A Framework for Context-Dependent Augmented Reality Applications Using Machine Learning and Ontological Reasoning

Fabian Muff[1], Hans-Georg Fill[1]

[1]*Digitalization and Information Systems Group*
Department of Informatics
University of Fribourg, Switzerland

Abstract
The concept of augmented reality permits to embed virtual objects and information within the real context of a user. This is achieved using various sensors to assess the current state of the environment and thus derive the artificially generated information for the user through visual means. For determining the current situation of a user based on sensor data and deriving according actions for information display, we describe a framework that combines machine learning services for object recognition with ontological reasoning. For demonstrating its feasibility, the framework has been prototypically implemented using the Microsoft HoloLens2 AR device and applied to a use case in the domain of work safety measures. Thereby we revert to business process models that have been annotated with concepts from an ontology for letting users specify the situations and actions in work safety scenarios, which can subsequently be processed using objects identified in the real environment of the user and classified based on the concepts in the ontology.

Keywords
Augmented Reality, Machine Learning, Metamodeling, Ontology, Reasoning

1. Introduction

In augmented reality (AR) the user is embedded in a combination of the real world and a virtual environment that is enriched with graphical content that does not exist in reality [1]. One big advantage of AR applications is the perception of the user's context based on different sensors [2]. This is achieved through analyzing and classifying the sensors' information, for example, for determining a user's location [3]. However, the problem of most current AR applications is that they are specifically developed for one use case and only work in an exactly predefined setup.

As a resolution, artificial intelligence (AI) may be used for a more flexible setup that automatically determines a user's context. Thereby, machine learning (ML) approaches can help for tasks

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022), Stanford University, Palo Alto, California, USA, March 21–23, 2022.
fabian.muff@unifr.ch (F. Muff); hans-georg.fill@unifr.ch (H. Fill)
https://www.unifr.ch/inf/digits/ (F. Muff); https://www.unifr.ch/inf/digits/ (H. Fill)
0000-0002-7283-6603 (F. Muff); 0000-0001-5076-5341 (H. Fill)

such as object recognition and ontological reasoning enables the inference of context-dependent actions. These actions then result in the display of information in the user's environment using AR devices.

The use of either machine learning or reasoning has already been explored in the AR community, e.g. [4, 5, 6, 2]. However, recent developments in artificial intelligence propose the combination of these approaches, including the involvement of humans in the sense of hybrid intelligence [7]. In the following we will therefore explore how such combinations of machine learning and ontological reasoning provide benefits for AR applications. It is expected that this could lead to a higher convergence between the real world and the cyberspace by collecting huge amounts of information via sensors, analyzing it through AI, and then feeding it back to humans in their current context [8].

As a running example, imagine a scenario where a human actor performs work in a manufacturing process. For complying with workplace safety, the user shall be informed about necessary safety measures - e.g., to put on ear protection in loud environments. With the help of AR we can derive environment information via sensors, identify objects and environment states using machine learning and classify this information through reasoning. Thereby it is reverted to a state and actions ontology that represents the knowledge on workplace safety situations and measures. This knowledge is derived from existing enterprise models determining the possible situations and necessary safety measures. As a result, the AR device can display warnings in dangerous work situations and guide the user how to take safety measures.

For addressing these challenges, we will present in the following a framework for combining machine learning and ontological reasoning for augmented reality applications. In contrast to previous approaches - such as for example described by Krings et al. [9] - we will however propose a platform-independent approach using most recent technologies for web-based AR applications that integrates machine learning and ontological reasoning.

The remainder of the paper is structured as follows: In Section 2 we will briefly discuss the foundation of augmented reality and context inference in AR. In Section 3, we will introduce a framework we developed for context-dependent augmented reality applications and present a prototypical implementation. This is followed by the discussion of a use case in Section 4. The paper ends with an evaluation of the benefits and pitfalls of such an approach in Section 5 and a conclusion with an outlook on further work in Section 6.

2. Foundations

In this section we briefly discuss the foundations of AR and context inference to achieve a common understanding of these terms and give an overview of the related work in these areas.

