The Visual Computer
https://doi.org/10.1007/s00371-024-03472-6

RESEARCH

Knowledge-based approach to adaptive XR interface design for non-programmers

Mikołaj Maik[1] · Jakub Flotyński[1] · Krzysztof Walczak[1]

Accepted: 7 May 2024

Abstract
Customizing extended reality (XR) interfaces presents a significant challenge, especially for users with limited programming expertise. This paper introduces the method for adaptation of XR interfaces (MAXI-XR), a novel approach to simplify the customization process of XR user interfaces through knowledge technologies. MAXI-XR offers a user-friendly solution for interface design, supporting users with varying levels of technical skills. The basis of MAXI-XR is its Semantic Knowledge Base, which facilitates intelligent adaptations through advanced querying and reasoning, enabling the extraction of user-specific information for context-based XR interface adaptation. The functionality of MAXI-XR is demonstrated by its application in a VR stock market data visualization system. This system demonstrates MAXI-XR's ability to adapt to complex and data-intensive environments according to user requirements, improving the interaction experience. Furthermore, the method's scalability and ease of maintenance make it a versatile tool for a wide range of applications beyond stock market visualization, suggesting its potential for broader adoption in various XR domains.

Keywords Extended reality · Knowledge representation · Adaptive 3D user interfaces · HCI · Stock exchange visualization

1 Introduction

The growing popularity of 3D interfaces is revolutionizing how we interact with digital environments. These interfaces, along with extended reality (XR) technologies such as virtual reality (VR), augmented reality (AR), and mixed reality (MR), are enhancing the way content is presented in a three-dimensional space. The interaction techniques employed surpass conventional methods, moving beyond keyboards and touchscreens to more immersive modalities [4]. This advancement in 3D interfaces has made significant impacts across various fields, including design [25], visualization [21], healthcare [19], education [14], cultural engagement [15], and gaming [20].

In the emerging field of XR, the design of user interfaces plays a critical role in facilitating smooth interaction between users and the virtual environment. Adaptive user interfaces in extended reality hold great potential as they can dynamically accommodate the specificity of user's environment, tasks, capabilities, and preferences, providing a personalized interaction experience [17, 22]. However, the development of adaptive XR interfaces is associated with challenges, such as accurately modeling the user's preferences and behavior for effective personalization [2, 16, 23].

Semantic technologies facilitate intelligent adaptations due to their advanced querying and reasoning capabilities [5, 8, 26]. These technologies enable the extraction of user-specific information from user's data, allowing for a real-time adaptation of the user interface to the user's context.

In this paper, we present a method for adaptation of XR interfaces (MAXI-XR), which enables the creation of more personalized immersive user experiences by utilizing the reasoning capabilities of the semantic technologies. MAXI-XR focuses on bridging the gap for individuals lacking advanced programming expertise when adapting XR interfaces. The progress of XR technologies has been rapid, but a significant obstacle persists in developing and customizing 3D interfaces. This obstacle arises from the need for expertise

✉ Mikołaj Maik
mikolaj.maik@ue.poznan.pl

Jakub Flotyński
jakub.flotynski@ue.poznan.pl

Krzysztof Walczak
krzysztof.walczak@ue.poznan.pl

[1] Department of Information Technology, Poznań University of Economics and Business, Poznań, Poland

Published online: 08 June 2024

in programming and 3D design. This constraint not only limits the number of potential creators but also restricts the broad appeal and user-friendliness of XR for non-technical users. The presented method addresses this disparity by offering a user-centric and intuitive approach to interface design. Our objective is to make the creation of adaptive XR interfaces easier by utilizing semantic technologies and user-friendly tools such as the Unity Editor and Stardog. This approach simplifies the design process and enhances usability for individuals with varying experiences and skill sets. Additionally, our approach emphasizes the semantic representation of user profiles, which allows for the customization of the XR interface based on each user's specific requirements and preferences. It is imperative that XR technologies demonstrate continuous technological advancement while prioritizing user-friendliness, as these systems must be able to effectively accommodate users with varying abilities, preferences and expectations.

The remainder of this paper is structured as follows: The "Related works" section reviews existing research and developments in XR interface adaptation, setting the stage for our contributions. The "Method for adaptation of XR interfaces" section details the MAXI-XR method, explaining its core concepts and implementation strategies. The "A methodological context" section discusses the application of MAXI-XR in industrial settings, emphasizing its role in enhancing training efficiency. Next section, "A methodological framework for developing adaptive XR interfaces" outlines the technical framework of MAXI-XR, including user profile integration and semantic reasoning. The "Experimental evaluation" section analyses MAXI-XR's effectiveness in a practical forklift training scenario. Finally, the "Conclusions and future works" section summarizes the study's key findings and suggests directions for future research in this area.

2 Related works

Exploring adaptive interfaces in extended reality is a dynamic and evolving field, pivotal in modern interactive system design. The diversification of XR applications across various hardware, data types, and user demographics has underscored the necessity for interfaces that can intuitively adjust to these parameters. This adaptation is essential to improve the user experience and spans content presentation, interaction techniques, and device compatibility [11].

Recent advancements in semantic technologies have greatly simplified the process of creating 3D content in XR environments. Using ontologies, a fundamental element of semantic technologies has enabled content representation that surpasses platform limitations, resulting in improved content discoverability and easier reuse [3]. This paradigm shift is especially beneficial in collaborative and co-creation scenarios, allowing individuals with varied technical proficiencies to participate in content development [6].

Current research in semantic technologies has mainly focused on two specific areas. One approach is to enhance 3D content by adding semantic annotations, which can aid in automated processing and improve user-friendliness [18]. Another approach involves modeling various elements of 3D content, including geometry and behavior, and incorporating them with domain-specific knowledge. This integration facilitates content creation for individuals without expertise [24].

Systems like PEGASE exemplify the potential of intelligent, adaptive guidance in learning environments. These systems underscore the importance of user modeling in tailoring content to individual preferences and contexts [1]. Similarly, adaptive hypermedia systems have demonstrated the effectiveness of personalizing content, considering user-specific preferences and situations [10].

By integrating user interface design with semantic technologies in XR environments, presented method stands out in this dynamic field. The purpose of this integration is to enable individuals who are not IT specialists to create and modify XR content effectively. It is achieved by utilizing tools like Unity Editor for interface design and Stardog for ontology and rule design. This approach facilitates the integration of intricate programming with the development of user-friendly interfaces, thereby democratizing the design of XR interfaces and expanding their accessibility to a broader range of creators and users.

Integrating XR development tools and semantic technologies is a notable advancement in streamlining the process of creating XR content. Recent studies have demonstrated positive outcomes in enhancing the intuitiveness and user-friendliness of these tools, particularly for individuals lacking programming expertise [13].

Research on user-centric design principles in extended reality (XR) has also gained attention. These studies emphasize the significance of adaptable interfaces that cater to the diverse needs of users, including individuals without programming skills. The study [7] demonstrates a comprehensive set of general recommendations for developing multimodal user interfaces in XR applications.

Furthermore, similar methods have been successfully implemented in various sectors, including education and healthcare, as evidenced by case studies. These implementations have demonstrated the practicality and effectiveness of these approaches in real-world contexts [12]. Challenges in adapting XR interfaces for a non-technical user base continue to be a research focus, with solutions increasingly leaning toward more intuitive, user-friendly methods. The emergence of no-code platforms in XR content creation has significantly lowered barriers for non-technical creators in areas such as 3D graphics and physics simulation [4].

3 Method for adaptation of XR interfaces

The method for adaptation of XR interfaces (MAXI-XR) operates within a framework designed to customize XR interfaces to individual user profiles using both system-driven adaptivity and user-driven adaptability [9]. The structure shown in Fig. 1 is divided into separate but interconnected domains: the interface presentation context and the context-independent interface representation, with the semantic knowledge base at the center of the interactions.

Adaptivity in MAXI-XR refers to the system's capacity to autonomously adjust the user interface in response to the user's context, behavior, and preferences. This automatic adaptation leverages semantic knowledge base reasoning to intelligently modify the UI, ensuring a personalized interaction experience without user intervention. Adaptability, on the other hand, empowers users to manually customize the UI according to their specific needs and preferences. This feature enables users to actively participate in the design of their interaction experience, reflecting a mixed-initiative approach that blends system-driven adjustments with user-driven customization.

