RAG (Retrieval-Augmented Generation)
***User* Experience**	**Task:** Run to a designated place for 1 km (N=2, tech-savvy casual runners) **Positive** • Effective than Smart Watches • Easier to find required information **Negative** • [Device] Heavy, Visual feedback is unstable due to head movement, Visibility is impaired in sunlight. • [Technical] Voice command misrecognition in noisy environments **Enhancements** • Customization of the UI elements according to user preferences and environmental conditions.	**Task:** Query items during 15-min shopping in a supermarket (N=3, tech-savvy students) **Positive** • Highly intuitive and useful. • Facilitates easy querying of additional information. **Negative** • [Device] Less fashionable • [Technical] UI anchoring issues, Difficulty in detecting small text, Retrieval of incorrect images (from Bing Image), Response time is slow (2-8 sec), Occasional hallucinations (from LLM) **Enhancements** • Customization to provide implicit inputs that accelerate information retrieval.	**Task:** Use the system during an exhibition (30-min) and explain the seen exhibits (N=2, tech-savvy students) **Positive** • Highly useful. • Simplifies the process of remembering and querying information. **Negative** • [Device] Cumbersome • [Technical] Tends to retrieve information similar but unrelated to the intended query, Challenges in quickly filtering and finding the correct stored information • [Privacy, Ethical] Continuously listens to user's voice, Takes photos without user and bystander consent. **Enhancements** • Enable subtle interactions that allow users to selectively record specific events.
***Developer* Feedback**	**Positive** • Significantly reduces time required to link multiple devices. • Offers easy customization of the logic and tweaking of assistance based on feedback from users and researchers. • Allows easy control of data flow through user configurations. **Enhancements** • Enable additional built-in support for long-term retrospective data visualizations.	**Positive** • Saves time through accessible user inputs (gestures, speech, gaze, manual input). • Includes many built-in libraries and API support, enhancing functionality. • Capable of executing multiple assistance services in parallel. • Offers flexibility to adjust code for operation on smartphones in addition to smart glasses. **Enhancements** • Enable built-in support for content anchoring.	**Positive** • Provides easy access to encoding APIs and locally running large language models (LLMs). • Significantly reduces time required for prototyping various services due to simplified management of data flows. • Easy to debug **Enhancement** • Reduces the necessity for background knowledge in Unity to customize user interfaces and interactions. • Enable support for visual programming for easier customization of data flows.
***Researcher* Feedback**	• Facilitates easy recording of multi-modal data from various devices, enabling the training of personalized models for future coaching applications. • Simplifies the visualization of data.	• Recorded data facilitate detailed post-interview analysis. • Enables easier prototyping of different interaction methods and aids in identifying suitable parameters for individual interactions.	• Allows for the implementation and testing of various algorithms to evaluate their pros and cons. • Simplifies the prototyping of different assistance services.

Figure 4: Details of the three preliminary case studies, including team, system, task, and feedback from users and team.

Table 2: Data from currently supported Clients. For the latest supported devices/capabilities, refer to the TOM-Platform.

Data	Type	Client				
		HoloLens2	**XReal Light**	**WearOS Watch**	**Web Client**	**Android Phone**
Context Understanding	Visual	Video	Video	-	-	Video
	Auditory	Audio	Audio	-	-	Audio
	Spatial	WorldMesh	WorldMesh	Location (GPS)	-	Location (GPS)
User Understanding	Physiological	-	-	Heart Rate	-	-
	Physical	-	-	Speed, Calories	-	-
Interactions (Output)	Visual	Text, Image, Video, 3D Object	Text, Image, Video, 3D Object	Text, Image, 2D Object	Text, Images, Video	Text, Image, Video, 3D Object
	Auditory	Audio, Text	Audio, Text	Audio	Audio	Audio, Text
Interactions (Input)	Voice	Audio (Speech)	Audio (Speech)	-	-	Audio (Speech)
	Gesture	Hand, Finger	Hand, Finger	-	-	-
	Gaze	3D Gaze, Gaze Collision	-	-	-	-
	Controller	2D Press	3D Press	-	-	2D Touch

indicates that *TOM* is able to meet their needs (Sec 2.1) and highlight areas for improvement.