2.1. Augmented Reality

Augmented reality is a technology that allows to overlay computer-generated virtual images with the real world [10]. A widely-used definition of AR comes from Azuma [1]. He describes AR as a technology that combines the real world and virtual imagery, is real time interactive and can register virtual images with the real environment.

As described in [11], there are some characteristics that all AR environments have in common. To make AR possible, we require an *electronic display device*, e.g., a smartphone or a head mounted display (HMD). Further, the devices have to dispose of different sensors for detecting the environment. This includes for example position or motion sensors. In any case they need to have a display sensor for representing visual information. If the AR device is a screen device, a simulacrum of the real world must be visualized on the display, since the real world is not directly visible by the user. If the AR device has a transparent display, the real world is directly visible for the user and thus must not be visualized again by the device. Additionally, virtual representations like 3D objects or other information can be visualized on the display. From the user's perspective this virtual information thereby merges with the real world.

2.2. Context Inference in Augmented Reality

A big advantage of AR applications is the possibility to infer information about the user's environment by an AR device and to display additional information to the user based on the real environment. Applications that allow such functionalities are called *Pervasive Augmented Reality* (PAR) or *Context-Aware Augmented Reality Applications*. Such functionality can be achieved in different ways. One option is to predefine the objects that shall be recognized in the AR application via image recognition approaches [2, 4]. Further, one can use the user's *index of pupillary activity* for cognitive load estimation and adaption of the degree of detail based on the workload [12], or perform a search in a knowledge base using acquired sensor data [13].

A framework for creating context-aware augmented reality applications has been presented by Krings et al. [9]. The framework provides a reusable approach for easing the development of context-dependent AR applications for mobile phones by describing the base structures to enable context-aware adaptions of AR content. Further, there are approaches for context-aware augmented reality that use either machine learning or knowledge reasoning approaches [10, 14, 15]. However, all these approaches are platform-dependent and do not combine the concepts of machine learning and knowledge reasoning. To the best of our knowledge there is no approach yet that combines knowledge engineering, ML, knowledge reasoning and AR in one process.

There are some approaches of context-aware semantic web approaches in the area of the *Internet of Things* (IoT) as well [16, 17]. Since AR devices can be seen as IoT devices, this research area must also be considered.

3. Framework for Context-Dependent AR Applications

For realizing context-dependent AR applications, we developed a framework that contains the concepts of *machine learning*, *ontologies*, and *reasoning*. As proposed in [18], there are different patterns on how to combine the concepts of machine learning and knowledge reasoning. Thereby, the two data structures *model-free data* and *model-based data* are distinguished, as well as the two algorithmic components *context reasoning* and *machine learning*. Since the information received from the sensors of AR devices is mostly in a raw format and must be further processed to infer useful context information, we classify this input as *model-free data*. This data can be processed, e.g., through classifying the *model-free data* to *model-based data* by using machine learning. Additionally, one can use *model-based data* as additional input for the ML process to narrow down the classification space for the *model-based* output data.

After the sensor data has been classified, the given information can be used to infer further actions or propositions for the user. This can be done by using ontologies, i.e., *knowledge reasoning*. Ontologies enable knowledge sharing, knowledge reuse, and logic-based reasoning and inference [19]. Since the output from the ML process is *model-based data*, we can apply a second pattern described in [18]. This pattern takes *model-based data* as input for the *knowledge reasoning* process. As output there is again *model-based data* with additional inferred information. By putting two patterns together, we obtain a new design pattern called "Learning an intermediate abstraction for reasoning" [18]. By combining this pattern with AR, we can create AR applications that enable context-based, adaptable AR environments. To the best of our knowledge there is no framework available yet, that combines such AI-based object recognition with ontology-based reasoning for providing the user with additional context-based information. The components of the framework and the according steps in the pipeline are illustrated in Figure 1 and correspond to the mentioned pattern in [18].