Within the interface presentation context, users provide personal data and preferences through a user-friendly form. The data are contained within a user profile ontology, which provides user information in a format the system can interpret and act upon. This format is known as the semantic user profile representation (SUPR). At the same time, the system's attributes and operational capabilities are abstracted within the system profile ontology. This ontology recognizes system states and contributes to the semantic system profile representation (SSPR). The semantic knowledge base carefully preserves both user and system representations. The ontologies used in MAXI-XR are defined using the Web Ontology Language (OWL), which provides a formal framework for describing the relationships between classes, subclasses, object properties, and data properties.

The MAXI-XR method relies on the semantic knowledge base to combine user and system data driven by semantic reasoning. Guided by a comprehensive set of rules, this reasoning utilizes logical frameworks to analyze ontologies, resulting in insights that impact the adaptive process of the XR interface. In the field of context-independent interface representation, the focus turns to semantic interaction interface representation (SIIR). The XR interface's presentation is abstracted according to user-specific profiles and system parameters. The adaptation process involves the interplay of semantic reasoning outcomes and the interaction representation, resulting in the conceptualization of an XR Meta-Scene.

This Meta-Scene serves as a framework for the XR environment, where the adaptation process uses the logic of the semantic knowledge base to create an XR Scene. This scene is presented as an environment tailored to the user, with 3D models, animations, and interactive components that work together to create a seamless and intuitive user experience. MAXI-XR enhances XR interface adaptation by integrating user data, system constraints, and semantic reasoning. This leads to a highly personalized XR experience that takes into account user preferences and requirements at both design time and run time. The method demonstrates the impact of semantic technologies in creating immersive spaces that are technologically advanced and user-centric.

3.1 Semantic user profile representation

The interaction interfaces of XR scenes consist of geometric components, sounds, and animations, among other elements. The MAXI-XR method enables the flexible adjustment of interface elements in real time, encompassing their type, size, color, and positioning. The main goal is to improve user comfort and efficiency during interactions. The SUPR user profile representation is used to describe various user characteristics and preferences regarding 3D interfaces, allowing for precise customization of the interface to meet the specific needs of each user. The diagram in Fig. 2 presents the user profile ontology schematic diagram, which defines the data model for semantic user profile representation.

The user profile ontology, expressed in OWL, defines the main classes (e.g., *User*, *CognitiveProfile*, *PhysicalProfile* or *PreferenceProfile*), subclasses (e.g., *AnxiousUser*, *DyslexicUser* or *LowFocusUser*), data properties (e.g., *age*, *colorBlindness*, *height* or *movementScore*), object properties (e.g., *hasDemographicProfile*, *hasCognitiveProfile* or *hasPhysicalProfile*), and rules to create a comprehensive and adaptable data model for representing user characteristics.

SUPR includes different categories of information. Below are examples of categories. It is important to note that the knowledge base is extensible and can accommodate a wide range of characteristics:

Demographics The *User* class is associated with data properties like *age*, *gender*, and *preferredLanguage*, defined in the *DemographicProfile* class. By examining these characteristics, content can be more effectively customized for various age groups (children, teenagers, middle-aged, and seniors) and take into consideration differences in abilities.

Physical properties The *PhysicalProfile* class includes data properties such as *height* and *armReach*, and subclasses like *ShortArmReachUser* and *TallUser*. The height feature can impact the adjustment of the 3D interface to the line of sight on the Y-axis, facilitating the comfortable use of interface elements. If a user exhibits slow movements or remains still, the interface can be adjusted to a closer position. If a user maintains a fixed gaze and does not frequently scan their surroundings, it may be more appropriate to position the interface directly in front of them. If a user encounters difficulty accessing specific buttons or interface elements,

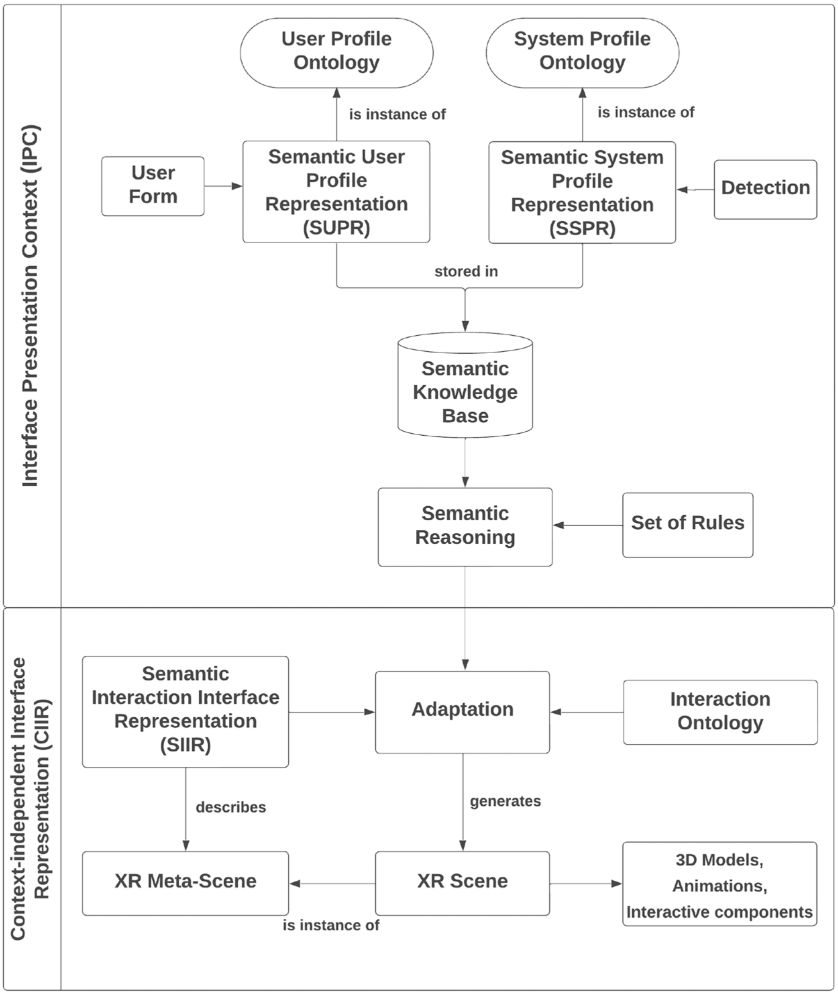

Interface Presentation Context (IPC)
User Profile Ontology
System Profile Ontology
is instance of
is instance of
User Form
Semantic User Profile Representation (SUPR)
Semantic System Profile Representation (SSPR)
Detection
stored in
Semantic Knowledge Base
Semantic Reasoning
Set of Rules
Context-independent Interface Representation (CIIR)
Semantic Interaction Interface Representation (SIIR)
Adaptation
Interaction Ontology
describes
generates
XR Meta-Scene
XR Scene
3D Models, Animations, Interactive components
is instance of

Fig. 1 Method for adaptation of XR interfaces

the interface can automatically adjust its position to enhance usability and facilitate smoother interaction with its functions.

Sensory limitations The *CognitiveProfile* class includes data properties like *colorBlindness* and *dyslexia*, and subclasses such as *ColorBlindUser* and *DyslexicUser*. The *UserWithDisability* class, a subclass of *PhysicalProfile*, represents users with various impairments, including *HearingImpairmentUser*, *VisualImpairmentUser*, and *MobilityImpairedUser*. Adaptations involve adjusting text size for visually impaired users, modifying brightness in low-light conditions, and reducing distractions during interactions. Color adjustment and distraction reduction techniques are employed to cater to individuals with challenges in perceiving colors or experiencing sensory overload. In addition, dyslexic users may find that text readability is improved by using wider spacing and vibrant colors.