Additionally, we conducted a preliminary technical evaluation (N=50 requests/calls) of major system components/APIs: communication (websocket) latency between client and server: 0.091 ± 0.012s, video streaming (HoloLens2): 0.90 ± 0.15s, voice interaction (HoloLens2): 2.45 ± 0.16s (average gaze and gesture points are sent every 1s), YoloV8 object detection: 0.023 ± 0.005s, Google Text/Object Detection: 1.04 ± 0.15s, GPT-3.5: 0.67 ± 0.24s, TinyLlama (local LLM): 1.30 ± 0.07s, CLIP embedding: 0.69 ± 0.24s. Additionally, WearOS sends the average data every 3s.

4 LIMITATIONS AND FUTURE WORK

In addition to technical and interaction challenges (Figure 4), situational impairments, such as decreased feedback accuracy (e.g., hallucinations) in dynamic environments, highlight the need for seamless input modality transitions [19] and more transparent AI explanations [10, 14, 30] to mitigate user mistrust. *TOM* also faces limitations in automatic service switching to optimize suitable assistance and could benefit from integrating Large Action Models (LAM) [24] to enhance interaction efficiency and understanding of user actions. Additionally, *TOM* requires improved visualization techniques for better long-term behavior analysis [11, 18] and advanced modeling [22, 23] to comprehend users' cognitive states and activity correlations. Moreover, developing such systems introduces significant privacy, security, and ethical concerns [2, 12, 28], particularly in real-world deployment affecting users and bystanders. These issues necessitate further advancements in on-device computing [32] and robust handling of sensitive data.

5 CONCLUSION

We have presented the anticipated capabilities of developing a wearable intelligent assistive system and introduced *TOM*, an architecture and open-source implementation (https://github.com/TOM-Platform) that enables researchers and developers to create and analyze assistive applications for supporting daily activities. We welcome contributions from the community to expand its supported devices and usage scenarios. We envision that *TOM* will serve as a software platform for researchers and developers to develop innovative, intelligent assistance in various tasks, facilitating human-computer, human-AI, and human-robot interactions. Our future plans include extending *TOM*'s capabilities to enable remote robot interactions, where humans can share information (e.g., intentions) with a remote robot to execute tasks.

ACKNOWLEDGMENTS

We would like to express our gratitude to the volunteers who participated in our studies and to the interns of *TOM* project, including Teo Yun Yew Jarrett, Taufiq Bin Abdul Rahman, and Lu Shaoqin, who developed certain system components. We also wish to thank the reviewers for their valuable time and insightful comments, which helped to improve this paper.

This research is supported by the National Research Foundation, Singapore, under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-016). The CityU Start-up Grant 9610677 also provides partial support. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

REFERENCES

[1] 2024. Ai Pin Overview. https://hu.ma.ne/aipin

[2] Fouad Alallah, Ali Neshati, Yumiko Sakamoto, Khalad Hasan, Edward Lank, Andrea Bunt, and Pourang Irani. 2018. Performer vs. observer: whose comfort level should we consider when examining the social acceptability of input modalities for head-worn display?. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3281505.3281541

[3] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300233

[4] Christoph Anderson, Isabel Hübener, Ann-Kathrin Seipp, Sandra Ohly, Klaus David, and Veljko Pejovic. 2018. A Survey of Attention Management Systems in Ubiquitous Computing Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (July 2018), 1–27. https://doi.org/10.1145/3214261

[5] Sean Andrist, Dan Bohus, Ashley Feniello, and Nick Saw. 2022. Developing Mixed Reality Applications with Platform for Situated Intelligence. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 48–50. https://doi.org/10.1109/VRW55335.2022.00018