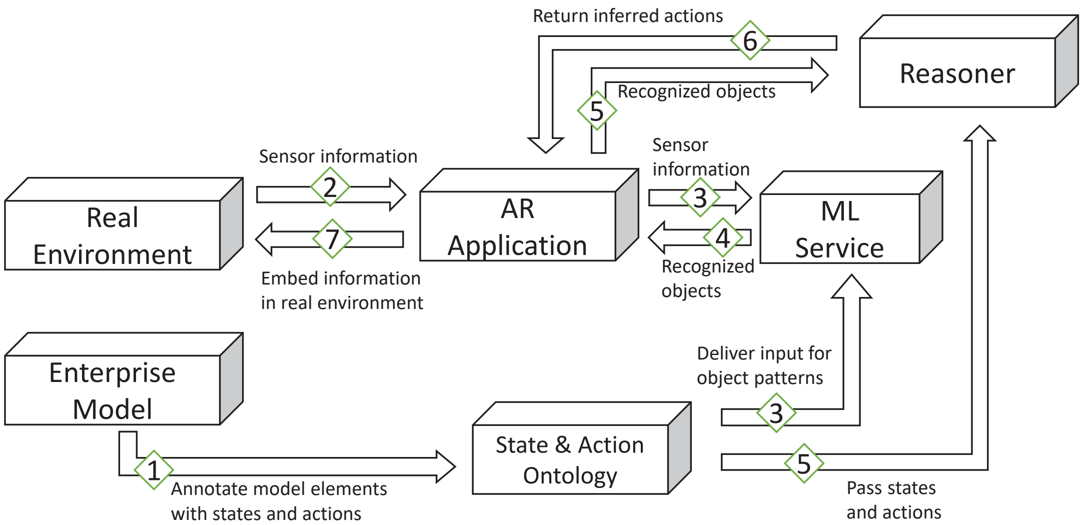

Figure 1: Framework for Context-Dependent AR Applications showing Data collection and Information processing Using Seven Steps

In the framework we consider the real environment and a mobile AR application. For describing the business environment of the AR application, we assume the existence of an enterprise model - e.g., a business process model - that is annotated with states in the form of object patterns and actions described by an ontology (1) – see Figure 3 [20]. Thereby, the information on the context of the user and necessary actions is formally represented. This information will be used later in the process for facilitating the recognition of objects and the inference of actions via a reasoner. When starting the application, the real environment is perceived by the various sensors of the AR device (2). This sensor information, as well as the object patterns of the ontology are then directed to a machine learning service (3). There, objects are recognized based on the data provided by the ontology. The recognized objects are returned to the AR application (4). Then, the recognized objects and the states from the ontology are forwarded to the reasoner (5) for inferring actions. The inferred actions are then sent back to the AR application (6). Based on the inferred actions, the AR application can finally embed visual information into the real environment (7).

3.1. Technical Realization

For evaluating the feasibility of the developed framework, we implemented it as a prototype. For this purpose, we used state-of-the-art web technology to set up a mobile, platform-independent AR environment. We used the JavaScript WebGL-based visualization framework *THREE.js*[1] in combination with the *WebXR Device API*[2]. This combination enables the creation of platform-independent applications.

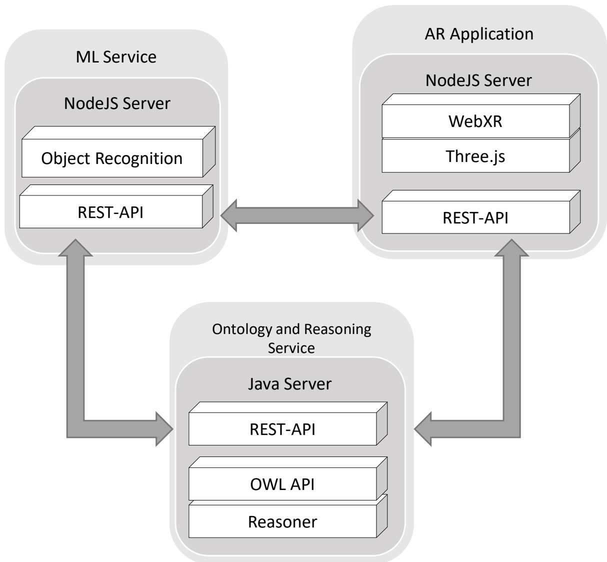

Figure 2: Technical Architecture of the Prototypical Implementation with the Three Main Components ML Service, AR Application and Ontology and Reasoning Service

A requirement of the application is the recognition of objects in the real world. Since the *WebXR Device API* does not yet contain machine learning based image recognition, we simulated the object recognition by using marker patterns that provide information about the recognized objects. In a next iteration of the implementation this will be replaced by an ML service for object recognition such as *Azure Object Detection*[3] or *AWS Rekognition*[4]. Thereby, images of the real world are sent to a cloud-based ML service to recognize objects in an image or other sensor data. Such a machine learning object recognition service can return many different objects.