User experience The *XRProficiencyProfile* class captures the user's experience level with XR technology through the *experienceLevel* data property. Subclasses like *NoviceUser*, *IntermediateUser*, and *ExpertUser* represent different proficiency levels. The *motionSicknessInXR* data property is used to describe the user's susceptibility to motion sickness, with subclasses such as *HighMotionSicknessUser* and *LowMotionSicknessUser*. Adjusted vignettes can be configured in XR environments to cover a significant portion of the field of view for individuals who experience discomfort. Users can focus on specific interface elements without being distracted by peripheral movements.

Preferences The *PreferenceProfile* class encompasses various user preferences, with subclasses like *AudioPreference*, *VisualPreference*, and *InteractionPreference*. These preferences are connected to the User class through the *hasPreferenceProfile* object property. Data properties such as *userAudioSetting*, *userVisualSetting*, and *userInteractionSetting* capture specific preference values. Individuals who prioritize comfort may benefit from an XR environment that minimizes potential side effects, such as dizziness, eye strain, or other symptoms commonly associated with using XR technology. Individuals with a lower preference for comfort may exhibit more excellent receptiveness toward fully immersive XR experiences and may require fewer limitations.

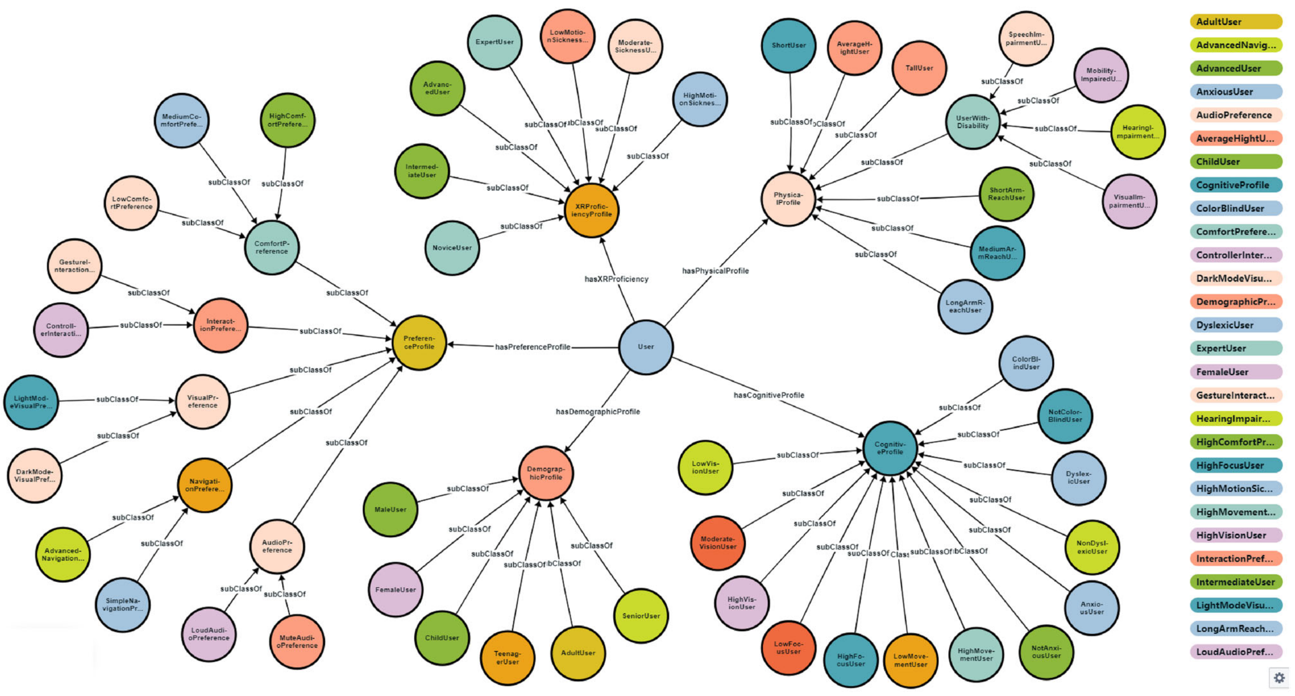


Fig. 2 User profile ontology

3.2 Semantic system profile representation

SSPR provides a systematic description of the hardware and software setup of the target XR environment. The characteristics encompass visual display properties, audio functionality, interaction mechanisms, and performance metrics, among other factors. The framework considers necessary configurations for advanced XR platforms, including high-resolution graphics, spatial audio processing, precise hand and eye tracking, adjustable field of view, and built-in support for augmented and virtual reality. The data from SSPR allow for retargeting functionality and precise calibration of XR interfaces. The primary objective is to meet the requirements and enhance XR scene presentation, ensuring immersive, responsive, and user-tailored XR experiences. The structure of the system profile ontology, which defines the data model for SSPR representation, is presented in Figs. 3 and 4.

The SSPR representation follows a hierarchical structure, as shown in Fig. 3. It begins with the XR system, which may consist of multiple XR devices. The XR device is connected to the XR configuration through the ’hasConfiguration’ relation. XR configuration subclasses focus on device aspects, including visual, audio, interaction, haptics, performance, and accessibility. The semantic system profile representation incorporates several information categories as outlined below.

Visual Configuration The *VisualConfiguration* class includes data properties such as *displayFieldOfView*, *resolution*, *refreshRate*, and *stereoscopicDisplay*. These factors collectively impact users’ clarity, immersion, and overall visual experience.

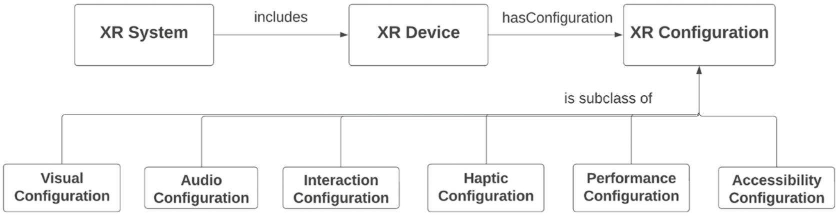
Audio Configuration The audio configuration provides information about the sound’s depth and immersion in virtual environments. The *AudioConfiguration* class captures the audio capabilities of the XR system, with subclasses like *BasicAudioConfiguration*, *InteractiveAudioConfiguration*, and *NoAudioConfiguration*. Data properties such as *audio3D*, *headphoneType*, and *microphonePresence* describe specific audio features.

Interaction Configuration The *InteractionConfiguration* class represents the interaction mechanisms supported by the XR system. Data properties like *eyeTracking*, *gestureRecognition*, *touchInput*, *voiceCommand*, and *spatialAwareness* indicate the presence or absence of these interaction capabilities. The *externalControllers* data property specifies the types of supported controllers.

Haptic Configuration Description of haptic features provided by connected interaction peripherals in the XR setup. It includes compatibility with controllers and tactile feedback mechanisms such as force feedback, touch sensitivity, and vibration intensity.

Performance Configuration The *PerformanceConfiguration* class captures the performance characteristics of the XR system, with subclasses like *LowPerformanceConfiguration*, *MediumPerformanceConfiguration*, and *HighPerformanceConfiguration*. Data properties such as *processorSpeed*, *ramSize*, and *gpuVRAMSize* define the system’s performance metrics.