[6] E. Z. Barsom, M. Graafland, and M. P. Schijven. 2016. Systematic review on the effectiveness of augmented reality applications in medical training. *Surgical Endoscopy* 30, 10 (Oct. 2016), 4174–4183. https://doi.org/10.1007/s00464-016-4800-6

[7] Mark Billinghurst, Adrian Clark, and Gun Lee. 2015. A Survey of Augmented Reality. *Foundations and Trends® in Human–Computer Interaction* 8, 2-3 (March 2015), 73–272. https://doi.org/10.1561/1100000049

[8] Dan Bohus, Sean Andrist, Ashley Feniello, Nick Saw, Mihai Jalobeanu, Patrick Sweeney, Anne Loomis Thompson, and Eric Horvitz. 2021. Platform for Situated Intelligence. https://doi.org/10.48550/arXiv.2103.15975 arXiv:2103.15975 [cs].

[9] Dan Bohus, Sean Andrist, Nick Saw, Ann Paradiso, Ishani Chakraborty, and Mahdi Rad. 2024. SIGMA: An Open-Source Interactive System for Mixed-Reality Task Assistance Research. https://arxiv.org/abs/2405.13035v1

[10] John M. Carroll. 2022. Why should humans trust AI? *Interactions* 29, 4 (June 2022), 73–77. https://doi.org/10.1145/3538392

[11] Sonia Castelo, Joao Rulff, Erin McGowan, Bea Steers, Guande Wu, Shaoyu Chen, Iran Roman, Roque Lopez, Ethan Brewer, Chen Zhao, Jing Qian, Kyunghyun Cho, He He, Qi Sun, Huy Vo, Juan Bello, Michael Krone, and Claudio Silva. 2023. ARGUS: Visualization of AI-Assisted Task Guidance in AR. https://doi.org/10.48550/arXiv.2308.06246

[12] Tamara Denning, Zakariya Dehlawi, and Tadayoshi Kohno. 2014. In situ with bystanders of augmented reality glasses: perspectives on recording and privacy-mediating technologies. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, Toronto, Ontario, Canada, 2377–2386. https://doi.org/10.1145/2556288.2557352

[13] Anind K. Dey, Gregory D. Abowd, and Daniel Salber. 2001. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human–Computer Interaction* 16, 2-4 (Dec. 2001), 97–166. https://doi.org/10.1207/S15327051HCI16234_02

[14] Malin Eiband, Hanna Schneider, Mark Bilandzic, Julian Fazekas-Con, Mareike Haug, and Heinrich Hussmann. 2018. Bringing Transparency Design into Practice. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. Association for Computing Machinery, New York, NY, USA, 211–223. https://doi.org/10.1145/3172944.3172961

[15] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, Cheng Peng, Chris Sweeney, Cole Wilson, Dan Barnes, Daniel DeTone, David Caruso, Derek Valleroy, Dinesh Ginjupalli, Duncan Frost, Edward Miller, Elias Mueggler, Evgeniy Oleinik, Fan Zhang, Guruprasad Somasundaram, Gustavo Solaira, Harry Lanaras, Henry Howard-Jenkins, Huixuan Tang, Hyo Jin Kim, Jaime Rivera, Ji Luo, Jing Dong, Julian Straub, Kevin Bailey, Kevin Eckenhoff, Lingni Ma, Luis Pesqueira, Mark Schwesinger, Maurizio Monge, Nan Yang, Nick Charron, Nikhil Raina, Omkar Parkhi, Peter Borschowa, Pierre Moulon, Prince Gupta, Raul Mur-Artal, Robbie Pennington, Sachin Kulkarni, Sagar Miglani, Santosh Gondi, Saransh Solanki, Sean Diener, Shangyi Cheng, Simon Green, Steve Saarinen, Suvam Patra, Tassos Mourikis, Thomas Whelan, Tripti Singh, Vasileios Balntas, Vijay Baiyya, Wilson Dreewes, Xiaqing Pan, Yang Lou, Yipu Zhao, Yusuf Mansour, Yuyang Zou, Zhaoyang Lv, Zijian Wang, Mingfei Yan, Carl Ren, Renzo De Nardi, and Richard Newcombe. 2023. Project Aria: A New Tool for Egocentric Multi-Modal AI Research. https://doi.org/10.48550/arXiv.2308.13561