[1]https://threejs.org/docs/
[2]https://www.w3.org/TR/webxr/
[3]https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/concept-object-detection
[4]https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html

Many of them are not necessarily useful for our application. Therefore, we must restrict the possible set of recognizable objects. This is done by defining an ontology of the states and actions necessary for the situation we want to cover by the application. For the creation and processing of the ontology, we used the Web Ontology Language (OWL)[5] and the Java-based OWL-API[6].

In our prototype, a marker refers always to an ontology individual and its according type definition. The information about the different markers and its visual representation is currently stored in a configuration file. After assigning the according types to the ontology individuals, the reasoning is conducted using the HermiT reasoner[7] for inferring further states and actions.

As shown in Figure 2, the different components are independent from each other. The *Marker pattern* component could be replaced easily by another image recognition component, e.g., a cloud ML service. The different components communicate via REST-APIs. To test the proposed prototype, we apply it in the following section to a use case and thereby illustrate the necessary steps in more detail.

4. Use Case for a Context-Dependent AR Application

As a sample scenario let us imagine a carpentry that manufactures different wood products. We assume here the business process has been described in BPMN notation as illustrated in Figure 3. In a first step we can look at the real-world environment for this process. For example, we can look at the task *start saw* in the process of Figure 3. We know that there must be a saw, that the saw produces a loud sound or that the temperature of the saw can be of importance when arriving at this task. Further, we know that the person who starts the saw stands still. According to safety measures, an expert knows which personal protective equipment is obligatory and what hazards and risks occur in this situation.

Based on that knowledge, we can annotate the process model with information on the context and safety measures [21]. For this purpose, we defined annotations with the following concepts from a specifically developed ontology: *Machine*, *PersonalProtectiveEquipment*, *PersonState*, *Sound*, *Temperature* and *Tool* as *Scene Annotations*, and *Risk*, *Hazard*, *Action* and *State* as *Action Annotations* – see the annotations in Figure 3. A similar approach for annotating workflows like this has been presented in [22].

Thereby, *Scene Annotations* serve as input for the AR application for determining the context of a scene. With *Action Annotations* we derive actions that are executed by the AR application based on given scenes. These annotations are modeled formally so that they can serve as a basis for reasoning over the concepts. For example, we define in the ontology that a circular saw is a subclass of a machine and that it has a specific sawing sound, which constitutes a hazard for the user since it is very noisy. Due to the formal specification ontology, we can later infer action types based on situational information. For example, whenever there is some loud noise, the user shall wear ear protection. Therefore, we can derive the action to inform the user to wear ear protection.

---

[5]https://www.w3.org/OWL/
[6]https://github.com/owlcs/owlapi
[7]http://www.hermit-reasoner.com/

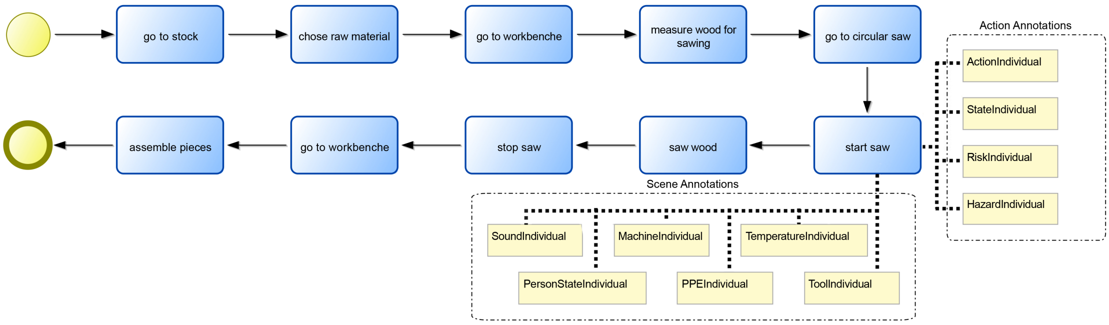

Figure 3: Business Process Model for the Use Case annotated with Concepts from the Situation and Actions Ontology.