Fig. 3 Semantic system profile representation schema



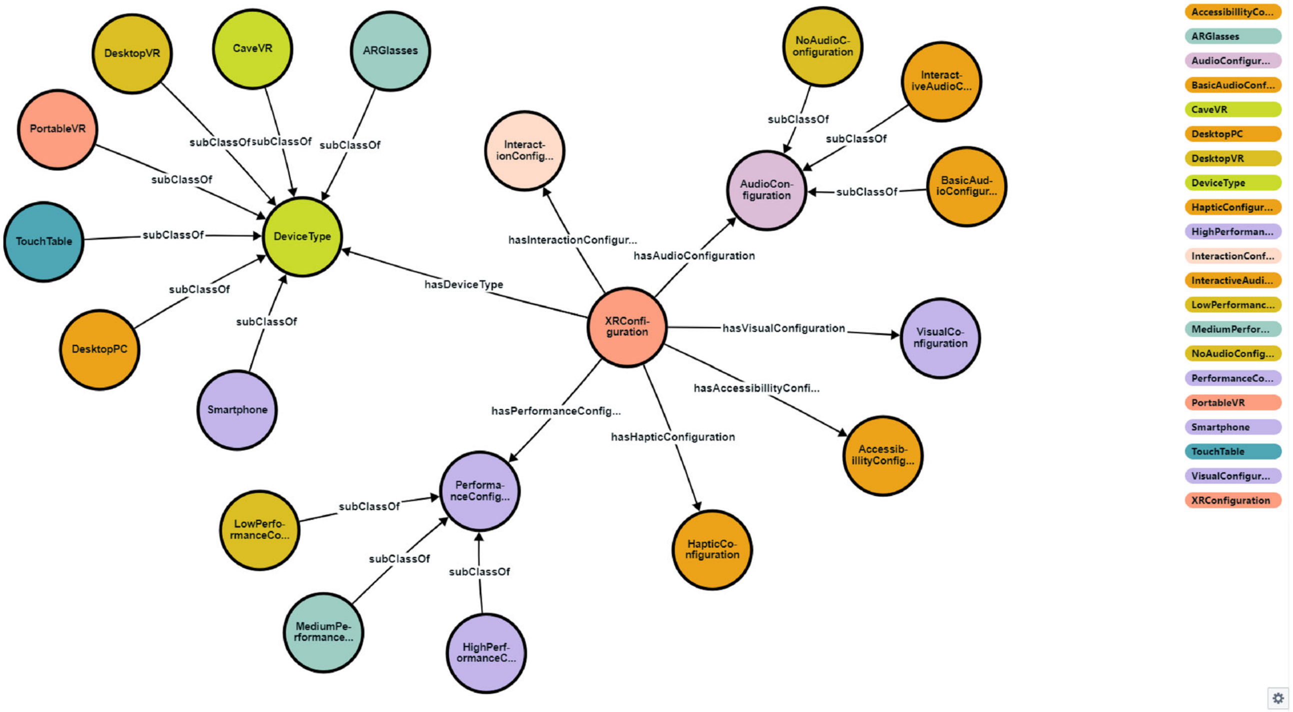


Fig. 4 Semantic system profile representation

Accessibility Configuration The available accessibility mechanisms in XR devices are discussed, considering features such as haptic feedback and subtitle support. These features aim to provide a barrier-free XR experience for all users, regardless of their needs.

Furthermore, the *XRConfiguration* class is associated with the *DeviceType* class through the *hasDeviceType* object property, allowing the specification of the target device, such as *ARGlasses*, *DesktopVR*, or *Smartphone*.

3.3 Interface adaptation

Reasoning is a systematic process that generates new knowledge based on existing information in the knowledge base. This process is crucial in the presented method as it allows the system to make informed decisions on adapting the user interface based on the user's profile and the description of XR system capabilities. The process relies on logical rules and relationships to deduce facts or classifications not explicitly mentioned in the knowledge base.

The SUPR representation involves categorizing users into distinct groups based on their characteristics and preferences. The categories, represented as classes, encompass physical attributes (such as height and arm reach), cognitive traits (like focus level and presence of dyslexia), and preferences (like audio, visual, and interaction preferences). The SUPR profile data linked to each user determine their membership in these classes. Rules have been implemented to assist in the reasoning process. These rules allow the system to analyze user data, thus creating a customized user interface. Here are some examples of rules that can guide the inference logic:

1. **Expert Rule**
   If a user's XR Proficiency Profile shows an experience level greater than 8, they are classified as an `ExpertUser`. This rule identifies users with high levels of experience in XR environments, categorizing them as experts.
   Adaptation: For expert users, the XR interface can offer advanced features and controls, such as customizable

gestures or shortcuts. The system can also reduce the frequency of tutorial prompts and help messages, assuming the user is already familiar with the environment.

2. **High MotionSickness Rule**
   If a user's XR proficiency profile indicates a motion sickness score of 7 or higher, they are labeled as `HighMotionSicknessUser`. This classification is crucial for adapting XR experiences to users' comfort levels.
   Adaptation: For users prone to high motion sickness, the system can implement techniques to reduce discomfort, such as adjusting the field of view, reducing camera motion, and providing stable reference points within the virtual environment. The system can also offer accessibility options like a vignette effect or a static reference frame to mitigate motion sickness.
3. **ColorBlind Rule**
   If a user's Cognitive Profile indicates color blindness, they are classified as a `ColorBlindUser`.
   Adaptation: For color-blind users, the XR interface can employ color-blind-friendly color schemes, such as high-contrast palettes or patterns, to ensure visual elements are distinguishable.
4. **LowVision Rule**
   If a user's cognitive profile indicates a vision score below 4, they are classified as a `LowVisionUser`.
   Adaptation: For users with low vision, the XR interface can offer enlarged text, high-contrast visuals, and adjustable font sizes. The system can also provide audio descriptions or haptic feedback to supplement visual information.
5. **DesktopVR Rule**
   If a user's XR configuration includes a stereoscopic display, head tracking, full environment integration, a refresh rate above 70Hz, and advanced external controllers, the device is classified as `DesktopVR`.
   Adaptation: For users with a DesktopVR setup, the XR system can leverage the advanced capabilities to provide a highly immersive experience. This can include realistic graphics, precise hand tracking, and complex interactions using the external controllers. The interface can also adapt to take advantage of the high refresh rate and stereoscopic display for smooth and convincing visuals.
6. **Loud Audio Preference Rule**
   If a user's audio preference is set to "Loud", they are categorized as `LoudAudioPreference`. This rule identifies users who prefer higher volume levels in their audio settings.
   Adaptation: For users with a preference for loud audio, the XR system can automatically adjust the default volume settings to a higher level. Additionally, the system can prioritize and enhance audio cues and notifications to ensure they are prominently heard by the user.

After the reasoning process is completed, the adaptation step is undertaken. In this phase, each element of the XR user interface included in the semantic interaction interface representation and described with the interaction ontology (expressed in OWL) is mapped to a specific component of the final XR scene. The data model of the SIIR representation is presented in Fig. 5 and shortly described below.

- Elements meant for users to input or set numerical values fall under "NumericalValueSet" within the "UserInteraction" class.
- Inputs designed to initiate specific actions or commands are classified as "CommandInput".
- Prompts or cues that provide contextual assistance are labeled "ContextualHint".
- Mechanisms designed for two-way interaction with users are categorized as "InteractiveDialog".
- Components facilitating adjustment or alteration of existing values are grouped under "ValueManipulation".
- Elements assisting users in navigating through the XR environment are termed "NavigationAid".
- Elements providing real-time visual response based on user actions are referred to as "VisualFeedback".
- Aesthetic features of the interface, such as "Color", "Texture", and "Shape", are part of the "AestheticProperty" subclass within the "InterfaceProperties" class.
- Spatial properties of interface elements, including "Size" and "Position", come under the "DimensionProperty" subclass.
- Mechanisms for delivering information through sound are under "AuditoryDelivery" within the "InformationPresentation" class, while those through touch are grouped under "TactileDelivery", and visual means are part of "VisualDelivery".

By utilizing data from both SUPR and SSPR, the interface is tailored and adapted to align with user preferences and the system's operational capabilities. For instance, a person who prefers tactile interactions may appreciate an interface with "TactileDelivery" components, while someone who values precise input may encounter enhanced "NumericalValueSet" and "ValueManipulation" features. In addition, a user who relies on auditory cues may notice an interface where "AuditoryDelivery" is highlighted. On the other hand, individuals who place importance on visual cues may receive a balanced combination of "VisualFeedback" and "VisualDelivery". The primary objective is to enhance user interaction within the XR framework.

