HUMAN-CENTRIC AI-ENABLED EXTENDED REALITY APPLICATIONS FOR THE INDUSTRY 5.0 ERA

D3.3 – PERSONALISED XR CONTENT DEVELOPMENT AND ADAPTATION V1

Lead Beneficiary	UNP
Work Package Ref.	WP3 – Human Centric Digital Twins and XR Personalisation
Task Ref.	T3.4 – Personalised Content Development T3.5 – XR Content Adaptation and Personalisation Mechanisms
Deliverable Title	D3.3 – Personalised XR Content Development and Adaptation v1
Due Date	2025-04-30
Delivered Date	2025-07-17
Revision Number	3.0
Dissemination Level	Public (PU)
Type	Report (R)
Document Status	Release
Review Status	Internally Reviewed and/or Quality Assurance Reviewed
Document Acceptance	WP Leader Accepted and/or Coordinator Accepted
EC Project Officer	Ms. Marta PALAU FRANCO

HORIZON-CL4-2023 Research and Innovation Action

This project has received funding from the European Union's Horizon Europe Research and Innovation Programme under grant agreement no 101135209.

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

DISCLAIMER

COPYRIGHT MESSAGE

CONTRIBUTING PARTNERS

Partner Acronym	Role[1]	Name Surname[2]
HOLO	Contributor	
IML	Contributor	
SUPSI	Contributor	
UNP	Lead Beneficiary	
GFT	Internal Reviewer	
UPRC	Quality Assurance	

REVISION HISTORY

Version	Date	Partner(s)	Description
0.1	2025-03-30	UNP	ToC Version
0.2	2025-05-27	UNP	Updated ToC Version
0.3	2025-06-27	UNP	Updated Chapter 2
0.4	2025-07-08	UNP	Updated Chapter 3
0.5	2025-07-15	HOLO IML SUPSI	HOLO, IML and SUPSI contributions to the document
0.6	2025-07-16	UNP	Final Contributions from UNP Document formatting
1.0	2025-07-16	UNP	1st Version
1.1	2025-07-16	GFT	Version for Peer Reviews
2.0	2025-07-17	UPRC	Version for Quality Assurance
3.0	2025-07-17	GFT	Version for Submission

[1] Lead Beneficiary, Contributor, Internal Reviewer, Quality Assurance
[2] Can be left void

LIST OF ABBREVIATIONS

Acronym	Definition
AI	Artificial Intelligence
AR	Augmented Reality
AAS	Asset Administration Shell
AUIs	Adaptive User Interfaces
DTLS	Datagram Transport Layer Security
GDPR	General Data Protection Regulation
HCI	Human Computer Interaction
HDTs	Human-Centric Digital Twins
HMIs	Human Machine Interfaces
I5.0	Industry 5.0
ID	Identification
IdP	Identity Provider
iOS	iPhone Operating System
JWT	JSON Web Token
MR	Mixed Reality
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
QR	Quick Response
RBAC	Role-Based Access Control
SAS	Stress Analog Scale
SDK	Software Development Kit
SP	Service Provider
SRTP	Secure Real-Time Transport Protocol
SSO	Single Sign-On
SSQ	Simulator Sickness Questionnaire
TLX	Task Load Index
TPAT	Training Programs Authoring Tool
UIs	User Interfaces
UX	User Experience
UXQ	User Experience Questionnaire
VR	Virtual Reality
XR	Extended Reality
XRTP	XR Training Plugin
WebRTC	Web Real-Time Communication

Executive Summary

Deliverable D3.3, titled "Personalised XR Content Development and Adaptation v1" is developed under WP3 – Human Centric Digital Twins and XR Personalisation, which objectives are:

- Modelling and specifying worker digital images.
- Collecting and analysing personal data for digital twin creation and maintenance.
- Designing and implementing digital twins that accurately represent individual workers and their unique contexts.
- Developing personalized XR content based on worker characteristics, skills, and preferences; and Creating mechanisms for adapting and personalizing XR content in real- time, considering various factors such as worker skills, preferences, cognitive load, and emotional state.

The main purpose of this deliverable is to enable human-centric, adaptive XR experiences by leveraging Human-Centric Digital Twins (HDTs), real-time data collection, and AI-driven personalization mechanisms. The deliverable outlines the architectural foundations of the Worker @ XR5.0 framework, including the integration of XR and wearable technologies for real-time monitoring and interaction. A unified worker model is proposed, encompassing both static and dynamic dimensions to support tailored training experiences.

Key components such as the XR Training Plugin, Hololight, Clawdite Platform, Wearable App, and Asset Administration Shell (AAS) are described, with emphasis on interoperability and secure data exchange via an API gateway. This document also introduces personalization algorithms that adapt training content based on user behaviour and preferences, enhancing engagement and learning outcomes.

A significant portion of the work is dedicated to the design and validation of Adaptive User Interfaces (AUIs), which dynamically adjust to user needs and contexts. A structured protocol for AUI development is presented and validated through a controlled experimental setup, demonstrating the potential of adaptive interfaces to improve usability, reduce errors, and increase user satisfaction.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The main goal of this section is to introduce the deliverable by summarizing the objective of the current deliverable, presenting the methodology, its relation to the other tasks, and analysing its structure.

1.1 Objectives of the Deliverable

Deliverable D3.3 represents the first version of the "Personalised XR Content Development and Adaptation". The objective is to define and implement the foundational elements necessary for delivering adaptive, human-centric XR training experiences within the Worker @ XR5.0 framework. Key goals include the design of a robust system architecture that enables real-time monitoring and interaction through the integration of XR and wearable technologies. A unified model for representing workers will be developed, incorporating both static attributes, such as physical characteristics and skills, and dynamic states like stress and attention levels, to support more precise personalization. Another goal is to ensure seamless interoperability between the XR Training Platform and the Clawdite Platform through the implementation of an API gateway. Additionally, personalization algorithms will be created to dynamically adapt training content based on user behaviour and preferences, fostering a behaviour-driven learning experience tailored to individual needs.

1.2 Insights from Other Tasks and Deliverables

Deliverable D3.3 – Personalised XR Content Development and Adaptation v1 is deeply interconnected with several tasks and deliverables across the XR5.0 project. These interdependencies ensure that the personalised XR content is grounded in human-centric data, dynamically adaptable, and effectively deployed within immersive training environments.

- **T3.4 – Personalised Content Development**: Provides the core content creation foundation for D3.3. It focuses on designing XR experiences tailored to individual workers by leveraging insights from human-centric digital twins (HDTs) developed in T3.3 and specifications from T3.1. The outputs of T3.4, simulations, training modules, and adaptive learning pathways, form the initial content base that D3.3 further adapts and personalises. D3.3 builds upon this by integrating these assets into dynamic, context-aware XR environments.
- **T3.5 – XR Content Adaptation and Personalisation Mechanisms**: Directly supports D3.3 by providing the technical mechanisms for real-time adaptation of XR content. It introduces AI-driven personalisation techniques that adjust XR experiences based on user context, cognitive load, emotional state, and environmental conditions. These mechanisms are essential for D3.3 to deliver responsive and user-centric XR content, ensuring optimal engagement and safety during training.
- **D3.1 – Human-Centric Digital Twin Design and Implementation** v1: Lays the foundational framework for the personalisation capabilities in D3.3. It defines the structure, data collection methods, and functional modules of HDTs, which serve as the primary data source for tailoring XR content. The Clawdite Platform, detailed in D3.1, enables real-time monitoring of worker states (e.g., stress, fatigue, skill level), which D3.3 uses to personalise and adapt content delivery. Additionally, the behavioural and physiological indicators defined in D3.1 inform the adaptive logic implemented in D3.3.
- **D5.3 – Training Platform for Operator 5.0 v1**: Provides the deployment infrastructure for the personalised XR content developed in D3.3. The XR5.0 Training Platform integrates the content and adaptation mechanisms into a cloud-based, device-agnostic environment. It supports the delivery of immersive training programs, user analytics, and real-time feedback loops. D3.3 content is hosted, managed, and streamed via this platform, ensuring scalable and secure access to personalised XR training across diverse industrial settings.

1.3 Structure

This deliverable comprises the following sections/chapters:

- Chapter 1 introduces the content of this deliverable.

- Chapter 2 presents the Architecture and Specifications
- Chapter 3 dedicated to the XR5.0 Training Platform Integration
- Chapter 4 presents the main conclusions of this deliverable and the plan for the next period.

2 Architecture and Specifications

2.1 Worker @ XR5.0 Components

In the XR5.0 ecosystem, seamless and secure user identification is a foundational requirement for enabling personalized, adaptive, and traceable interactions across all integrated systems. The XR5.0 architecture comprises several interconnected components – XR Training Plugin, Hololight, Clawdite Platform, Wearable App, and Asset Administration Shell (AAS) – each of which relies on accurate user identity to function effectively and securely (Figure 1).

To ensure coherent operation and data integrity, each of these systems must be able to recognize and authenticate the user interacting with it. This is essential for:

- **Personalization**: Systems like the XR Training Plugin and AAS tailor content and assistance based on the user's role, skill level, and training history.
- **Access Control**: Hololight Hub and Clawdite Platform enforce role-based access to XR applications, training materials, and sensitive data.
- **Data Synchronization**: The Wearable App and AAS collect and analyse user-specific data (e.g., physiological signals, performance metrics) that must be correctly attributed to the right individual.
- **Analytics and Reporting**: Accurate user identification enables meaningful tracking of progress, engagement, and outcomes across training sessions.
- **Security and Compliance**: Ensuring that only authorized users access specific content or systems is critical for GDPR compliance and enterprise data protection.

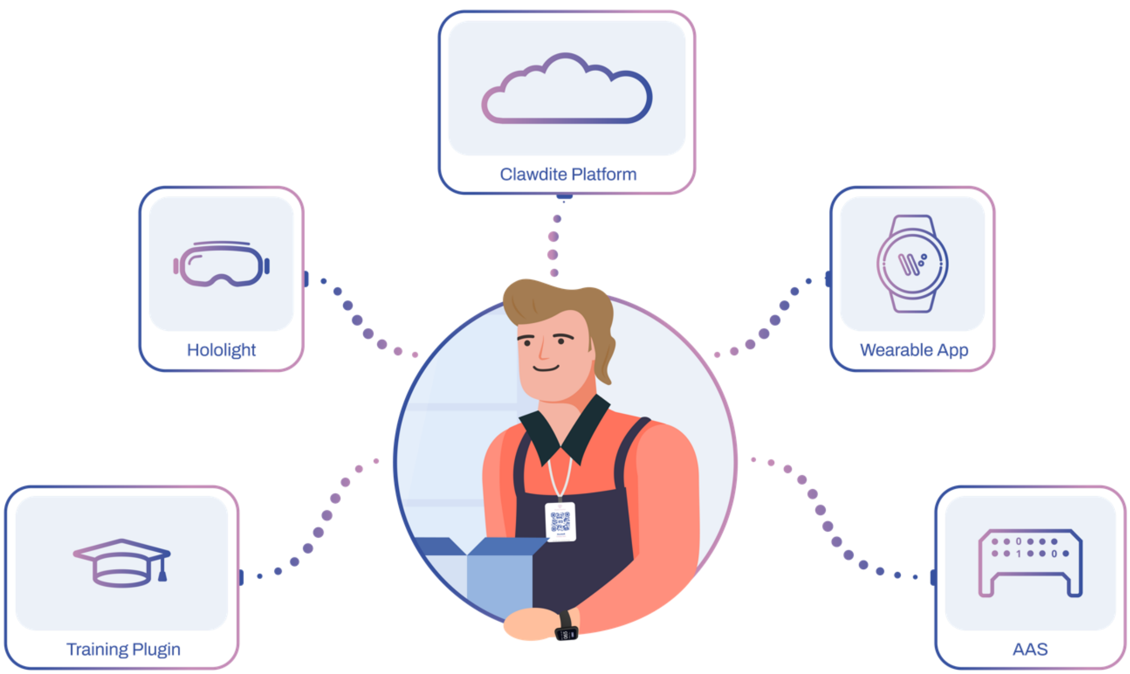

Figure 1 – XR5.0 Components

XR Training Plugin

XR Training Plugin (XRTP) is a core component of the XR5.0 Training Platform. It is designed to enable the visualisation and interaction with XR Training Programs and Materials directly within XR devices. The XRTP is a Unity-based plugin that integrates into XR applications. It works in tandem with the Training Programs

Authoring Tool (TPAT) to deliver immersive, interactive training content to end users in industrial environments [2].