The core *classes* and *object properties* of the ontology that are relevant for the example are shown in the diagram in Figure 4. For each annotated concept we assume the existence of an individual with the according class type definition and properties assigned to it. In the next step we will show the application of this information during the use of the AR application.

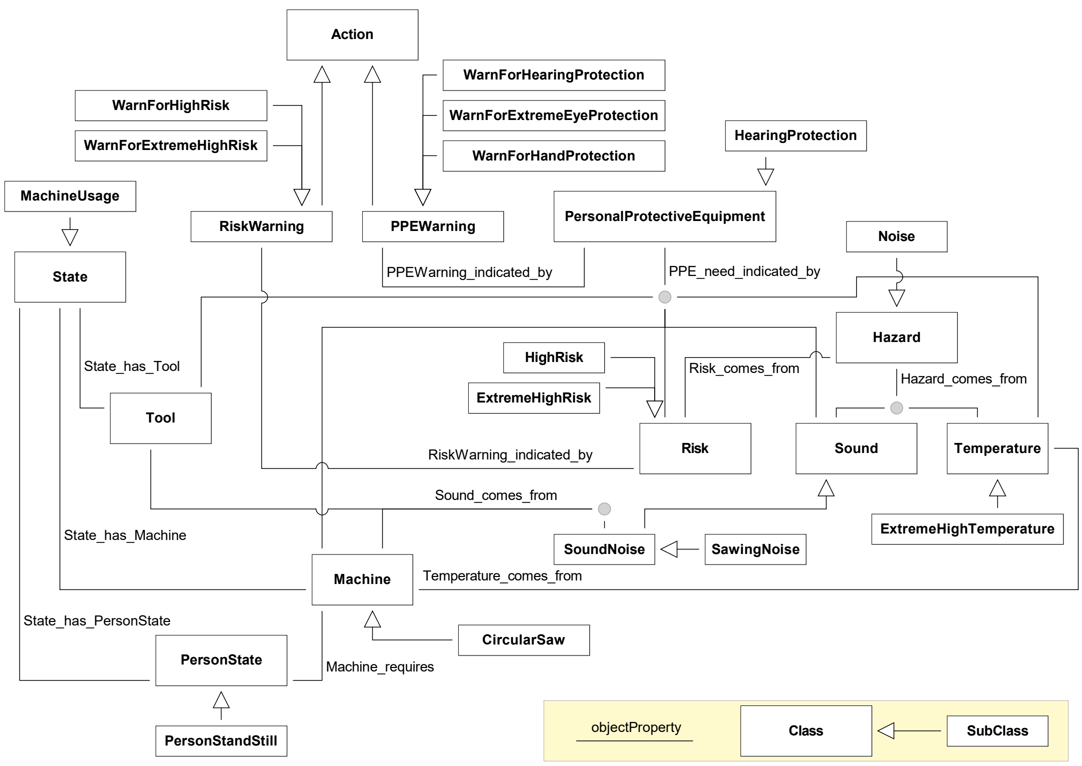

Figure 4: Visualization of the ontology main classes and object properties used for the use case

Let's imagine that a user performs some tasks of the process shown in Figure 3. As the process is not automated, the AR application does not yet know which task the user currently