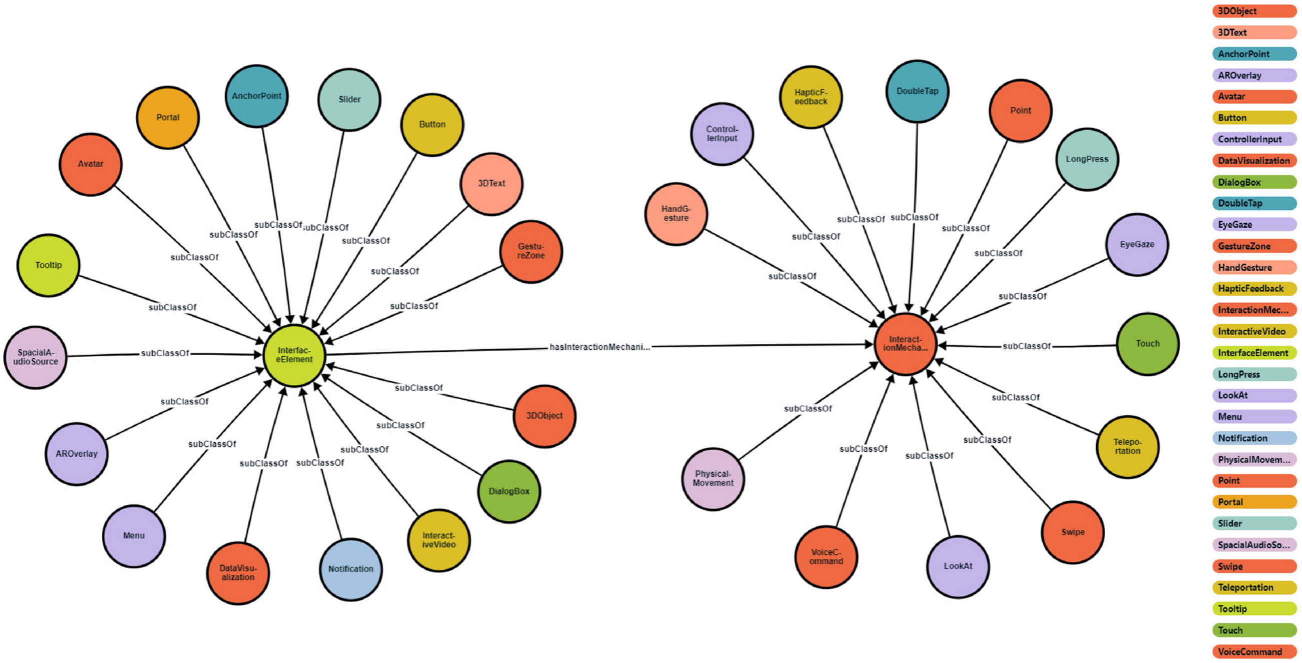


Fig. 5 Semantic interaction interface representation

4 A methodological example: stock market data visualization

Stock market data visualization in VR presents a unique challenge due to the need for real-time data processing and intuitive presentation. This paper introduces an adaptive methodology leveraging MAXI-XR to enhance the understanding and interaction with complex financial data in a VR environment.

An instance of this methodology is observed in a VR stock market dashboard (Fig. 6) developed using the Unity 3D game engine and the Microsoft Azure PlayFab platform. The dashboard serves as a simulation platform for MAXI-XR, allowing for real-time visualization and interaction with various financial data sets tailored to the user's specific needs. This dashboard represents a step forward in financial data analysis, providing users with an immersive and customizable environment.

Using VR controllers such as the HTC Vive, VR gloves like Forte Data, or hand-tracking devices like Leap Motion, users can interact with the dashboard in a highly intuitive manner. The system's flexibility to deploy in both VR and AR environments enhances user engagement with real-world surroundings while utilizing sophisticated financial data visualization tools.

The interface of the stock market dashboard is designed for dynamic interaction and customization. With real-time updates facilitated by the Float Rates API, users can adjust data display settings, such as the time period for stock prices and currency selection. Interaction with the dashboard is user-friendly and intuitive, allowing natural gestures or simple taps to control various elements.

Furthermore, user profiles are captured through a structured form (Fig. 7), which, upon submission, updates the knowledge base. This base, managed by the dotNetRDF library, is critical for adapting the interface to individual user needs, leveraging stored information to adjust interface elements like visuals, interaction methods, and real-time data presentation.

The integration of the MAXI-XR method in this stock market visualization system exemplifies its practical application, showcasing how it can transform the user experience by providing a tailored and interactive environment for financial data analysis. Below, we demonstrate how the adaptation rules from subsection 3.3 are applied.

- If a user is classified as an `ExpertUser` based on their XR proficiency profile, the stock market interface provides advanced technical tools and allows customization of financial charts.
- If a user is labeled as `HighMotionSicknessUser`, the XR system adds a vignette when navigating through the virtual stock market environment.
- For users classified as `ColorBlindUser`, the stock market interface employs high-contrast color schemes for financial charts.

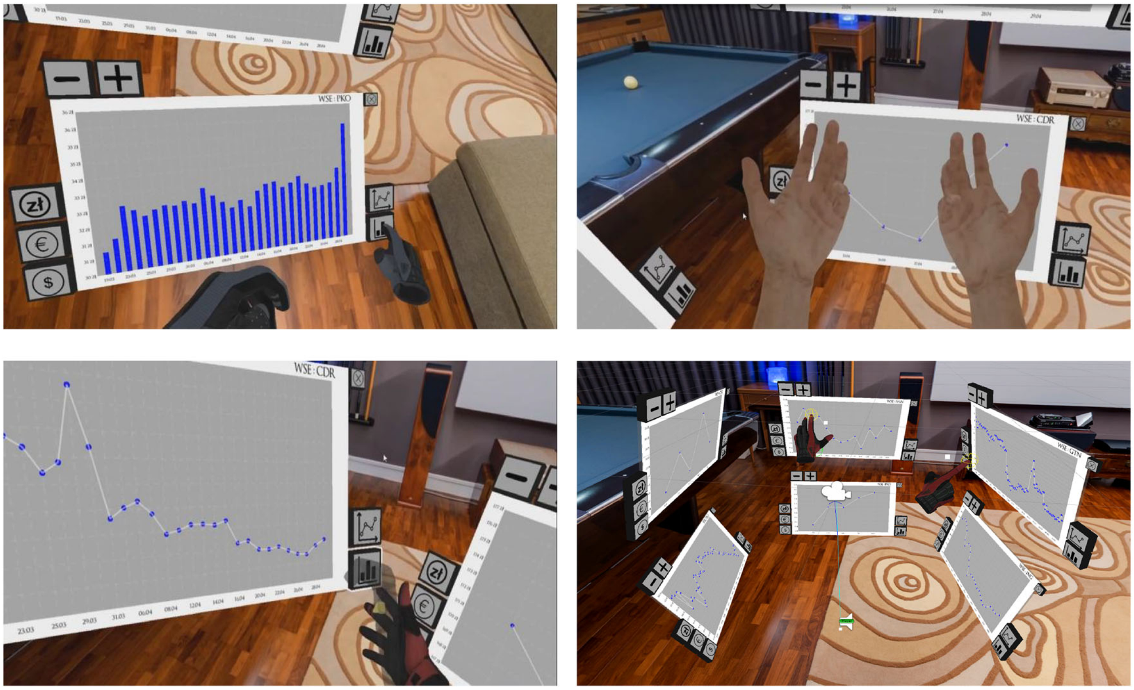
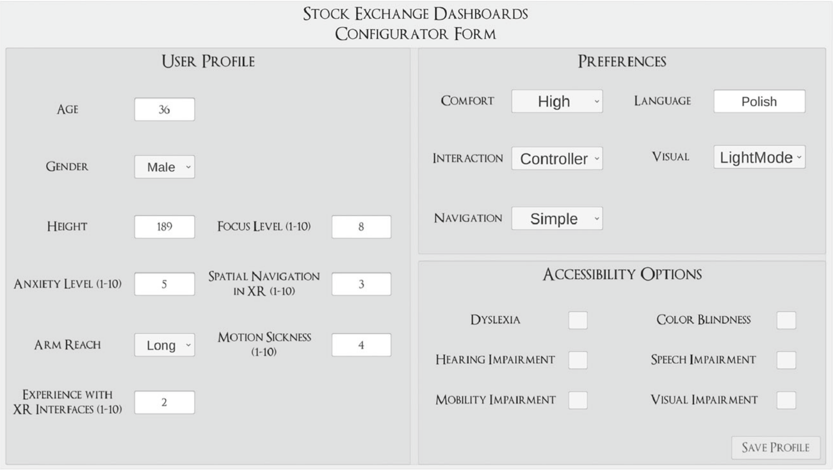


Fig. 6 VR stock market dashboard

Fig. 7 User profile form



Stock Exchange Dashboards Configurator Form

User Profile

Age	36		
Gender	Male		
Height	189	Focus Level (1-10)	8
Anxiety Level (1-10)	5	Spatial Navigation in XR (1-10)	3
Arm Reach	Long	Motion Sickness (1-10)	4
Experience with XR Interfaces (1-10)	2		

Preferences

Comfort	High	Language	Polish
Interaction	Controller	Visual	LightMode
Navigation	Simple		

Accessibility Options

Dyslexia	□	Color Blindness	□
Hearing Impairment	□	Speech Impairment	□
Mobility Impairment	□	Visual Impairment	□

- Spatial properties of interface elements, including “Size” and “Position”, come under the “DimensionProperty” subclass.
- If a user is identified as a `LowVisionUser`, the XR system offers enlarged text and voice-based assistance to help them access and interpret stock market data.