Hololight

Hololight refers to a cloud-based XR streaming and orchestration platform [2]. It is composed of two main components:

1. Hololight Hub – The core orchestration and management platform for XR applications. It enables:
    - Hosting and streaming of XR applications in real time.
    - Centralized application management, including version control and deployment.
    - User and access management with support for Single Sign-On (SSO) and Role-Based Access Control (RBAC).
    - Analytics for tracking usage, performance, and user engagement.
    - Scalability through integration with cloud infrastructure.

It follows a microservices architecture, ensuring modularity and flexibility, and acts as the backbone of the XR5.0 Training Platform.

2. Hololight Stream SDK – A remote rendering solution that allows Unity-based XR applications to be streamed from powerful cloud or local servers to lightweight XR devices. The key features are:
    - Real-time streaming of high-fidelity, graphics-intensive XR content.
    - Device-agnostic: Works across AR/VR headsets like HoloLens 2, Meta Quest, Magic Leap, and even iOS devices.
    - Low-latency and secure transmission using WebRTC, SRTP, and DTLS protocols.
    - No local data storage: Ensures GDPR compliance and minimizes security risks.

Clawdite Platform

The Clawdite Platform is a human-centered platform that characterizes workers and their production environments and supports the creation of Human-Centric Digital Twin (HDTs) that provide scientifically robust foundations and ready-to-use packages for instantiating HDTs [1]. These packages include human-centered aspects such as interactions, events, software-based simulations and predictions. This platform is designed to be interoperable, extensible, scalable, and customizable ensuring flexibility in various industrial applications. It collects multi-dimensional data from workers, structured into three main categories:

1. Physiological Data Collected, via wearables (smartwatches and wearable rings):
    - Heart Rate.
    - Heart Rate Variability.
    - Electrodermal Activity.
    - Skin Temperature.
    - Accelerometry.
2. Behavioural and Cognitive Data, via eye trackers and XR devices:
    - Eye movements (fixations, saccades, smooth pursuits).
    - Gaze metrics (position, duration, path).
    - Pupil diameter (cognitive load, emotional arousal).
    - Blink rate and duration.
    - Head and hand movements (via XR visors)
3. Self-Reported Data, via questionnaires:
    - Stress Analog Scale (SAS).
    - NASA-TLX (Task Load Index).
    - Presence and Usability Questionnaires (UXQ, SSQ).
4. Metadata and Identity, used for session management and data association:
    - Worker ID.
    - Factory ID.
    - Name and company affiliation.
    - Task context and role.

Wearable App

The Wearable App is a software application designed to run on wearable devices such as smartwatches or fitness trackers. In the context of Industry 5.0 and XR5.0, this app plays a critical role in enabling human-centric monitoring and data collection. The Wearable App helps virtualize human entities by collecting real-time data from workers, which is then used to build digital representations of their behaviour and performance. In this regard, the Wearable App are used to:

- Monitor workers continuously.
- Collect physiological and behavioural metrics.
- Adapt data collection based on the worker's task.
- Stream data to platforms for analysis and visualization.

This data is essential for creating HDTs of workers, improving safety, productivity, and enabling immersive XR environments.

Asset Administration Shell

Asset Administration Shell (AAS) is a standardized digital abstraction layer used to represent and manage data from industrial assets, including both machines and environmental sensors. In the XR5.0 project, AAS is also extended to support human-centric data collection, particularly from the eye tracker Pupil Labs and the wearable ring Research Ring. AAS supports the integration of:

1. Pupil Labs to collect:
   - Gaze metrics.
   - Pupil dilation.
   - Blink rate.
   - Other behavioural indicators.
2. Research Ring to collect:
   - Electrodermal data.
   - Pulse.
   - Temperature.
   - Accelerometry.
   - Electrocardiogram.

2.2 Common Worker Approach

One of the main goals of XR5.0 is the personalization of the Training content to each user based on her/his characteristics and preferences. For XR5.0 to be able to provide such behaviour, a common view of the workers using the XR5.0 platform is required, continuously defining the context. The worker context is essential to ensure that the different systems have access to the same user information and information generated by some systems are visible for other systems.

This Worker context sharing between systems can be achieved through the usage of a common user authentication system. In such cases, Single Sign-On (SSO)[3] approaches can be followed to provide a robust authentication mechanism that only require users to login once and be able to able to access the resources and functionalities of the different XR5.0 tools.

Single Sign-On (SSO) is an authentication mechanism that enables users to access multiple related but independent software systems with a single set of credentials. Upon successful authentication with one application, the user's identity is propagated across other federated systems via secure token exchange protocols (e.g., SAML[4], OAuth[5], or OpenID Connect[6]), regardless of domain boundaries, platform

[3] https://datatracker.ietf.org/doc/html/rfc7642
[4] https://auth0.com/intro-to-iam/what-is-saml
[5] https://oauth.net/2/
[6] https://openid.net/developers/how-connect-works/

heterogeneity, or underlying technology stacks. This centralized authentication model reduces credential fatigue, minimizes attack surfaces associated with password reuse, and simplifies identity lifecycle management across distributed environments.
SSO functions through a federated identity model that establishes trust between an Identity Provider (IdP) and a Service Provider (SP). The IdP is responsible for provisioning, managing, and validating user identities, and it issues authentication tokens, and the SP corresponds typically to an application, website, or cloud-based service, that relies on the IdP to authenticate users before granting them access to its services. Therefore, SSO relies on a trust relationship between the SP and IdP.

In the figure below is shown how SSO can be applied to XR5.0 using the concepts of OAuth, as it is already supported by some technologies used by XR5.0. The same login information can be used to login access to each tool and the exact same user information can be shared with each application. This approach is presented in the figure below, where an unauthenticated user attempts to access to any XR 5.0 tools, she/he is redirected to the XR5.0 Identity Provider where her/his login information (i.e. Email/Password) is requested. After the login is performed, a secure token that includes the user information (SSO Token) is generated and provided to Service Provider. That information is then shared with each XR5.0 tool when the user uses it.

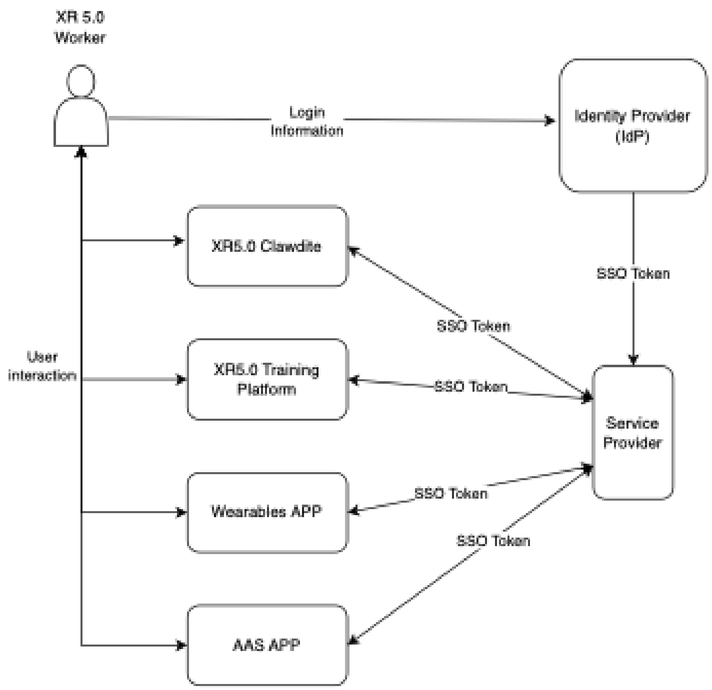

Figure 2 – SSO Approach from XR 5.0 platform

SSO Tokens are implemented using JSON Web Tokens[7] (JWT). JSON Web Token (JWT) access tokens conform to the JWT standard[8] and contain information about an entity in the form of claims. They are self-contained therefore it is not necessary for the recipient to call a server to validate the token. SSO Tokens play a key role in the process on the sharing in the user information across all the tools in the ecosystem. Encode in the token can be stored custom information that can be used to represent any relevant user information. An example of the structure that can be used to describe a XR5.0 worker can be found in the figure below. Ignoring the fields "*sub*" and "*iat*" that are needed for authentication purposes, the remaining fields provide the needed information to identify the worker, the company where she/he works, as well as

[7] https://jwt.io/introduction
[8] https://datatracker.ietf.org/doc/html/rfc7519

identify the working place. Such information provides the needed set of information required for each tool to manage their internal databases and impose any local set of user permissions. This information needs to be collected during the user registration phase and is stored directly in the IdP database as custom user information.


```
{
  "sub": "1234567890",
  "name":"John Doe",
  "company_name":"XR 5.0 company",
  "company_short_name":"XR50",
  "factoryId":"df513e82-d2c2-42e7-ab72-828b03ae01fa",
  "workerID":"38364f22-98b3-49b6-ae75-3b44c5c9bdb4",
  "iat": 1516239022
  }
```


Figure 3 – JSON structure with user information to be shared across the different XR5.0 tools

The implementation of SSO approach requires adaptions to some XR5.0 to adapt and align the user authentication process with the SSO approach and to implement/adapt the methods to acquire/process user information.
From the tools represented in Figure 2, the XR5.0 Training platform is the component the requires less adaptions as they already support SSO, implemented using the OAuth 2.0 protocol. On the other hand, the Wearables mobile app, which currently identifies the Worker through QR code, requires modifications to implement an SSO compatible to login where the worker inputs his credentials before the start of the data collection. Figure 4 shows the graphical interface for supporting such login mechanism. After the login the user receives the workerID that will be used upload the data to the Clawdite Platform.

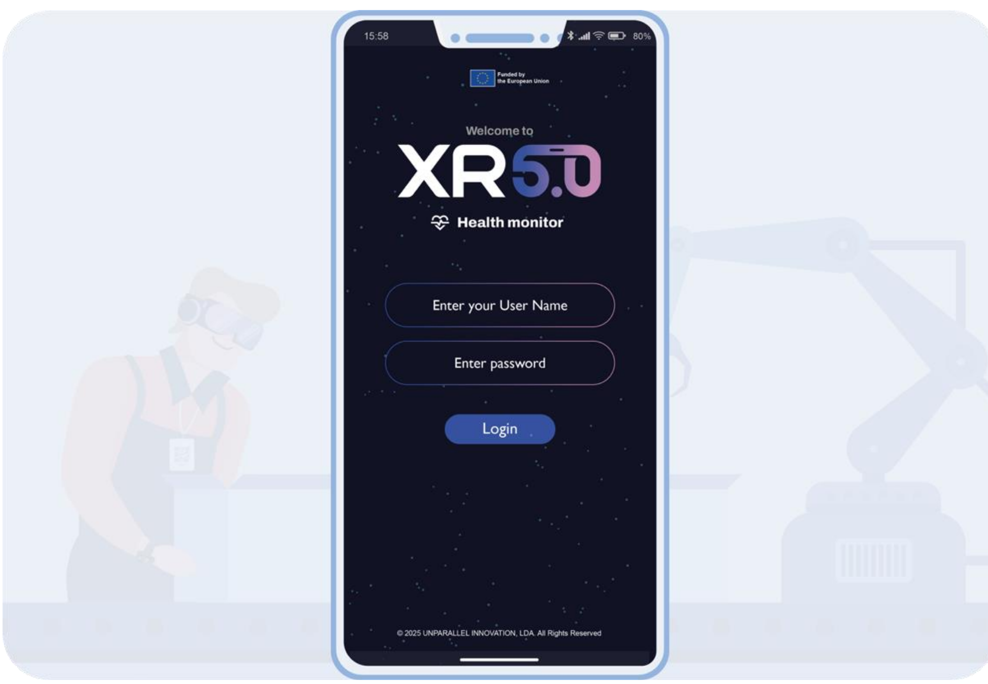

Figure 4 – Login Interface of the Wearables Mobile Application

Similarly, for the connection of AAS with the Clawdite Platform and the corresponding upload of worker-related information, there is the need to ask for worker to perform the login before. Figure 5 presents the login interface provided to the worker for the login in the AAS desktop application and Figure 6 shows the screen presented to the user after a successful login where data is being associated to the logged in user and uploaded to the Clawdite Platform.