[16] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2017. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (June 2017), 1706–1724. https://doi.org/10.1109/TVCG.2016.2543720

[17] Kasper Hornbæk and Antti Oulasvirta. 2017. What Is Interaction?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver Colorado USA, 5040–5052. https://doi.org/10.1145/3025453.3025765

[18] Nuwan Janaka, Runze Cai, Ashwin Ram, Lin Zhu, Shengdong Zhao, and Yong Kai Qi. 2024. PilotAR: Streamlining Pilot Studies with OHMDs from Concept to Insight. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (Sept. 2024). https://doi.org/10.1145/3678576

[19] Nuwan Janaka, Jie Gao, Lin Zhu, Shengdong Zhao, Lan Lyu, Peisen Xu, Maximilian Nabokow, Silang Wang, and Yanch Ong. 2023. GlassMessaging: Towards Ubiquitous Messaging Using OHMDs. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 3 (Sept. 2023), 100:1–100:32. https://doi.org/10.1145/3610931

[20] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3173574.3173610

[21] D. Scott McCrickard and C. M. Chewar. 2003. Attuning notification design to user goals and attention costs. *Commun. ACM* 46, 3 (March 2003), 67. https://doi.org/10.1145/636772.636800

[22] Roderick Murray-Smith, Antti Oulasvirta, Andrew Howes, Jörg Müller, Aleksi Ikkala, Miroslav Bachinski, Arthur Fleig, Florian Fischer, and Markus Klar. 2022. What simulation can do for HCI research. *Interactions* 29, 6 (Nov. 2022), 48–53. https://doi.org/10.1145/3564038

[23] Antti Oulasvirta, Jussi P. P. Jokinen, and Andrew Howes. 2022. Computational Rationality as a Theory of Interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3491102.3517739

[24] rabbit research team. 2023. Learning human actions on computer applications. https://rabbit.tech/research

[25] B. J. Rhodes and P. Maes. 2000. Just-in-time information retrieval agents. *IBM Systems Journal* 39, 3.4 (2000), 685–704. https://doi.org/10.1147/sj.393.0685

[26] Christine Rzepka and Benedikt Berger. 2018. User Interaction with AI-enabled Systems: A Systematic Review of IS Research. *ICIS 2018 Proceedings* (Dec. 2018). https://aisel.aisnet.org/icis2018/general/Presentations/7

[27] Albrecht Schmidt. 2014. *Context-Aware Computing*. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/context-aware-computing-context-awareness-context-aware-user-interfaces-and-implicit-interaction

[28] Mel Slater, Cristina Gonzalez-Liencres, Patrick Haggard, Charlotte Vinkers, Rebecca Gregory-Clarke, Steve Jelley, Zillah Watson, Graham Breen, Raz Schwarz, William Steptoe, Dalila Szostak, Shivashankar Halan, Deborah Fox, and Jeremy Silver. 2020. The Ethics of Realism in Virtual and Augmented Reality. *Frontiers in Virtual Reality* 1 (2020). https://doi.org/10.3389/frvir.2020.00001

[29] X. Wang, S. K. Ong, and A. Y. C. Nee. 2016. A comprehensive survey of augmented reality assembly research. *Advances in Manufacturing* 4, 1 (March 2016), 1–22. https://doi.org/10.1007/s40436-015-0131-4

[30] Xuhai Xu, Anna Yu, Tanya R. Jonker, Kashyap Todi, Feiyu Lu, Xun Qian, João Marcelo Evangelista Belo, Tianyi Wang, Michelle Li, Aran Mun, Te-Yen Wu, Junxiao Shen, Ting Zhang, Narine Kokhlikyan, Fulton Wang, Paul Sorenson, Sophie Kim, and Hrvoje Benko. 2023. XAIR: A Framework of Explainable AI in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–30. https://doi.org/10.1145/3544548.3581500