These adaptation rules are triggered based on the user’s profile information and the XR system’s capabilities. When a user launches the stock market application, their SUPR data are processed by the reasoning engine, which infers their classification based on the defined rules. The SSPR data are also considered to ensure that the adaptations are compatible with the XR system’s capabilities. Once the relevant adaptations are determined, they are applied to the XR interface, providing a personalized and optimized user experience.

5 Evaluation

This section presents an evaluation of the method for adaptation of XR interfaces (MAXI-XR) applied to a distinct domain: stock market data visualization. We aim to evaluate MAXI-XR’s applicability and efficiency in a practical setting, demonstrating its ability to create adaptive VR interfaces tailored to the dynamic needs of stock market analysis.

It involves developing a specialized ontology, encompassing attributes and rules that accurately represent the intricacies of stock market data interpretation and user interaction within a VR environment.

A crucial part of this evaluation is rationalizing our methodological choices, particularly highlighting how MAXI-XR significantly improves over traditional programming methods in Unity for VR interface adaptation in financial data visualization.

5.1 MAXI-XR versus traditional programming

We perform a comparative analysis between the MAXI-XR approach and conventional coding practices, mainly focusing on RDF + SPARQL. This comparison aims to highlight the user-centric design of MAXI-XR, emphasizing its potential to improve the efficiency and usability of XR interface development for stock market data visualization. This approach is particularly beneficial for users with limited coding expertise, broadening the scope of advanced XR technologies across various user groups and domains.

- Inherent Limitations in Game Engine Reasoning: Traditional game engines, including Unity, inherently lack built-in reasoning capabilities crucial for dynamic, context-aware adaptations. With its semantic reasoning, MAXI-XR fills this gap, allowing for more nuanced and context-sensitive interface adaptations that are not feasible with standard programming approaches in game engines.
- Barrier to Non-Programmers: Creating adaptive interfaces in environments like Unity requires significant programming skills, making it challenging for professionals with limited or no coding background. MAXI-XR, on the other hand, allows non-programmers, like financial analysts, to create and customize XR environments to meet their needs without extensive technical knowledge.
- Semantic Reasoning and Dynamic Adaptation: MAXI-XR uses semantic technologies to dynamically adapt VR interfaces based on detailed user-profiles and the specific requirements of stock market analysis, offering a level of personalization and adaptability beyond traditional programming.
- Scalability through Ontology Enhancement: The ontology-based nature of MAXI-XR allows for straightforward scalability. By simply adding new rules, classes, object properties, and expanding the knowledge base, the system can adapt to new requirements and scenarios without extensive code modification. This feature makes it easier to integrate additional stock market data or refine user profiles.
- Efficiency in Handling Complex Data: The ontology-based approach of MAXI-XR efficiently handles the adaptability required in the complex and data-intensive environment of stock market analysis, significantly reducing the effort and time required to code specific scenarios and adaptations.

The following sections will analyze the specifics of ontology design, rule implementation, and the practical integration of MAXI-XR within the Unity scene, further demonstrating its effectiveness and adaptability in the context of stock market data visualization.

5.2 Ontology design and rule implementation

5.2.1 Ontology framework development

We developed a detailed ontology named “StockMarketVisualization” using tools like Stardog Designer. We defined specific attributes for classes such as *User* (hasTradingExperience, hasRiskTolerance, hasFrequencyOfUse, prefersDataPresentation, interestedInMarketSegment) and subclasses for *DataPresentation* aligned with user preferences. We established rules and relationships to dynamically adapt the XR interface based on user profiles.

5.2.2 Enhanced user profile classification

We included detailed subclasses for user classification such as BasicTradingExperience, IntermediateTradingExperience, AdvancedTradingExperience, LowRiskTolerance, MediumRiskTolerance, HighRiskTolerance, FrequentUser, OccasionalUser, and RareUser. We integrated these with data presentation styles like BasicDataPresentation, IntermediateDataPresentation, and AdvancedDataPresentation, facilitating a more nuanced user interface adaptation.

5.2.3 Classification rules and relationships

- Trading Experience Rule: “IF a User has a hasTradingExperience level of less than 3, THEN classify them as a ’BasicTradingExperience’.”
- Risk Tolerance Rule: "IF a User has a hasRiskTolerance of ’High’, THEN classify them as having ‘HighRiskTolerance’.”
- Usage Frequency Rule: "IF a User has a hasFrequencyOfUse of ‘Daily’, THEN classify them as a ‘FrequentUser’."
- Data Presentation Preference Rule: "IF a User prefersDataPresentation of ‘Graphical’, THEN classify them as preferring ‘AdvancedDataPresentation’."
- Market Segment Interest Rule: "IF a User is interestedInMarketSegment of ‘Equity’, THEN connect them to ‘EquityMarket’."

These classes and rules, encoded using SPARQL syntax, ensure that the XR interface dynamically adapts, providing a personalized and effective data visualization environment tailored to each user's trading experience, risk tolerance, usage frequency, data presentation preference, and market segment interest.

5.3 Creating user profiles in unity

User Profile Creation Using MAXI-XR Method We used MAXI-XR for constructing detailed user profiles with attributes suitable for comprehensive adaptability in a financial context.

Unity Editor Integration We imported the ontology into the Unity Editor for interactive profile generation, enabling input of attributes and creation of individualized user profiles for financial analysts or traders.

Sample User Profiles and Data Presentation Attributes Examples of user profiles and data presentation attributes include:

- User1: Trading Experience 2 years, Prefers graphical data representation, Visual Acuity moderate. Engages with real-time stock price charts.
- User2: Trading Experience 10 years, Prefers numerical data representation, Visual Acuity high. Analyzes historical data trends.
- User3: Trading Experience 5 years, Prefers mixed data representation, Visual Acuity low. Accesses combined real-time and historical data visualizations.

5.4 Ontology design for XR interface elements

This section explores the development and utilization of an ontology for XR User Interface Elements, focusing on the essential classes and their interrelations. This ontology is specifically designed for dynamic adaptation of scene elements within Unity, based on real-time inference from user profiles. The objective is to enhance the flexibility and user-centric design of XR interfaces in various application scenarios.

Core Classes and Object Properties This ontology is centered around key classes such as *BasicInterfaceElement*, *UserInteractionMode*, and *AdaptiveFunctionality*. These classes form the backbone of the XR interface structure. Subclasses like *GestureControl*, *VoiceCommand*, *InteractiveGraph*, and *DynamicPanel* categorize interface elements by their interaction modes and functionalities.

User Interaction Modes The core class *UserInteractionMode* with subclasses such as *GestureControl* and *VoiceCommand*, specifies the interaction methods available within the XR environment. These modes are designed to offer intuitive and diverse ways for users to engage with the interface.

Adaptive Functionality The class *AdaptiveFunctionality* encompasses adaptive aspects of the interface, with subclasses like *ColorSchemeAdjustment*, *SizeAdjustment*, and *VisibilityAdjustment*. These enable the interface to dynamically respond to user preferences and system constraints, offering a tailored user experience.