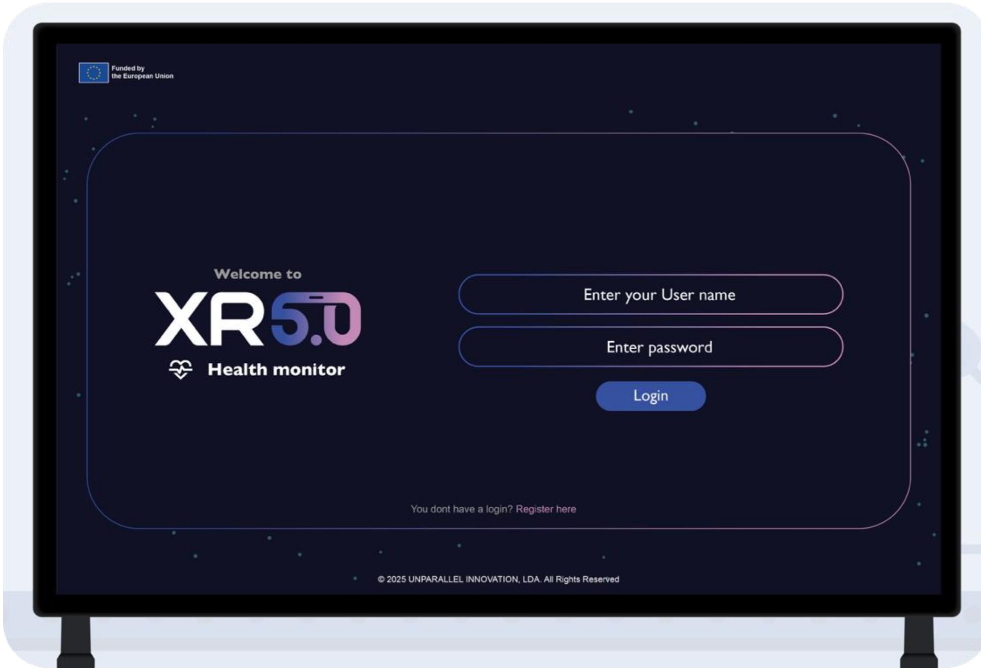

Figure 5 – Login Interface of AAS Desktop Application

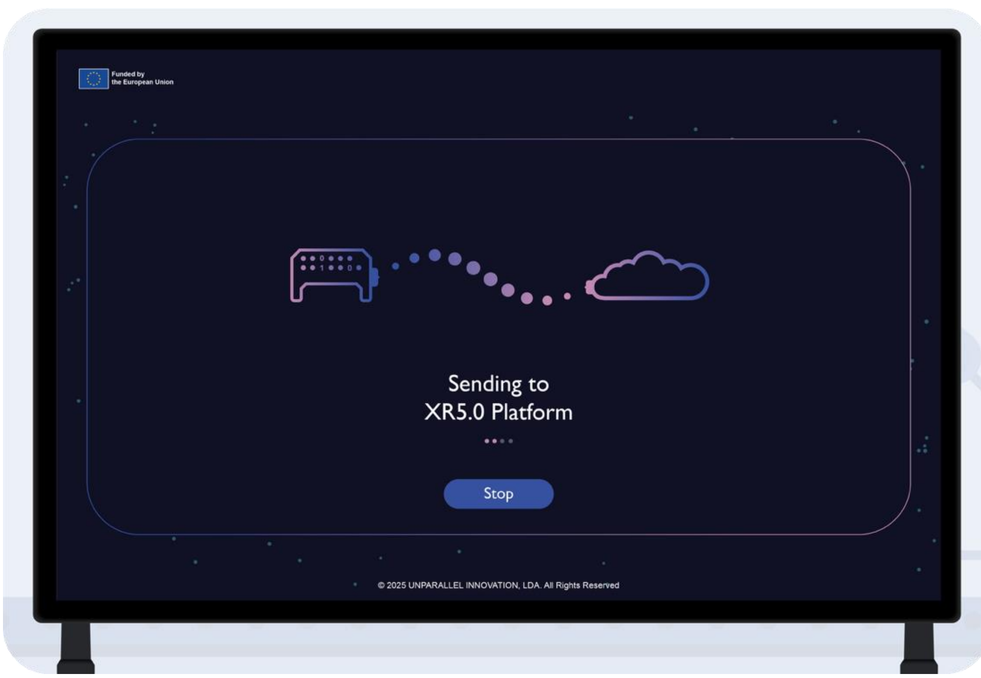

Figure 6 – Interface for Sending Data from the AAS to the Clawdite Platform

Figure 7 presents how the AAS works in a SSO system. In this case the AAS is deployed in the worker's workstation alongside with an app providing a user interface. When the user accesses the user interface, she/he is presented with the interface described in Figure 5. In this interface the user provides the login credentials that is sent to the Identity Provider. After a successfully login the Identity Provider generates the SSO token with the user information and sends it to the Service Provider that in its turn sends it the application here the user performed the login action. When this application receives the SSO token and the confirmation that the Login action as successful, it extracts the Worker ID from the token and sends it to the AAS, triggering the data collection process. In this processes data is collected from devices monitoring the worker and is sent to the Clawdite Platform using the Worker ID as context.

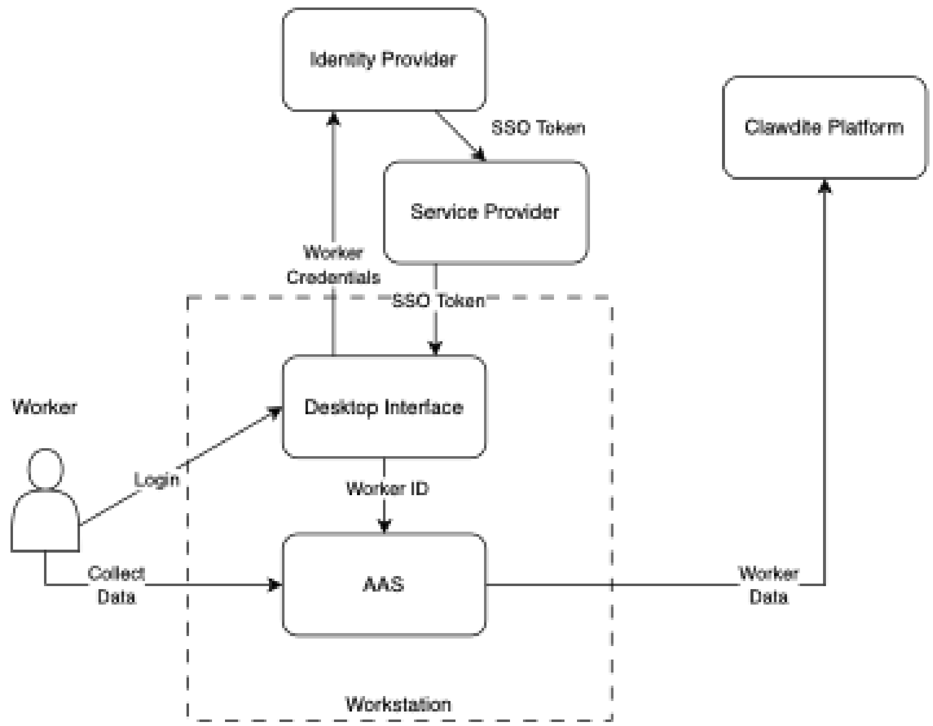

Figure 7 – SSO Login in AAS Application and Data Upload

While the Clawdite Platform is not intended to have workers performing login directly on it, the user information encode on the SSO tokens is relevant for the creation records to store workers Static and Dynamic data.

2.3 Worker Dimensions

The development of HDTs within the XR5.0 framework relies on a comprehensive understanding of the human operator through a structured set of worker dimensions. These dimensions are essential to ensure that the digital twin accurately represents the individual's capabilities, context, and real-time state, enabling personalized, adaptive, and effective training and support.
Worker dimensions are typically categorized into two main types (Figure 8):

- **Static Dimensions**, represent stable characteristics of the worker that define their profile and operational context, and include:
  - Role and responsibilities within the organization.
  - Skill level, certifications, and training history.
  - Physical attributes relevant to ergonomics and task execution.
  - Assigned equipment, tools, or workstations.
  - Preferred learning styles or interface configurations.

Static dimensions are foundational for initializing the HDT and tailoring training programs, access rights, and user interfaces to the individual's baseline.

- **Dynamic Dimensions**, reflect the real-time or frequently changing aspects of the worker's state and environment, and include:
  - Cognitive state.
  - Emotional state.
  - Physiological signals.
  - Performance metrics.
  - Interaction behaviour.
  - Environmental context.

Dynamic dimensions enable the HDT to adapt continuously to the worker's current condition, providing real-time feedback, adjusting training difficulty, and ensuring safety and well-being.
Together, these dimensions form a holistic model of the operator, allowing the HDT to function as a responsive, intelligent, and personalized digital companion. This approach supports the goals of Industry 5.0 by enhancing human-machine collaboration, improving training outcomes, and promoting a safer, more inclusive, and efficient industrial environment.

Figure 8 – Worker Dimensions

2.3.1 Static Dimensions

Static dimensions are foundational attributes that define the baseline characteristics of a human operator within an HDT.
Incorporating static dimensions is essential for ensuring that the HDT is initialized with accurate and relevant information about the individual. This enables the system to assign appropriate training programs, configure user interfaces, and enforce access control policies. Static dimensions also serve as a reference point for interpreting dynamic data, allowing the system to distinguish between expected and anomalous behaviour.
Moreover, static dimensions support interoperability across XR5.0 components by providing a consistent identity and profile structure. This ensures that all systems operate with a shared understanding of the user, enabling coherent personalization, analytics, and decision-making.
In summary, static dimensions provide the structural backbone of the HDT, enabling personalized, secure, and context-aware interactions across the XR5.0 ecosystem. The most relevant static dimension to considered include:

1. Demographic Data:
    - Age, gender, nationality.
    - Location (current and past).
    - Languages spoken.
2. Educational Background:
    - Degrees and certifications.