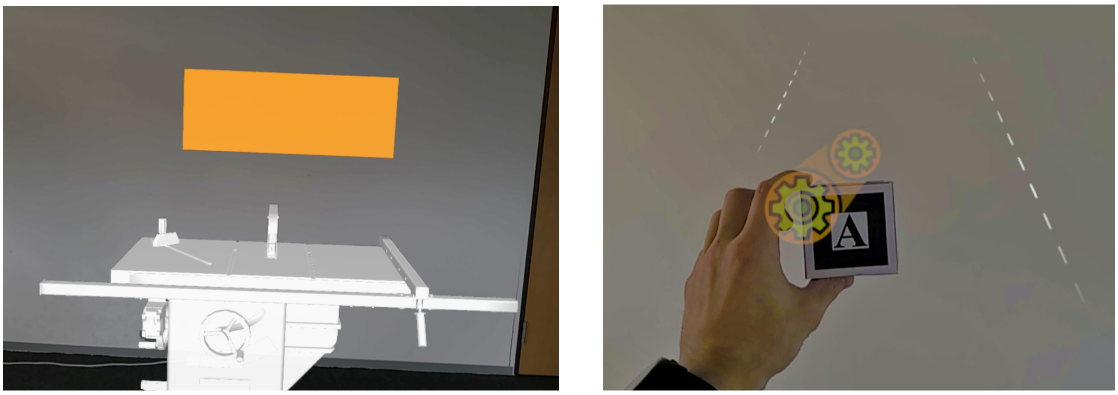

Figure 5: Initial scene with a circular saw and an information pane on top of the object (left) and an example of the marker recognition (right)

performs. With the proposed framework that aims for an automatic derivation of the context, it is not necessary to exactly predefine a situation, e.g., through stating that the AR application only works for *process step 6*. Rather, different pieces of information are used for inferring the current situation and state and for deriving according actions. Therefore, it can be dynamically assessed on which task the user is working. For getting the required information, we interpret the camera stream and the acceleration sensor data of the AR device. With the help of ML-based object recognition in combination with the predefined ontology, we can detect a circular saw and analyze this situation further – see the left side of Figure 5. For facilitating the process in a first implementation step, we replaced the ML-based object recognition with marker tracking. A marker stands for a particular object or state and can be more easily recognized than real objects. Therefore, we scan the markers to give the application the according information – see right side of Figure 5.

In this use case we provide two markers which serve as a proxy for indicating to the application that there is a circular saw, and that the user does not move. For signaling to the user that the marker has been recognized, we visualize in the prototype a representative image of the information contained in the marker, e.g., gear wheels for representing a machine. Now, the application assigns the recognized objects as individuals to a type from the ontology, e.g., the individual *Machine* to the type *CircularSaw* and the individual *PersonState* to the type *PersonStandStill*. This information is then passed to the reasoner, which infers the following information: Since there is a circular saw and the person stands still, we know that the individual State is of the type *MachineUsage*.

In the ontology we have the two object properties (1) and (2). Thus, the inferred state type can be described formally as in (3):

$$ObjectPropertyAssertion(:State\_has\_Machine\ :s\ :m) \tag{1}$$

$$ObjectPropertyAssertion(:State\_has\_PersonState\ :s\ :ps) \tag{2}$$

$$CircularSaw(m) \wedge PersonStandStill(ps) \wedge State(s) \rightarrow MachineUsage(s) \tag{3}$$

As a circular saw is a sawing machine, we can further infer via the ontology that the individual *Sound* is assigned to the type *SawingNoise*. That again is defined through object properties in the

Figure 6: Example of the scene when the information *PersonStandStill* and *SawingMachine* has been given to the application (left) and with the additional information *ExtremeHighTemperature* (right)

ontology – see Figure 4. Sawing noise indicates that the individual *Hazard* is of the type *Noise* and with that we can infer that the individual *Risk* is of the type *HighRisk*. Further, the individual *Sound* of the type *SawingNoise* indicates that the individual *PersonalProtectiveEquipment* is of the type *HearingProtection*. This infers again a type *WarnForHearingProtection* for the individual *Risk*. As shown on the left side in Figure 6, the warnings inferred by the application are then displayed as text in an additional object above the machine in the real world.

We can now extend the use case by assigning the individual *Temperature* the type *Extreme-HighTemperature* by scanning the according marker in the AR application. The reasoner will then try to infer further information via the ontology. Thereby, the individual *Risk* is inferred as *ExtremeHighRisk* and there are additional warnings for wearing eye protection and hand protection – see right side of Figure 6.