Inter-Class Relationships This ontology establishes relationships such as *BasicInterfaceElement* incorporating *AdaptiveFunctionality* and utilizing *UserInteractionMode*. This structure facilitates a coherent and adaptive approach to interface element customization, ensuring that the XR environment is responsive to the unique requirements and preferences of each user.

5.5 Ontology-based scene development

Importing Ontology into Unity The "StockMarketVisualization" ontology is imported directly into the Unity Editor, allowing immediate access to its classes and attributes for scene development. This integration is crucial for adapting the VR interface to the specific requirements of stock market visualization.

Scene Object Tagging with Ontology Elements In the Unity environment, interface elements such as interactive graphs, gesture control buttons, and dynamic panels are tagged using classes from the imported XRInterface ontology. This involves assigning tags like *XRInterface:InteractiveGraph* from the *BasicInterfaceElement* class to an interactive market graph or *XRInterface:GestureControl* from the *UserInteractionMode* class to a button enabling gesture-based data manipulation.

Unity Scene Hierarchy and Element Classification The Unity scene for XR-based financial data visualization comprises various elements, each tagged with a specific class from the "XRInterface" ontology:

- Interactive Market Graph (1 Object):
  - Ontological Class: XRInterface:InteractiveGraph
  - Description: Central element for displaying real-time financial data interactively.
- Gesture Control Buttons (4 Objects):
  - Ontological Class: XRInterface:GestureControl
  - Description: Buttons enabling switching between different data sets and views via gestures.
- Dynamic Information Panels (2 Objects):
  - Ontological Class: XRInterface:DynamicPanel
  - Description: Panels showing contextual information and updates in the financial world.

- Visualization Adjustment Sliders (3 Objects):
  - Ontological Class: XRInterface:SizeAdjustment
  - Description: Sliders for customizing visualization parameters, including time frames and data types.
- Market Alert Notifications (2 Objects):
  - Ontological Class: XRInterface:VisibilityAdjustment
  - Description: Notification alerts for significant movements in the market or breaking news.
- Descriptive Text Labels (2 Objects):
  - Ontological Class: XRInterface:VoiceCommand
  - Description: Provides descriptive labels and explanations for various data points and UI elements.

The implementation of this Unity scene ensures that each element is accurately tagged with its corresponding ontological class from the "XRInterface" ontology. This structured approach enables dynamic interaction within the XR environment, where interface elements adapt based on the user's profile, preferences, and specific requirements for financial data visualization. Integrating this ontology into the Unity scene demonstrates the practical application of adaptive VR interfaces, offering an effective platform for immersive financial analysis. This methodology contributes to a personalized and immersive experience in financial data visualization, highlighting the adaptability and utility of the ontology in crafting a customized VR environment for financial analysis.

6 Evaluation metrics

This section focuses on a comparative analysis of the efficiency of ontology creation methods, particularly contrasting the user interaction effort in a graphical interface with the coding effort in RDF (resource description framework) and SPARQL (SPARQL protocol and RDF query language). The goal is to objectively evaluate these methodologies' effectiveness in developing the XRInterface ontology for XR environment design. For the purposes of this evaluation, we expertly assume that one line of code corresponds to 3–6 interactions in the graphical interface, providing a baseline for comparing the efficiency of the two approaches.

Table 1 summarizes the evaluation results providing a clear comparison between the graphical interface approach and traditional coding methods in various tasks involved in the development of XRInterface for XR environments. It highlights the efficiency and user-friendliness of using a graphical interface for ontology design and integration in XR interface development.

6.1 Ontology design and rule implementation

Evaluation Method: Comparative analysis of ontology creation using a graphical interface versus traditional coding in Turtle + SPARQL.

Metrics:

- Interactions: The number of user actions (clicks, selections, input) required to complete the task using the graphical interface. Each interaction was manually counted during the ontology creation process.
- Lines of Code (LOC): The number of lines of Turtle + SPARQL code written to achieve the same ontology design and rule implementation. The code was written by an experienced developer familiar with the domain.

Findings:

- The graphical interface approach required 371 interactions for ontology creation.
- The equivalent process in Turtle + SPARQL coding involved 513 lines of code.

Conclusion: The graphical interface approach demonstrates enhanced user-friendliness and efficiency compared to direct coding, especially in complex ontology designs. Based on the 3–6 interactions per line of code metric, the graphical interface offers a more efficient solution for ontology design and rule implementation.

6.2 Creating user profiles in unity

Evaluation Method: Comparison of user profile creation using a visual editor versus writing SPARQL code.

Metrics:

- Interactions: The number of user actions required to create a user profile using the visual editor in Unity.
- Lines of Code (LOC): The number of lines of SPARQL code written to create the same user profile.

Findings:

- Creating user profiles using the visual editor in Unity required 22 interactions.
- Writing SPARQL code for the same task involved 20 lines.

Conclusion: The visual editor in Unity shows similar efficiency to SPARQL coding for user profile creation, with a clear advantage in user-friendliness. Considering the 3–6 interactions per line of code metric, the visual editor proves to be a viable alternative to SPARQL coding.

Table 1 Comparative analysis of the MAXI-XR method using graphical interface and traditional coding

Task	Number of clicks (Graphical Interface)	Lines of code (Coding)
Ontology design and rule implementation	371	513
User profile creation in Unity editor	22	20
Ontology design for XR interface elements	61	68
Integration of ontology in scene development	50	34

6.3 Ontology design for XR interface elements

Evaluation Method: Comparison of ontology creation efforts for XR interface elements using a graphical interface and Turtle + SPARQL coding.

Metrics:

- Interactions: The number of user actions required to create the ontology for XR interface elements using the graphical interface.
- Lines of Code (LOC): The number of lines of Turtle + SPARQL code written to create the same ontology.

Findings:

- Creating the ontology with the graphical interface necessitated 61 interactions.
- The corresponding Turtle + SPARQL code amounted to 68 lines.

Conclusion: The graphical interface approach is slightly more efficient and user-friendly for designing XR interface elements compared to traditional coding. The 3–6 interactions per line of code metric indicates that the graphical interface requires fewer user actions to achieve the same result.

6.4 Ontology-based scene development

Evaluation Method: Analysis of the effort required for integrating ontology in scene development using a visual editor versus RDF triplet coding.

Metrics:

- Interactions: The number of user actions required to integrate ontology concepts into a scene using a visual editor.
- Lines of Code (LOC): The number of lines of RDF triplet code written to achieve the same ontology integration.

Findings:

- Integrating ontology concepts into a scene using a visual editor required 50 interactions.
- Creating RDF triplets for the same purpose involved 34 lines of code.

Conclusion: While RDF coding is more concise for ontology integration, the visual editor offers an intuitive and user-friendly approach, aligning with the objectives of creating adaptable XR environments. Based on the 3–6 interactions per line of code metric, the visual editor provides a more efficient solution for ontology-based scene development.

6.5 Overall assessment

The implementation of the MAXI-XR method in XR interface design for stock market data visualization has shown promising potential. This solution offers adaptability, efficiency, and user-friendliness, especially for users with limited coding skills. The method's effectiveness in using semantic technologies for dynamic interface adaptation allows for a significant level of personalization, which is essential in complex and data-intensive environments like stock market analysis. The solution's scalability and ease of maintenance indicate that it can be applied to a wide range of domains and user groups. Nevertheless, the possibility of enhancing real-time adaptation algorithms and optimizing the graphical interface suggests potential for future development.

However, the current evaluation metrics, focusing solely on interactions and lines of code, provide a limited understanding of the method's efficiency and user-friendliness. To fully assess the MAXI-XR method's effectiveness, further research should include additional metrics such as time taken to complete tasks, perceived difficulty levels, and user feedback. These metrics would provide a more comprehensive picture of the method's performance in different contexts and user groups.

The MAXI-XR method's approach to integrating ontological reasoning within the gaming engine goes beyond traditional programming paradigms. It can set a new standard in user interface design, especially for complex data visualization contexts such as the stock market. This integration simplifies the development process and allows non-programmers to contribute to creating and adapting VR environments. This development in interface creation represents an essential advancement toward XR experiences that are more inclusive, adaptive, and intuitive.