   - Institutions attended.
   - Areas of expertise or study.
3. Professional Experience:
   - Job titles and roles.
   - Industries worked in.
   - Skills and competencies.
   - Career goals
4. Health and Wellness History (if relevant and with consent):
   - Medical history.
   - Fitness and lifestyle habits.
   - Mental health indicators.
5. Cultural and Social Context:
   - Cultural background.
   - Religious or ethical values.
   - Social environment and influences.
6. Digital Behaviour and Preferences:
   - Online activity patterns.
   - Preferred platforms and tools.
   - Content consumption habits.
7. Personality and Cognitive Style:
   - Personality traits.
   - Learning style (visual, auditory, kinesthetic).
   - Decision-making preferences.
8. Goals and Motivations:
   - Short- and long-term personal goals.
   - Motivational drivers (e.g., achievement, connection, curiosity).
9. Emotional and Behavioural Patterns:
   - Typical emotional responses.
   - Behavioural tendencies in different contexts.
10. Ethical and Privacy Preferences:
    - Data sharing preferences.
    - Consent boundaries.
    - Trust levels with AI systems.

2.3.2 Dynamic Dimensions

In the XR5.0 framework, dynamic dimensions are essential to ensure that the HDT accurately reflects the real-time state and behaviour of the human operator. These dimensions enable the HDT to adapt continuously to the user's needs and environment.

By integrating these dynamic elements, the HDT becomes a responsive and personalized digital representation. This allows for real-time adaptation of training content, enhanced safety through monitoring of fatigue or stress, and improved learning outcomes through tailored feedback. Additionally, dynamic dimensions support interoperability across XR5.0 components, ensuring that all systems operate with a consistent understanding of the user.

Ultimately, including dynamic dimensions transforms the HDT into a proactive and intelligent assistant, aligning with Industry 5.0's vision of human-centric, adaptive, and sustainable industrial environments. The most relevant dynamic dimension to considered include:

1. Cognitive State:
   - Reflects the user's mental workload, attention, and focus.
   - Informs adaptive content delivery and interface adjustments.
2. Emotional State:
   - Captures stress, frustration, or engagement levels.
   - Enables empathetic AI responses and pacing of training.

3. Physiological Signals:
    - Includes heart rate, eye tracking, posture, and motion data.
    - Used to assess fatigue, ergonomics, and safety.
4. Skill Level and Performance:
    - Tracks task completion time, error rates, and learning progression.
    - Supports dynamic difficulty adjustment and personalized feedback.
5. Contextual Awareness:
    - Considers environmental factors (e.g., noise, lighting) and task context.
    - Ensures relevance and realism in training scenarios.
6. Interaction Behaviour:
    - Monitors how users interact with XR content (e.g., gaze, gestures).
    - Helps refine UI/UX and improve training effectiveness.

These adaptive capabilities are grounded in the continuous monitoring of dynamic worker dimensions, which are essential for tailoring the training experience to each individual's real-time needs and context. The XR5.0 Training Platform incorporates user tracking and analytics as foundational elements to support adaptive and personalised training experiences. Through integrated analytics modules, the platform monitors user interactions, session durations, training content usage, and performance metrics. This data is used to generate real-time insights for administrators and trainers, enabling them to assess engagement, identify learning trends, and refine training programs accordingly (Table 1). Building on this, the platform envisions a behaviour-driven adaptation framework. A key component currently under development is the Adaptive Interface Engine, which will dynamically adjust interface elements based on user profiles and interaction history. This personalised interface aims to enhance usability, reduce cognitive load, and increase overall training efficiency. Additionally, future iterations of the platform will include AI-powered assistants capable of responding to user queries within the XR environment and suggesting relevant content or guidance based on contextual understanding. While these adaptive features are still evolving, their inclusion reflects the platform's commitment to providing intelligent, user-centric training aligned with Industry 5.0 principles.

Table 1 – Training Platform User Tracking

Category	Description	Outcome
Enrollment	Opens program X page	-
Enrollment	Reads program X about before enrolling	Suggests reading the full about
Enrollment	Opens program X learning path before enrolling	-
Enrollment	Opens program X resources before enrolling	-
Enrollment	Checks program X learning path before enrolling	-
Enrollment	Checks program X resources before enrolling	-
Enrollment	Enrolls program X	Suggests enrolling program
Program	Opens program X resource Y	Suggests opening commonly used resources
Material	Opens material Y description	Suggests opening description
Material	Opens material Y related material Z	Suggests opening related material
Material	Toggles material X fullscreen	Suggests toggling fullscreen view
Checklist	Opens checklist X item Y related material Z	Suggests opening related material
Checklist	Toggles checklist X item Y after item Z	Suggests reordering items
Image	Zooms image X from zoom Y to zoom Z	Suggests changing zoom level
Image	Pans image X from coords Y to coords Z	Suggests panning image to position
Video	Skips video X from chapter Y to chapter Z	Suggests skipping to specific chapters

Video	Pauses video X at timestamp Y	Suggests pausing video at specific timestamp
Video	Replay video X from timestamp Y to timestamp Z	Suggests replaying video between timestamps
Video	Opens video X annotation Y	Suggests opening annotation
Video	Opens video X annotation Y related material Z	Suggests opening annotation related material
Workflow	Reads workflow X about at step Y	Suggests reading the full about
Workflow	Opens workflow X step Y related material Z	Suggests opening related material
3D Model Explorer	Explodes 3d model view	Explodes the 3d model to a certain level
Voice Assistant	Makes query X	Suggests and even makes the query for the user?

3 XR5.0 TRAINING PLATFORM INTEGRATION

3.1 XR5.0 Training Platform

The XR5.0 Training Platform is a cloud-based, modular, and scalable infrastructure designed to deliver immersive, AI-enhanced training experiences for industrial operators [1]. It is a central component of the XR5.0 project, aligning with Industry 5.0 principles by prioritizing human-centric, adaptive, and intelligent training solutions.
The platform enables the hosting, orchestration, and real-time streaming of XR applications to a wide range of XR devices. It integrates several key components:

- **Hololight Hub**: A cloud-based XR streaming service that manages and delivers XR applications with low latency.
- **XR Asset Repository**: A secure, cloud-based storage system for training materials, 3D models, and multimedia content.
- **Training Programs Authoring Tool (TPAT)**: A web-based tool for creating, managing, and structuring XR training programs.
- **XR Training Plugin (XRTP)**: A Unity-based plugin that allows XR devices to access and visualize training content.
- **User Management and Analytics**: Features such as SSO, RBAC, and real-time analytics to personalize training and monitor performance.

The platform supports adaptive learning through integration with HDTs and AI models, enabling training content to adjust dynamically based on user profiles, behaviour, and performance. It also ensures device-agnostic access, allowing operators to use various XR headsets and mobile devices without compromising performance or security.
Ultimately, the XR5.0 Training Platform empowers industrial workers – referred to as Operators 5.0 – to engage in personalized, efficient, and context-aware training experiences that enhance their skills, safety, and productivity in line with the evolving demands of Industry 5.0.

3.1.1 Collection of Personalization Data

The personalization of training content requires the continuously collection of data regarding how the worker uses training content. Such information can be used to define worker preferences and training visualization habits. Moreover, content must be also aware of the worker expertise, experience and of the training material that the user already visualised.
This information can be classified as **static information** (information that is never updated or updated with low frequency) or **dynamic information** (e.g. metrics collected during the visualisation of a specific training content). As dynamic information we can consider actions that users perform to during the visualization of the training content, such as changes in the positioning of the training elements, level of zoom applied, moments when worker pauses/resumes videos or the attention level/engagement of the worker to the different types of content.
While this dynamic data by itself cannot be directly used to personalize the training content, it can be used by specific algorithms that will infer preferences, behaviours and configurations made by workers. This inferred information can potentially be updated each time the worker visualises training content and can be considered **quasi-static data** there are static information subject to more frequent updates.
The main component responsible for the collection of data is the Clawdite Platform, being able to store static data and collect dynamic data. In Figure 9 is presented architecture of Clawdite Platform. In the context of the current discussing, we would like to highlight the capability to support pluggable modules (highlighted in the architecture by an orange box).

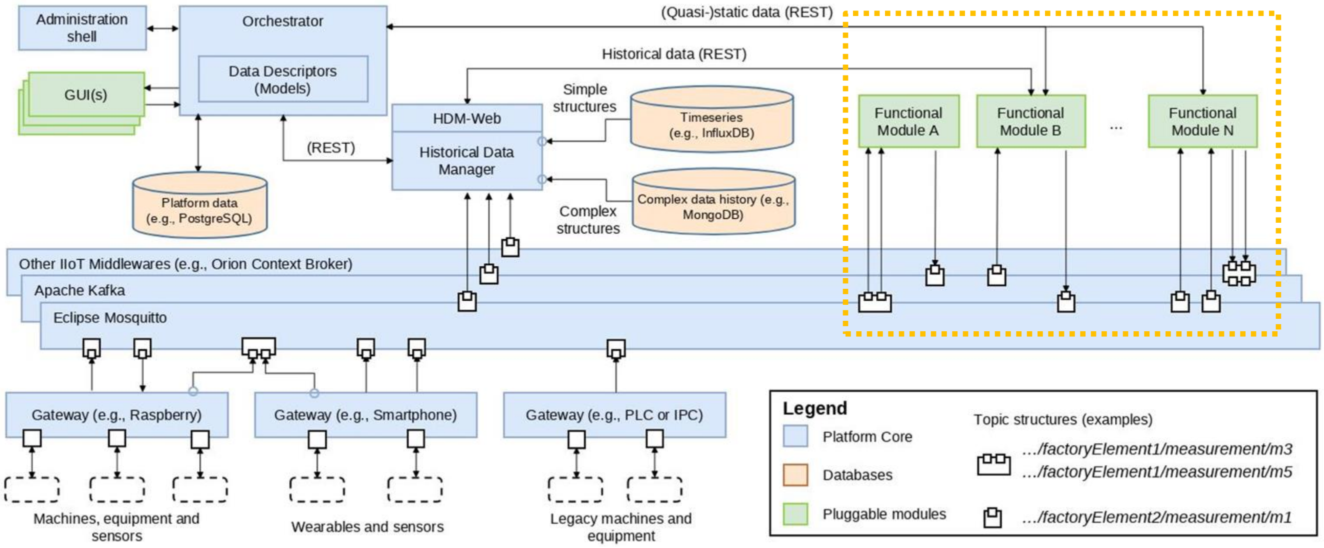

Figure 9 – Clawdite Platform Architecture[9]

The goal of these modules is to allow external components to be plugged onto the platform to provide additional functionalities. Those functionalities may include the processing of data already stored on the platform and/or dynamic data collected from the IIoT Middleware and produce new data (quasi-static data). Specific modules will be developed to consume data produced during training visualisations and produce relevant information that can be used for the personification of the training content.

The process of collection of worker preferences and customization of training content is continuous, being triggered whenever a worker requests the access to a training content. The process is detailed in Figure 10. After the worker requests training content to the XR Training Plugin (XRTP), this component is responsible for the personalization of the requested content before providing it to worker. To do so, it needs to have access to the worker historic data that is available in the Clawdite Platform. Therefore, as represented in the figure below, after the receiving a request for training content, XRTP asks Clawdite Platform for the characteristics and preferences of that worker. With this data, the XRTP runs the personalization algorithms over the requested content and then delivers the personalized training content to the worker.

While the worker is exploring the training content, data about how the worker navigate through the content and how she/he chooses to visualise the information is collected and sent to the Clawdite Platform. There, external modules will periodically analyse the information to infer worker preferences that will be stored in Clawdite and will be taken in consideration the next time the worker accesses training content.