[31] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376301

[32] Jiale Zhang, Bing Chen, Yanchao Zhao, Xiang Cheng, and Feng Hu. 2018. Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues. *IEEE Access* 6 (2018), 18209–18237. https://doi.org/10.1109/ACCESS.2018.2820162

[33] Shengdong Zhao, Felicia Tan, and Katherine Fennedy. 2023. Heads-Up Computing Moving Beyond the Device-Centered Paradigm. *Commun. ACM* 66, 9 (Aug. 2023), 56–63. https://doi.org/10.1145/3571722

A DETAILED SYSTEM ARCHITECTURE

Figure 5 depicts the high-level components of the system architecture. The implemented Server currently supports three distinct types of Services: running assistance, providing speed/distance training with directional support; learning assistance, enabling inquiries about objects in the frame selected through gestures and/or gaze; and translation assistance, which facilitates text translation within the frame.

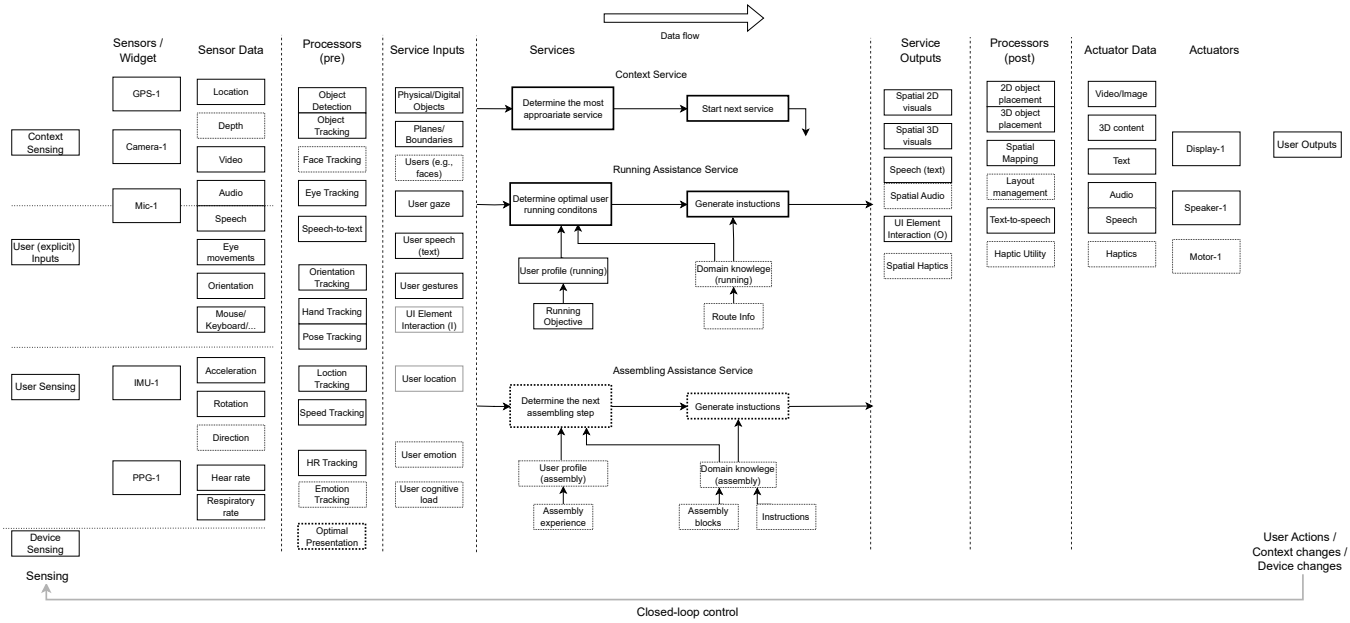


Figure 5: A snapshot of the system architecture and the data flow between high-level components. Solid-lined boxes indicate implemented components, and dotted lines represent components in development.