7 Conclusions and future works

In this paper, we proposed the MAXI-XR method for the adaptation of XR interfaces, aimed at enhancing user satisfaction and engagement through semantic technologies, abstract representation of interface features, and a thorough understanding of users' profiles and XR system capabilities. We outlined the implementation of this methodology and provided examples of its adaptations in action.

There are a number of ways that future study could improve enhancing the MAXI-XR method:

- Expanding the representations: We aim to expand SUPR and SSPR by including a wider range of user and system attributes. This will improve the precision and adaptability of our method.
- Incorporation of Additional Inference Rules: By adding more inference rules, we aim to increase the depth and precision of our method's reasoning processes, allowing for more nuanced and accurate adaptations.
- Testing and Validation: In order to evaluate the practicality, efficiency, and durability of our adaptation method, we intend to carry out thorough user-centric testing. This study aims to analyze the compatibility between XR interfaces, user preferences, and system capabilities in real-world scenarios, offering valuable insights into our methodology.
- Methodological Confirmation and Tailoring: These evaluations will be crucial in confirming that our method can effectively tailor XR interfaces to individual user needs and system constraints, ensuring a more personalized and engaging user experience in XR environments.

The MAXI-XR method represents a significant step forward in the field of XR interface design. Its focus on semantic technologies and user-system synergy opens up new possibilities for creating more adaptive, intuitive, and user-friendly XR environments. By continuing to refine and expand this method, we aim to contribute substantially to the advancement of XR technology, making it more user-friendly, engaging, and effective for a diverse range of users and applications.

Acknowledgements The presented research was funded by the Polish National Center for Research and Development under grant number LIDER/55/0287/L-12/20/NCBR/2021.

References

1. Buche, C., Bossard, C., Querrec, R., Chevaillier, P.: Pegase: a generic and adaptable intelligent system for virtual reality learning environments. Int. J. Virtual Real. **9**(2), 73–85 (2010)
2. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A unifying reference framework for multi-target user interfaces. Interact. Comput. **15**(3), 289–308 (2003)
3. De Troyer, O., Kleinermann, F., Pellens, B., Bille, W.: Conceptual modeling for virtual reality. In John Grundy, Sven Hartmann, Alberto H. F. Laender, Leszek Maciaszek, and John F. Roddick, editors, Tutorials, posters, panels and industrial contributions at the 26th Int. Conference on Conceptual Modeling - ER 2007, volume 83 of CRPIT, pages 3–18, Auckland, New Zealand, (2007). ACS
4. Evangelista Belo, J.M., Lystbæk, M.N., Feit, A.M., Pfeuffer, K., Kán, P., Oulasvirta, A., Grønbæk, K.: Auit–the adaptive user interfaces toolkit for designing xr applications. In Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology, pages 1–16, 2022
5. Flotynski, J., Sobocinski, P., Sliwicki, M., Maik, M.: Future and backward exploration of xr environments
6. Flotyński, J., Walczak, K.: Ontology-based representation and modelling of synthetic 3d content: A state-of-the-art review. In Computer Graphics Forum, volume 36, pages 329–353. Wiley Online Library, (2017)
7. Garcia, S., Andujar, M.: Ui design recommendations for multimodal xr interfaces using a collaborative system. In International Conference on Human-Computer Interaction, pages 3–14. Springer, (2023)
8. Heitmayer, M., Russell, M.G., Lahlou, S., Pea, R.D.: An Ontology for Human-Centered Analysis and Design of Virtual Reality Conferencing. In TMS Proceedings 2021, nov 3 2021. https://tmb.apaopen.org/pub/3rbumwgw
9. Horvitz, E.: Principles of mixed-initiative user interfaces. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pages 159–166, (1999)
10. Khan, M.J., Mustafa, K.: Modelling adaptive hypermedia instructional system: a framework. Multimed. Tools Appl. **78**, 14397–14424 (2019)
11. Lacoche, J., Duval, T., Arnaldi, B., Maisel, E., Royan, J.: A survey of plasticity in 3d user interfaces. In 2014 IEEE 7th workshop on software engineering and architectures for realtime interactive systems (SEARIS), pages 19–26. IEEE, (2014)
12. Lee, Y.C.J., Takenaka, B.P.: Extended reality as a means to enhance public health education. Front. Public Health **10**, 1040018 (2022)
13. MacCallum, K.: The integration of extended reality for student-developed games to support cross-curricular learning. Front. Virtual Real. **3**, 888689 (2022)
14. Maik, M., Sobocinski, P., Walczak, K., Strugala, D., Górski, F., Zawadzki, P.: Flexible photorealistic vr training system for electrical operators. In Proceedings of the 27th International Conference on 3D Web Technology, pages 1–9, (2022)

15. Malik, U., Tissen, L.N., Vermeeren, A.P.: 3d reproductions of cultural heritage artifacts: evaluation of significance and experience. Stud. Digit. Herit. **5**(1), 1–29 (2021)
16. Miñón, R., Paternò, F., Arrue, M., Abascal, J.: Integrating adaptation rules for people with special needs in model-based ui development process. Univ. Access Inf. Soc. **15**, 153–168 (2016)
17. Miraz, M.H., Ali, M., Excell, P.S.: Adaptive user interfaces and universal usability through plasticity of user interface design. Comput. Sci. Rev. **40**, 100363 (2021)
18. Pittarello, F., De Faveri, A.: Semantic description of 3d environments: a proposal based on web standards. In Proceedings of the eleventh international conference on 3D web technology, pages 85–95, (2006)
19. Pottle, J.: Virtual reality and the transformation of medical education. Future healthc. j. **6**(3), 181 (2019)
20. Ranosz, J., Leszczyński, C., Kumor, S., Popiel, A., Głowaczewska, J., Garwol, P., Kaczmarek M., Maik, M.: Advancing stem education in primary schools with an integrated system of 3d games. In The 28th International ACM Conference on 3D Web Technology, pages 1–5, (2023)
21. Rumiński, D., Maik, M., Walczak, K.: Visualizing financial stock data within an augmented reality trading environment. Acta Polytech. Hungarica **16**(6), 223–239 (2019)
22. Todi, K., Jonker, T.R.: A framework for computational design and adaptation of extended reality user interfaces. arXiv preprint arXiv:2309.04025, 2023
23. Todi, K., Lafreniere, B., Jonker, T.: Computational adaptation of xr interfaces through interaction simulation. arXiv preprint arXiv:2204.09162, (2022)
24. Walczak, K.: Flex-vr: Configurable 3d web applications. In 2008 Conference on Human System Interactions, pages 135–140. IEEE, (2008)
25. Walczak, K., Flotyński, J., Strugała, D., Rumiński, D., Maik, M., Englert, A., Jenek, T.: Virtual and augmented reality for configuring, promoting and servicing household appliances. In Advances in Manufacturing II: Volume 1-Solutions for Industry 4.0, pages 368–380. Springer, (2019)
26. Yannick, D., Bouyer, G., Otmane, S., Mallem, M.: 3d interaction assistance through context-awareness. In Proceedings of the 19th IEEE Virtual Reality Conference. Orange County: IEEE, pages 103–104, (2012)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mikołaj Maik is an assistant at the Department of Information Technology at the Poznań University of Economics and Business. He focuses his research on virtual and augmented reality technologies in business, multimedia systems, game development, and the Internet of Things. He participates in research projects enhancing training and education through XR. Mikołaj also organizes workshops promoting XR technologies and supervises a student science club focused on game development.

Jakub Flotyński is a researcher in the Department of Information Technology at the Poznań University of Economics and Business, Poland. His research interests are focused on the intersection of virtual, augmented and extended reality (XR), and knowledge representation technologies, including the semantic web, description logics, and logic programming. He is the author or co-author of 60 research papers and has participated in multiple research and industrial projects in these fields.

Krzysztof Walczak heads the Department of Information Technology and the VR/AR Laboratory at the Poznan University of Economics and Business. His research interests include virtual and augmented reality, human-computer interaction, multimedia databases, and artificial intelligence. He has led numerous research and industrial projects in these areas. Prof. Walczak has authored or co-authored two books and over 150 scientific articles published in journals, monographs, and proceedings of international conferences. He is a longstanding board member of the EuroXR Association.