[9] https://clawdite.spslab.ch/docs/architecture/

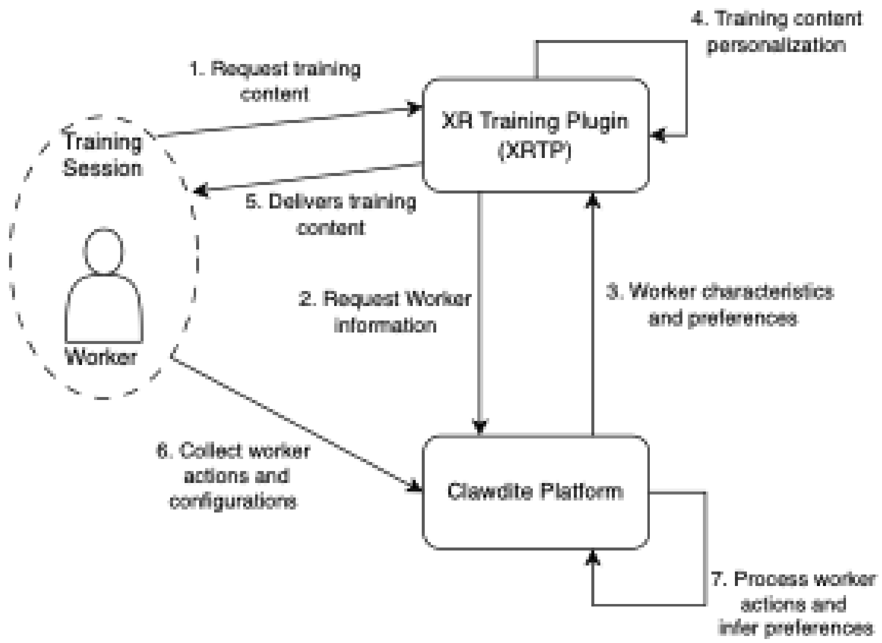

Figure 10 – Process of training content personalization and delivery

3.1.2 Personalization Algorithm

The Personalization Algorithm is designed to dynamically adapt the immersive training environment to the individual preferences and behaviours of each operator. This feature enhances usability, engagement, and learning efficiency by tailoring the XR experience to how users naturally interact with content. The algorithm continuously monitors and learns from user interactions within the XR environment (Figure 11). For example:

- If a user consistently places training videos on the left side of their XR view and zooms in for better visibility, the system will begin to automatically position and scale videos in that preferred location upon launch.
- If the same user regularly opens images in full-screen mode and places them on the right side, the platform will preconfigure the environment to reflect this behaviour whenever image-based content is accessed.

This adaptive behaviour is powered by a combination of:

- **Interaction tracking**: Monitoring spatial placement, scaling, and engagement patterns.
- **User profiling**: Associating preferences with user roles, tasks, and historical usage.
- **Real-time environment configuration**: Automatically adjusting the XR interface layout and content presentation based on learned preferences.
- **Technical Implementation**

The personalization engine is integrated into the Adaptive Interface Engine of the XR Training Plugin that leverages user session analytics from the Hololight Hub, training material metadata from the XR Asset Repository and training program structure from the Training Programs Authoring Tool. These components work together to deliver a context-aware and user-centric interface, ensuring that each training session begins with an environment optimized for the individual user. This offers several key benefits that enhance the overall training experience. By presenting familiar layouts and interactions, it significantly reduces cognitive load, allowing users to focus on the training content rather than adjusting their environment. This streamlined interaction leads to improved training efficiency, as users spend more time engaging with the material and less time configuring their settings. Additionally, the personalized nature of the experience fosters a sense of comfort and control, which in turn boosts user satisfaction, motivation, and content retention.

Figure 11 – Personalization Algorithm Representation

3.1.3 Human-Centric Behaviour Driven Personalization

The advent of Industry 5.0 (I5.0) is reshaping manufacturing by prioritizing human-centric goals such as worker well-being, safety, and productivity alongside technological advancements, particularly Artificial Intelligence[3][4]. A significant challenge in this shift is balancing these objectives with efficiency and productivity while enhancing Human Computer Interaction (HCI) [5][6][7]. User Interfaces (UIs) are fundamental to I5.0, facilitating communication between users and systems to enable task execution. However, in high-pressure manufacturing environments, traditional UIs often fail to accommodate varying skill levels, task complexities, and real-time operational needs, as they primarily rely on contextual features and enabling technologies such as hardware [8].

In this context, Adaptive User Interfaces (AUIs) are crucial for assisting operators in interacting with increasingly complex systems, improving operator efficiency, promoting learning, and enhancing the overall User Experience (UX). AUIs enable dynamic adaptation of interfaces, tailoring them to user needs and preferences while considering contextual factors like skill level, task complexity, and environmental variables. This approach aligns with research on human-centered adaptive Human Machine Interfaces (HMIs) [9], context-aware Mixed Reality (MR) interfaces [10], and pattern-based adaptive interfaces supported by behavioural models [11]. For instance, context-aware AUIs have shown significant benefits in manufacturing environments, improving productivity, reducing errors [12], and providing task-specific guidance through various modalities, enabling non-specialists to perform complex tasks with minimal training. Adaptive multi-modal UIs facilitate real-time interaction with physical objects [13] and dynamically adjust based on contextual parameters through predefined adaptation rules.

While various forms of dynamic adaptation are known (e.g., adjusting information density, modifying control layouts, providing task-specific assistance, or guiding users through complex procedures with minimal cognitive effort [14][15][16]), several challenges remain in implementing AUIs:

- **Algorithmic adaptability** is limited by the constraints of real-time contextual understanding [17]. Systems can dynamically adjust layouts or information density, but inferring user intent across diverse scenarios requires significant advances in multi-modal sensing and predictive modelling [5][18]. Misinterpretations of operator behaviour or environmental conditions may lead to inappropriate adaptations, such as oversimplifying interfaces for expert users or overwhelming novices with excessive data [13].

- **User trust** in AUIs remains a critical psychosocial hurdle [19]. Operators often exhibit scepticism toward automatic and opaque interface modifications and may prefer to disable adaptation. Transparent adaptation logics and explicit user feedback are essential for building user trust.
- **Development frameworks** supporting the AUI lifecycle are under investigated in the literature [20]. Systematic approaches are desirable, particularly concerning UX design and usability.

To review recent studies on AUIs in manufacturing contexts, the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology was used. The scope of the review was restricted to studies published in English between January 2019 and January 2025. Scopus and Web of Science were selected as source databases, and articles were retrieved using the following set of keywords: *adaptive interface; adaptive user interface; adaptive system; manufacturing; human-computer interaction; human-machine interface; human-centered; intelligent user interface; user experience; user-centered design*. Keywords were combined with logical operators to compile a comprehensive list of articles ('OR'), or relate technological solutions with domains of interest ('AND').

The following criteria were used to filter relevant studies:

- Studies must examine Human-Machine Interface and Human-Computer Interaction in industrial manufacturing contexts, with a focus on user interfaces.
- Studies must describe and analyse the use of both adaptive and non-adaptive UI systems.
- Studies must provide empirical evidence supporting their conclusions, including data obtained through evaluation methods and usability studies.

An overview of the results is presented in Figure 12.

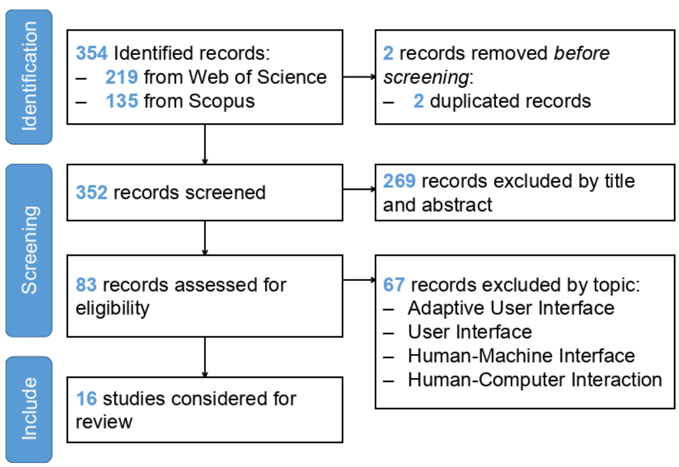

Figure 12 – Overview of the Selection Process with the PRISMA Method

A total of 354 studies were retrieved from the two databases (219 from Scopus, 135 from Web of Science). From these, 269 studies were excluded by screening titles and abstracts. At this stage, the selection of studies was guided by their relevance to HMI and HCI in manufacturing contexts, with a particular focus on UI design and adaptivity mechanisms, ultimately leading to the selection of 16 articles. Table 2 provides an overview of the final studies selected, along with the year and topics that contributed to the study.

Table 2 – Studies Selected from Literature Review

Source	Year	Topic
[19]	2019	Designing 3D printer software interfaces for nontechnical users
[20]	2021	AUI using nonlinear autoencoders to enhance humanmachine interaction
[21]	2021	Human-centered framework for selecting worker assistance systems
[22]	2021	Design of models HMIs for software applications in

		I4.0
[23]	2022	Conceptual model of the Internet of Behaviors for adaptive intelligent systems
[12]	2022	Reference architecture and algorithms for adaptive assistance systems in industrial environments
[24]	2022	Interface and interaction design principles for MR
[25]	2023	Framework for designing adaptive dynamic automation between humans and machines
[26]	2023	Framework for designing AUI in XR environments
[27]	2023	Human-Machine Interaction dataset for AUIs
[28]	2024	Automated methods for evaluating usability and UX in smart products
[29]	2024	Adaptive virtual training system based on universal design
[30]	2024	Framework and development of a human-centered UI/UX model for a collaborative system
[31]	2024	User preference analysis for AI-assisted strategy recommendation systems
[32]	2024	Design guidelines for cognitive ergonomics in humancentered cobot applications
[16]	2024	Framework for the development of AUI in S-PSS

Literature Review Findings

In recent years, numerous systems have been developed and evaluated for various manufacturing applications. These systems extend beyond production to include supply chain management [21], personnel skill development [22], and equipment maintenance [23]. The design principles of HMI and HCI have emerged as central themes in most studies, with a strong emphasis on personalization and user experience evaluation for human-centered systems. Within the context of manufacturing systems, these concepts are frequently associated with worker support and training systems, including XR, VR, and AR. These studies significantly contribute to the design of human-centered UI, providing methodologies and tools for developing more intuitive, efficient, and adaptive solutions.

However, there is a lack of standardization in the design of AUIs, which limits the development of consistent and effective interaction paradigms. Additionally, most AUI solutions are designed for single users, despite the growing need for systems that dynamically adapt to multiple operators simultaneously.

A key challenge lies in the absence of a clear definition of the constituents of UI adaptation [24]. Many existing methodologies and frameworks do not incorporate UX or usability evaluations, focusing primarily on solution proposals or validation research [18]. Only a minority of studies disjointedly include either UX or usability evaluations. In several studies, UX design and analysis are limited to the early stages of product development and do not address the evolution and adaptability that interface systems should possess [18]. By incorporating adaptive and user-centered features, personalized experiences can be created that enhance learning, support individual performance, and improve ergonomic assistance compared to current support systems [14].

This study explores these critical aspects by addressing the following research questions:

- What key factors should be considered in AUIs compared to traditional static UIs to ensure a user-centered design?
- What methodologies can be developed to dynamically adapt traditional UIs to user-centered characteristics in smart manufacturing systems, addressing their needs and enhancing usability, safety, and performance?

3.1.3.1 The Protocol for AUI Design

The proposed protocol focuses on designing user-centered adaptive logic for AUIs. It offers a systematic approach, combining principles from the *morphological box for adaptation design of assistance systems* [14] and the five general dimensions of the adaptive context life-cycle model [25]. This framework guides developers and designers through seven critical steps in AUI design, illustrated in Figure 13.

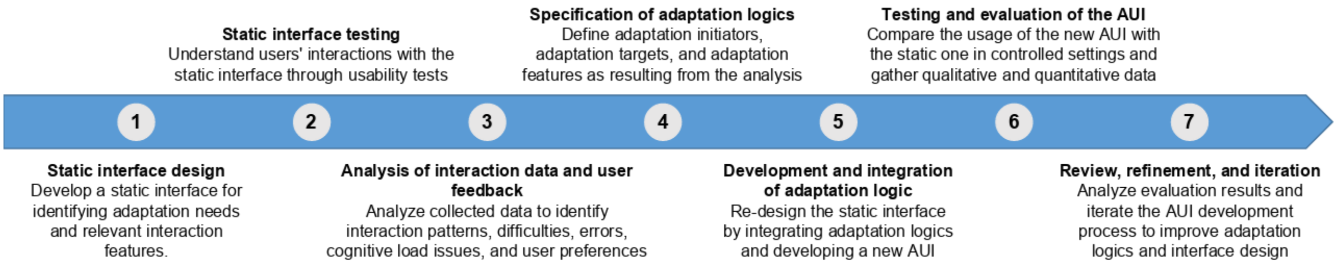

Figure 13 – Framework for the Design of AUIs

1. **Static interface design:** Create a static interface prototype with a clear information architecture that aligns with natural workflow. The main components are summarized in Table 3.

Table 3 – Interface Components and Elements

Component	Elements
Task	Visualization, tracking, progression indicators
Machine	Status indicators, control panels
Instructions	Step-by-step guidance, multimedia support
Errors	Handling, troubleshooting guidance, corrective action prompts
Feedback	Task performance, system response
Logging	Button clicks, menu selections, form completions, task completion sequences and timing, error occurrences, recovery attempts, system state changes

2. **Static interface testing:** Conduct usability tests to gather performance metrics and user feedback. Recruit diverse participants and simulate an industrial scenario. Use methods like:
    - Pre-test interviews for baseline data.
    - Think-aloud protocols during tasks.
    - Log tracking via the UI.
    - Post-task interviews on interface usefulness and challenges.
3. **Analysis of interaction data and user feedback:** Perform quantitative analysis of interaction logs to measure task completion times, error rates, navigation efficiency, information access patterns, and idle periods. Use sequential pattern mining and timeseries clustering to identify behavioural sequences and user strategies. Construct a feature engineering framework from raw interaction data and compare user groups (e.g., novice vs. experienced) to find adaptation opportunities. Use heat maps to visualize attention distribution and identify underutilized or overused features.
4. **Adaptive logic specifications***:* Utilize structured brainstorming techniques to identify adaptation scenarios based on patterns uncovered in prior analysis. Figure 13 presents a framework supporting the design of AUIs by guiding the definition of adaptation rules through *What-When-How* questions. These are tailored to the dimensions of adaptation features, adaptation targets, and initiators of adaptation.

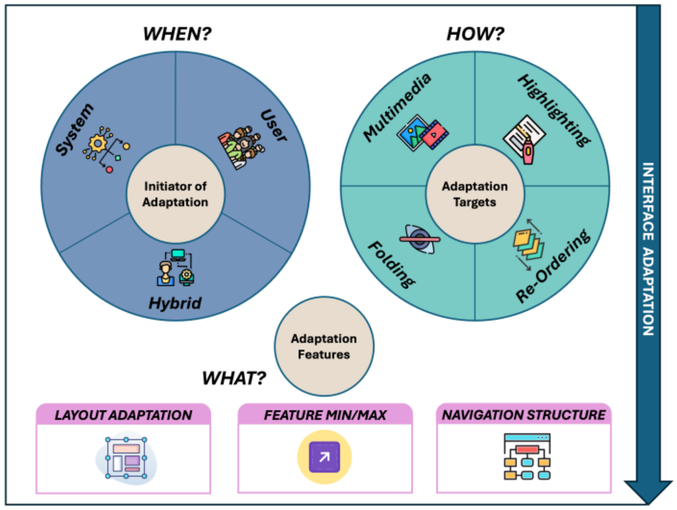

Figure 14 – The Seven Steps of the Proposed Protocol for AUI Development

For initiators of adaptation, develop a taxonomy categorizing triggers along three dimensions:

- *User-driven initiators*: Require explicit characterization of user actions that trigger adaptations. Document these through mappings of user actions, detailing interactions (e.g., button presses, gestures) that signal adaptation needs. Establish thresholds for repeated behaviours indicating user preferences or difficulties.
- *System-driven initiators*: Necessitate comprehensive environmental sensing capabilities. Define contextual conditions such as workstation configurations, ambient lighting changes, noise levels, and production variances that should trigger adaptations. Develop a context-aware sensing framework that monitors these conditions in real-time.
- *Hybrid initiators*: Design a multimodal fusion mechanism combining explicit user actions with implicit behavioural cues. Document physiological indicators (e.g., gaze patterns, interaction hesitation) and contextual factors that collectively signal adaptation needs.

For adaptation targets, conduct card sorting exercises with manufacturing workers to understand their mental models of interface organization. Use these insights to define specific adaptation mechanisms:

- *Content reordering*: Establish priority algorithms based on task relevance, frequency of use, and user expertise levels. Document rules for when and how interface elements should dynamically reposition.
- *Highlighting mechanisms*: Define visual treatment guidelines (colour, size, animation) and determine intensity scaling based on urgency and importance.
- *Information folding*: Establish information hierarchy principles determining which content to expand or collapse. Define criteria for hiding secondary information and methods for users to re-access it.
- *Multimedia enhancements*: Document rules for supplementing text with images, videos, or audio guidance based on task complexity and user preferences.

For adaptation features, combine the initiator of adaptation and the adaptation target to identify how this interaction reflects on the interface. The adaptation features can be classified as follows:

- *Layout adaptation*: Modifies visual elements, including text font and size, colour schemes, spacing, and content arrangement. Techniques such as folding or reordering elements can enhance readability and visual relevance.
- *Task feature minimization/maximization*: Adjusts the amount and complexity of information presented based on the current context. For instance, utilize visual folding to hide non-essential content.

- *Navigation structure*: Alters navigation flows and links to reflect user needs or specific contexts. Adaptation targets, such as highlighting, reordering, and visual reduction, can optimize interface navigation.

A formal adaptive interface requirements document should be derived from this analysis, yielding:
- A classification of adaptation triggers based on observed interaction patterns.
- Prioritized adaptation targets addressing identified usability challenges.
- Empirically-grounded adaptation mechanisms aligned with user preferences.

The process of identifying the interface specification logics should follow an iterative methodology: starting with a minimum viable product implementing core adaptation features and progressively introducing more sophisticated adaptation mechanisms in subsequent iterations.

5. **Development and integration of adaptation logic***:* This involves creating a framework for adaptation decisions, using both rule-based systems and AI components like Reinforcement Learning. Rule-based algorithms translate adaptation specs into if-then rules with priority mechanisms to resolve conflicts. Theoretical rules are then implemented as functional software components. A logging system captures all adaptation decisions and outcomes.
6. **Testing and evaluation of AUIs***:* This step involves designing an evaluation framework to compare the AUI with the static UI. To ensure a fair comparison, participants should represent the same user demographics as in step 2, including a mix of experience levels. The testing environment should be consistent, collecting the same initial metrics to check for improvements, and integrating new ones focused on validation specific to the AUI characteristics. The test should include routine operations and unusual situations that trigger adaptive behaviours. Table 4 categorizes some evaluation metrics, which can be customized or extended depending on the application.

Table 4 – Evaluation Metrics for AUIs

Category	Metric	Description
Performance Metrics	Completion time	Measures the interface efficiency with step completion times.
	Number of interactions	Tracks all user interactions, including unnecessary ones, to assess information clarity.
	Error rate	Counts process mistakes as an indicator of usability and cognitive load.
	Navigation efficiency	Counts unnecessary backtracking, repetitions or detours within the interface.
User Interaction Metrics	Number of suggestion requests	Evaluates how often users request additional information.
	Preferred adaptation mode	Analyzes which adaptation mechanism is most frequently accepted or ignored.
	Interaction patterns	Captures user preferences in navigating different interface elements.
Cognitive Metrics	Workload assessment	Quantifies perceived task difficulty (e.g., NASA-TLX).
	Gaze tracking	Identifies areas of user focus and cognitive demand.
	Completion confidence rating	Measures how confident users feel in task execution with each interface.

Subjective User Feedback	Post-test usability questionnaires	Collects user perceptions of ease of use and adaptability (e.g., System Usability Scale (SUS), User Experience Questionnaire (UEQ)).
	Preference ranking	Allows users to rate their experience with different adaptation modes.
	Qualitative feedback	Gathers insights on frustration points, feature usefulness, and improvement suggestions.

7. **Review, refinement, and iteration***:* This phase transforms evaluation results into concrete improvements through several structured activities. A refined adaptation model must be developed based on evaluation insights. This refinement should address adjustments to triggering thresholds for specific adaptations, visual implementation enhancements, timing or progression modifications of adaptation sequences, and improved transparency of adaptation logics for users.

Guidelines for future adaptive interface development should be established based on project learnings, encompassing best practices for defining adaptation rules, interface design patterns supporting effective adaptation, implementation architectures balancing flexibility with performance, and evaluation methodologies specific to adaptive interfaces.

A continuous improvement process should be established where usage data from deployed interfaces feeds back into adaptation algorithm refinement, ensuring ongoing enhancement of the adaptive system beyond initial implementation.

3.1.3.2 Protocol Application

To validate the protocol introduced in section The Protocol for AUI Design, it has been applied in a simplified yet representative experimental manufacturing scenario. The experimental application involves assembling a LEGO® forklift following a 16-step assembly process requiring the execution of three typical manufacturing operations: picking, assembly, and quality control. The experimental setup, depicted in Figure 15, includes a tablet displaying assembly instructions and a storage box containing the necessary LEGO® components.

Figure 15 – The Experimental Setup to Validate the Protocol

Static interface design

Following the initial step of the protocol, a static interface (Figure 16) was designed with an emphasis on simplicity and functionality, featuring interactive buttons to enhance usability. It includes a title displaying the current task or activity, navigation buttons to move back and forth through the assembly process, and a progress bar showing the process advancement.

Users can access five types of instructional content via dedicated buttons: i) *short text*, offering quick and concise guidance; ii) *long text*, providing detailed and comprehensive instructions; iii) *single components image*, illustrating the individual parts involved; iv) *assembly image*, showing the assembled components; and v) *video*, presenting a step-by-step demonstration of the operation.
As shown in Figure 16, some instructional content might be hidden or unavailable, depending on the current operation (i.e., picking, assembly, quality control).

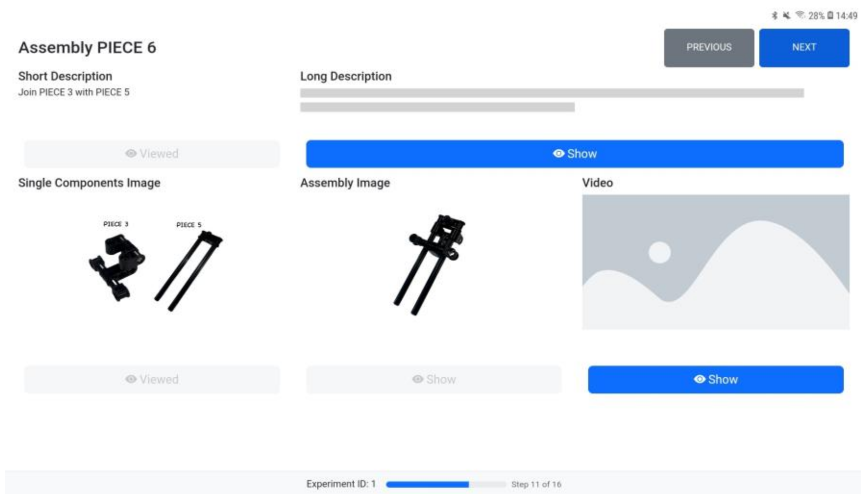

Figure 16 – The Static Interface Guiding Participants Through the Assembly Process

Static Interface Testing

Ten volunteer users, aged between 23 and 40, participated in the experiment after agreeing through a consent form, ensuring data anonymity. Participants were assigned the task of assembling the LEGO® forklift and provided demographic and background information prior to commencing the activity, including gender, age range, and previous experience with LEGO®.
At the beginning of each step, instructional content remained hidden from view. Users had to interact with specific buttons to reveal relevant information before proceeding with their tasks, which included picking, assembly, and quality control operations. This method ensured that every user interaction was recorded in an interaction log for later analysis.
An expert supervised the experiment, monitoring the assembly process throughout. If a participant proceeded to the next step without correctly completing the previous one, the expert advised them to review the instructions and correct the step before continuing.
A total of 625 interactions were logged during the study. These logs will be used for further analysis to explore potential correlations between user characteristics and interaction patterns or task performance. Additionally, post-task interviews with participants provided insights into critical points of the interface and their expectations at each step.

Analysis of Interaction Data and User Feedback

The interaction logs were analysed to extract key insights, including completion times and button interaction sequences. The average duration of the entire experiment was 472.6 seconds.
Completion times varied by task type, with assembly actions taking approximately 34 seconds per action, component picking 23 seconds, and final verification requiring 42 seconds per action. Notably, the first assembly operation (step 2) took longer (65.0 seconds on average) than subsequent parts as users became acquainted with the user interface (UI) and the assembly process. By the second picking onward, users increased their speed and maintained stable interaction times, indicating a growing familiarity with the interface. A comparable pattern was observed for picking times, with the initial picking taking longer. Assembly times decreased progressively until part 6, where complexity led to a significant increase, often necessitating video assistance, demonstrating user reliance on videos during challenging steps.

Analysis of information suggestion requests revealed considerable variations among participants and steps, as illustrated in Figure 17.

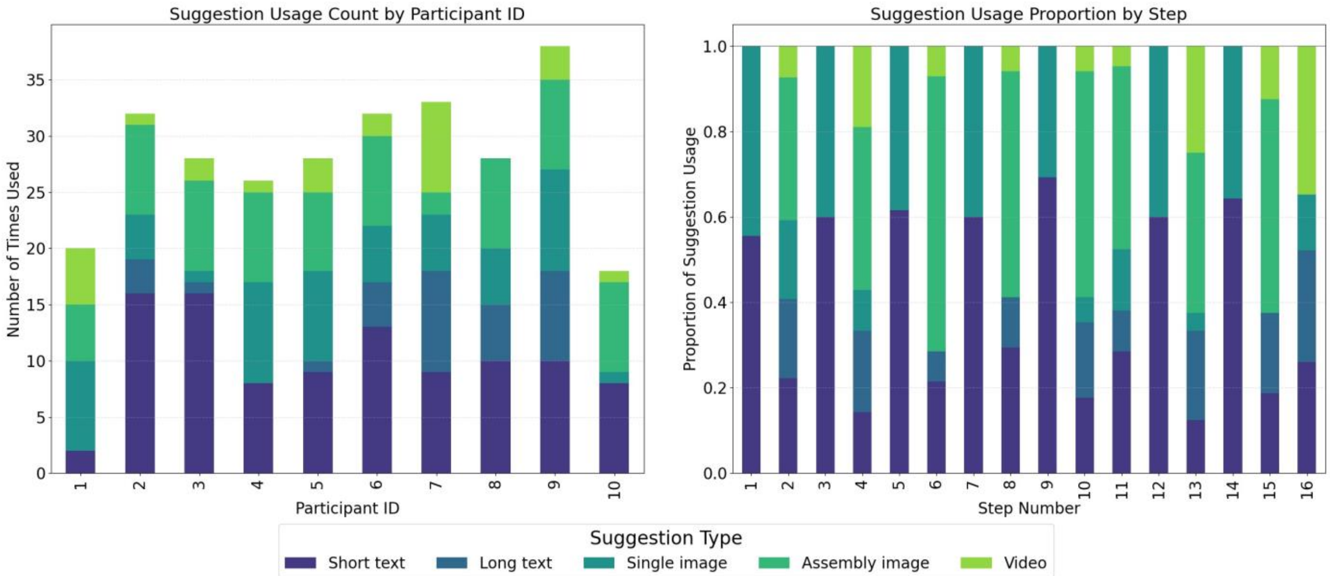

Figure 17 – Information Suggestion Requests per Participant and Step

Short texts emerged as the most frequently used resources, indicating a strong preference for concise, text-based instructions. Visual content, such as individual component images and assembly-focused visuals, played a secondary but important role. Conversely, videos and extensive texts were the least favoured, as shown in Table 5.

Table 5 – Requested Information Suggestions

Information	Suggestion requests	Percentage
Short text	101	35.7%
Long text	31	11.0%
Single image	55	19.4%
Assembly image	70	24.7%
Video	26	9.2%

A noteworthy distinction was identified between expert users (i.e., participants in the fastest 50%) and novice users (i.e., participants in the slowest 50%). Experts showed a preference for visual content (31.7% for individual parts, 27.3% for assembly images), while novices relied more heavily on textual instructions (59.5% short texts, 16.3% long texts). This suggests that experts benefit from direct visual guidance, whereas novices require more detailed textual descriptions.

Moreover, transition patterns indicated a strong dependence on short texts and assembly views, with frequent transitions from assembly image to short description (53 occurrences), from short description to single image (39 occurrences), and from single image to assembly image (30 occurrences).

These insights underscore opportunities to enhance guidance mechanisms through AUI integration to improve user experience, expedite assembly time, and reduce errors. This can be achieved by refining content presentation to prioritize concise descriptions and images. Additionally, the interface should dynamically adjust information suggestions based on the assembly phase and user behaviour, ensuring more relevant and efficient guidance.

Adaptative Logic Specifications

The adaptive logic is crafted to initiate alterations in the interface based on particular events within the application. It delineates which elements should be adjusted and at what times, adhering to the Event-Condition-Action architecture, commonly utilized in AUI [13]. This structure seamlessly integrates with events occurring during the construction of the LEGO® forklift and the application's specifications. The adaptive logic specification emphasizes *task feature minimization*, ensuring that users encounter only the most pertinent information for each task. This *folding* adaptation strategy streamlines the interface by highlighting critical elements.

This system uses both *system-driven* and *user-driven* initiators to manage information complexity. It adapts to user preferences based on past interactions:

- **Event**: New assembly step initiated.
  - **Condition**: User often requested training videos before.
  - **Action**: Display a training video, hide less used instructions.
- **Event**: New picking step initiated.
  - **Condition**: Data shows users prefer image-based instructions.
  - **Action**: Display image-based instructions, hide others.

For initial testing, only a few specifications were used to validate the protocol's effectiveness and interface adaptability. User skills, task times, and other variables were excluded to simplify testing and isolate the impact of adaptation strategies. Future iterations will include more parameters to enhance the adaptive logic.

Development and Integration of Adaptation Logics

To implement AUIs, various strategies were developed in accordance with the adaptation logic specification. Two straightforward yet effective adaptation modes were defined alongside a static baseline to facilitate structured comparison. Each mode initiates with a single piece of unlocked information per step, allowing users to request additional suggestions via corresponding buttons, thereby indicating their preferences. The adaptation modes are specified as follows:

- **Pattern-Guided Mode:** This mode determines the initial information displayed based on common interaction patterns observed during the static interface testing phase. For instance, it may automatically present a component image for picking tasks or an assembly image for assembly tasks.
- **Preference-Adaptive Mode:** This mode begins by displaying the same content as the Pattern-Guided Mode. However, it continuously adapts by tracking the information users most frequently request for each task category (picking, assembly, control). Recent user selections are given higher priority to reflect evolving preferences and guide information pre-selection in subsequent steps.

Within these modes, if a user proceeds to the next step without requesting further suggestions, the initially provided information is considered sufficient, even if it may not be the preferred option. Conversely, when additional suggestions are requested, those selections are likely more informative and essential for the user. In some cases, a combination of specific suggestions may be necessary to complete a task effectively, as a single piece of information might not always be adequate.

Testing and Evaluation of AUIs

To compare the AUI with a static user interface, a Static Mode was developed and utilized *as a base*line condition. In this mode, the short text, identified as the most frequently selected content during static interface testing, is always visible by default. Other types of content can be accessed through interactions with the corresponding buttons.

A total of 30 participants were involved in the validation step. Each participant interacted with only one type of interface. The defined modes are integrated into the UI to dynamically adjust the level of guidance provided. Regardless of the mode, the user journey follows a structured approach:

- The execution begins with a single predefined piece of information for each step, as outlined in Table 6.

- Users can request additional suggestions through interface interactions.
- Once a specific step is completed, the system selects future suggestions based on predefined adaptation logics.
- User interactions, information requests, and errors are tracked to compute evaluation metrics.

Table 6 – Predefined Content Information

Step type	Static Mode	Pattern-guided Mode	Preferenceadaptive Mode
Picking	Short text	Short text or single image	Single image*
Assembly	Short text	Assembly image	Assembly image*
Control	Short text	Video	Video*

*Dynamically adapted based on user interactions

This approach persists until the assembly procedure concludes. Each participant completes a single assembly task using a designated adaptation logic. To ensure fair evaluation, 30 new participants – distinct from the initial static testing group – were involved in the experiments, with 10 assigned to each adaptation mode.
The following quantitative and qualitative metrics, defined in Table 4, were used to assess the AUIs:

- *Completion time*: Measures efficiency by assessing the total duration required to complete the assembly process, reflecting cognitive load and interface usability.
- *Number of interactions and suggestion requests*: Tracks how often users interact with the UI and request additional information suggestions beyond the initial one, indicating potential gaps in information clarity or adaptability logic.
- *Error rate*: Errors suggest insufficient or misleading information. They are manually annotated by the experiment supervisor based on assembly errors.
- *Qualitative metrics questionnaires*: Collects subjective feedback on the UI's adaptability, usefulness, clarity, usability, and user satisfaction using targeted questions.

The evaluation of validation metrics led to several observations. The completion times per step across different adaptation modes and step types were quite similar, with the Preference-adaptive mode being the fastest and the Static mode slightly slower than the Pattern-guided mode, as illustrated in Figure 18 Notably, the three worst performers were all associated with the adaptive modes.

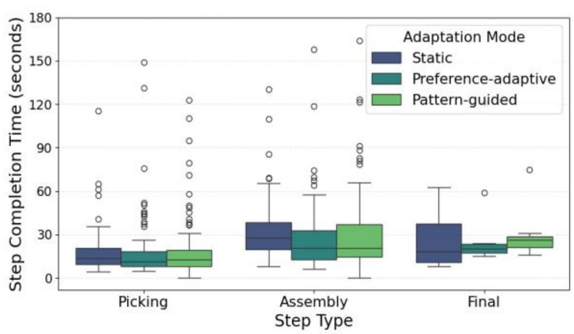

Figure 18 – Step Completion Time by Step Type and Adaptation Mode

Figure 19 illustrates quantitative and qualitative metrics. The Static mode showed a higher number of interactions and suggestion requests, suggesting potential issues with user understanding and alignment. In comparison, both adaptive modes required fewer additional suggestions. Additionally, the number of errors was lower in the adaptive modes, possibly due to enhanced UI clarity and suggestions. The qualitative metrics also indicated better performance in the adaptive modes, particularly regarding usefulness, clarity, usability, and user satisfaction, with the Pattern-guided mode performing the best. Some participants noted that the Static UI provided an adequate level of adaptation.

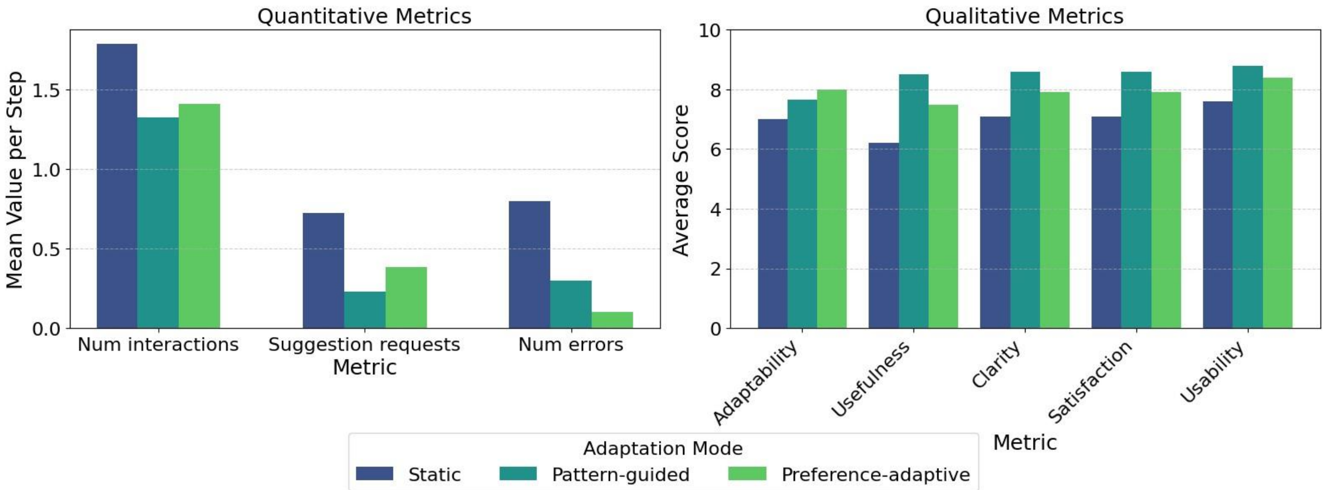

Figure 19 – Comparison of Quantitative and Qualitative Metrics by Adaptation Mode

Review, refinement, and iteration

Following the evaluation in the previous phase, findings were analysed to refine the adaptation strategies. The primary focus is on optimizing information presentation to reduce unnecessary interactions while ensuring adequate guidance. Progressive disclosure should be structured to provide short text initially, followed by assembly views, with video used for more complex steps. Videos are intended to serve as problem-solving resources rather than primary guides. Step-specific resource emphasis should prioritize short text and single-piece views early in the process, assembly views in the middle, and a combination of short text and assembly views for later steps.

Future research should emphasize A/B testing of personalized interfaces. Additionally, extensive data collection will facilitate the development of Machine Learning models or recommendation systems aimed at providing tailored information suggestions based on user preferences, characteristics, and past interactions. Further areas for refinement include the introduction of user clustering to identify participants with similar behaviours and preferences, evaluating a combination of unlocked information based on execution patterns, and conducting additional iterations with other participants to validate existing findings across diverse user profiles. Future enhancements will also integrate metrics such as user skills and step completion times into the interface adaptation process. These factors will provide deeper insights into individual performance, enabling more precise identification of each user's cluster and, consequently, allowing the system logic to better adapt to the user's specific situation.

Considering these aspects, the iterative protocol phases enable the continuous improvement of the AUI, ensuring its effectiveness in supporting task execution while minimizing user effort.

3.1.3.3 Conclusions

The advancement of Human-Machine Interface through adaptive interfaces in manufacturing environments has demonstrated substantial potential to enhance productivity and efficiency. Key findings indicate that adaptive interfaces effectively reduce assembly time variability and errors, while also increasing user satisfaction by personalizing interactions based on user behavior. The deployment of adaptive algorithms has been instrumental in optimizing these interfaces, facilitating a more intuitive and responsive manufacturing ecosystem.

Nevertheless, the primary limitation of this study lies in the evaluation protocol, which is based on a single laboratory-developed use case. Further validation with diverse applications and more complex adaptation methods is warranted. Future work will encompass A/B testing and adaptive content suggestions grounded in machine learning recommendations, user clustering, skills, and past interaction patterns. Additionally, the assembly processes were not monitored, allowing participants to progress without completing previous steps correctly. Future research should include the installation of cameras to monitor task progression and prevent users from advancing in case of errors or omissions. Moreover, additional features and targets will be considered to enhance the adaptive and user-centered design of the AUI.

4 CONCLUSIONS

Deliverable D3.3 is the first version of "Personalised XR Content Development and Adaptation". Through the integration of Human-Centric Digital Twins (HDTs), wearable technologies, and AI-driven personalization mechanisms, demonstrates how immersive training can be tailored to the unique characteristics, preferences, and real-time states of individual workers.

The proposed architecture and system components – XR Training Plugin, Hololight, Clawdite Platform, Wearable App, and Asset Administration Shell (AAS) – form a cohesive ecosystem that enables seamless data flow, real-time monitoring, and dynamic content adaptation. The use of a unified worker model, incorporating both static and dynamic dimensions, ensures that training content is not only relevant but also responsive to the evolving needs and conditions of each user.

A major contribution of this deliverable is the development and validation of Adaptive User Interfaces (AUIs), which represent a shift from static, one-size-fits-all interfaces to intelligent systems that adjust in real time. The experimental validation, conducted through a structured protocol and user testing, confirms that AUIs can reduce cognitive load, improve task performance, and enhance user satisfaction. These findings underscore the potential of adaptive interfaces to transform industrial training by making it more intuitive, efficient, and engaging.

Despite these achievements, the deliverable also identifies areas for further development, including broader validation across diverse industrial scenarios and the integration of more advanced adaptation strategies. These insights will inform the continued evolution of the XR5.0 platform toward more intelligent, responsive, and human-centric training solutions.

4.1 Plan for the next period

The upcoming phase will focus on the development of personalized XR content tailored to individual workers. This includes designing immersive experiences that reflect each worker's specific characteristics, skill levels, and preferences. The goal is to enhance engagement, learning effectiveness, and overall user experience through adaptive and human-centric XR applications.

REFERENCES

[1] XR5.0, 'D3.1 - Human-Centric Digital Twin Design and Implementation'. 2025.

[2] XR5.0, 'D5.3 - Training Platform for Operator 5.0 v1'. 2025.

[3] A. Bettoni, D. Matteri, E. Montini, B. Gładysz, and E. Carpanzano, 'An AI Adoption Model for SMEs: A Conceptual Framework', *IFAC-Pap.*, vol. 54, no. 1, pp. 702–708, 2021.

[4] M. Da Bormida in Cugurra, S. Koussouris, V. Cutrona, and L. Canetta, 'A Human Rights-Driven AI Assessment of Ethical Industry 5.0 Transition and Sustainability in Circular Value Chains', in *2024 IEEE International Conference on Engineering, Technology, and Innovation (ICE/ITMC)*, 2024, pp. 1–9.

[5] D. A. de J. Pacheco and B. Iwaszczenko, 'Unravelling Human-Centric Tensions Towards Industry 5.0: Literature Review, Resolution Strategies and Research Agenda', *Digit. Bus.*, p. 100090, 2024.

[6] R. Pushpakumar *et al.*, 'Human-Computer Interaction: Enhancing User Experience in Interactive Systems', in *E3S Web of Conferences*, EDP Sciences, 2023, p. 04037.

[7] S. Masiero, J. Qosaj, A. Bettoni, and B. Gladysz, 'Technology-Driven Measures for Human Centricity in the Manufacturing Sector', in *International Association for the Management of Technology Conference*, Springer, 2024, pp. 81–88.

[8] V. Villani *et al.*, 'The Inclusive System: A General Framework for Adaptive Industrial Automation', *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1969–1982, 2020.

[9] E. Montini *et al.*, 'Impact of Collaborative Robots on Human Trust, Anxiety, and Workload: Experiment Findings', in *IFIP International Conference on Advances in Production Management Systems*, Springer, 2024, pp. 401–415.

[10] D. Lindlbauer, A. M. Feit, and O. Hilliges, 'Context-Aware Online Adaptation of Mixed Reality Interfaces', in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 147–160.

[11] J. Y. Chew, M. Kawamoto, T. Okuma, E. Yoshida, and N. Kato, 'Adaptive Attention-Based Human Machine Interface System for Teleoperation of Industrial Vehicle', *Sci. Rep.*, vol. 11, no. 1, p. 17284, 2021.

[12] D. Reguera-Bakhache, I. Garitano, C. Cernuda, R. Uribeetxeberria, U. Zurutuza, and G. Lasa, 'An Adaptive Industrial Human-Machine Interface to Optimise Operators Working Performance', in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2021, pp. 1213–1219.

[13] G. Ghiani, M. Manca, F. Paterno, J. Rett, and A. Vaibhav, 'Adaptive Multimodal Web User Interfaces for Smart Work Environments', *J. Ambient Intell. Smart Environ.*, vol. 7, no. 6, pp. 701–717, 2015.

[14] H. Oestreich, et al., 'Human-Centered Adaptive Assistance Systems for the Shop Floor', in *Human-Technology Interaction: Shaping the Future of Industrial User Interfaces*, Springer, 2022, pp. 83–125.

[15] V. N. S. K. Challa, 'Adaptive User Interfaces: Personalizing User Experience User Profiling and Modeling', *J. Artif. Intell. Cloud Comput.*, vol. 2, no. 2, pp. 1–12, 2023.

[16] S. Dell'Oca, D. Matteri, E. Montini, V. Cutrona, Z. M. Barut, and A. Bettoni, 'Improving Collaborative Robotics: Insights on the Impact of Human Intention Prediction', in *International Workshop on Human-Friendly Robotics*, Springer, 2024, pp. 1–15.

[17] A. Ghosh, 'Enhancing Human-Computer Interaction in Smart Manufacturing Environments: Designing Intuitive and Contextual Interfaces for Operators and Supervisors', *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 4, no. 12, pp. 2151–2155, 2022.

[18] A. Carrera-Rivera, F. Larrinaga, G. Lasa, G. Martinez-Arellano, and G. Unamuno, 'AdaptUI: A Framework for the Development of Adaptive User Interfaces in Smart Product-Service Systems', *User Model. User-Adapt. Interact.*, pp. 1–52, 2024.

[19] F. Hartwich, C. Hollander, D. Johannmeyer, and J. F. Krems, 'Improving Passenger Experience and Trust in Automated Vehicles Through User-Adaptive HMIs: "The More the Better" Does Not Apply to Everyone', *Front. Hum. Dyn.*, vol. 3, p. 669030, 2021.

[20] S. Brdnik, T. Hericko, and B. Šumak, 'Intelligent User Interfaces and Their Evaluation: A Systematic Mapping Study', *Sensors*, vol. 22, no. 5830, 2022.

[21] K. Sallam, M. Mohamed, and A. W. Mohamed, 'Internet of Things (IoT) in Supply Chain Management: Challenges, Opportunities, and Best Practices', *Sustain. Mach. Intell. J.*, vol. 2, pp. 3–1, 2023.

[22] V. Villani *et al.*, 'A User Study for the Evaluation of Adaptive Interaction Systems for Inclusive Industrial Workplaces', *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3300–3310, 2021.

[23] D. Mourtzis, J. Angelopoulos, and V. Zogopoulos, 'Integrated and Adaptive AR Maintenance and Shop-Floor Rescheduling', *Comput. Ind.*, vol. 125, p. 103383, 2021.

[24] M. H. Miraz, M. Ali, and P. S. Excell, 'Adaptive User Interfaces and Universal Usability Through Plasticity of User Interface Design', *Comput. Sci. Rev.*, vol. 40, p. 100363, 2021.

[25] K. Todi and T. R. Jonker, 'A Framework for Computational Design and Adaptation of Extended Reality User Interfaces'. 2023.