Although the use case is strongly simplified, it already illustrates how a user can be supported in a work process and how it is possible to warn a user for potentially dangerous situations through AR with the help of the "Learning an intermediate abstraction for reasoning" pattern.

5. Evaluation

For a first evaluation of the proposed framework, we discuss in the following the benefits and shortcomings that have been identified through the prototypical implementation and the application to a use case.

One of the advantages of the proposed framework is its flexibility. Since the *ML Service*, the *Ontology and Reasoning Service* and the *AR Application* are modular, the framework is extendable and adaptable. In addition, as the framework is platform independent, it can be used for any AR device supporting the WebXR standard, e.g., a Microsoft HoloLens2 as well as state-of-the-art smartphones and tablets.

Another advantage of this framework is that it is generic. The framework is not coupled to a specific use case, but it can be used for a multitude of application areas. Since the ontology is not directly derived from a task or situation, but from different states and objects, one can use the framework for any situation in which such states and actions can be defined formally. Further, the approach could be integrated with context-aware workflow management systems.

A drawback of the framework is its performance. Since the application has a chain-like workflow (see Figure 3), the different steps in the process are processed one after the other. This means that each step must be processed in a very short time. In the prototype shown above, the ontology is rather small and the time to reason over it is short, i.e., approximately 20ms for an individual type assertion with the update of the ontology reasoning and 5ms to get the types of a given individual when running the application in a local network. Therefore, there are no delays noticeable for the user. However, if the ontology becomes very big or the network connection is not adequate, the application could become slow, which is a critical issue for AR applications. Further, the knowledge acquisition process for modeling the ontologies requires a high effort. If the domain experts have no or only little IT knowledge in ontology engineering this might be a problem. The dependencies between the different classes in very complex ontologies might be a problem as well. If terminological reasoning is not sufficient, rule-based reasoning could be used in addition [23].

Another limitation is the current state of technology. Head-mounted AR devices, e.g., the Microsoft HoloLens2, are still heavy and not very comfortable to wear over a long time. Mobile devices such as smartphones require a free hand to use them, which is not possible in all situations. This limitation, however, may be eliminated in the near future by technological progress, e.g., by using lightweight AR glasses or even AR contact lenses [24].

6. Conclusion and Outlook

The goal of our research was to develop a flexible, platform-independent framework for combining *AR*, *ML* and *ontological reasoning* in a hybrid way to facilitate the development of context-aware augmented reality applications. We focused thereby on open, state-of-the-art web technology to make the framework platform-independent and the different modules flexible and adaptable. We showed the use of the framework in a prototypical application for a carpentry process for informing the user about safety measures. Finally, we evaluated the framework based on this use case.

Future research will entail solving the described limitations by testing the framework with more complex use cases including bigger ontologies. Further, we will include an AI *object recognition* service in the architecture to complete the whole process. Moreover, the process shall be extended to allow the prediction of likely next activities in a process. Finally, it is planned to integrate the approach with on-going effort for AR-based enterprise modeling to enhance existing modeling approaches with AR functionalities [25].

References

[1] R. T. Azuma, A survey of augmented reality, Presence Teleoperators Virtual Environments 6 (1997) 355–385.

[2] E. Yigitbas, I. Jovanovikj, S. Sauer, G. Engels, On the Development of Context-Aware Augmented Reality Applications, volume 11930 of *Lecture Notes in Computer Science*, Springer International Publishing, 2020, p. 107–120. URL: http://link.springer.com/10.1007/978-3-030-46540-7_11. doi:10.1007/978-3-030-46540-7_11.

[3] K. Lee, J. Lee, M.-P. Kwan, Location-based service using ontology-based semantic queries: A study with a focus on indoor activities in a university context 62 (2017) 41–52. URL: https://doi.org/10.1016/j.compenvurbsys.2016.10.009. doi:10.1016/j.compenvurbsys.2016.10.009.
[4] S. Akbarinasaji, E. Homayounvala, A novel context-aware augmented reality framework for maintenance systems, J. Ambient Intell. Smart Environ. 9 (2017) 315–327. URL: https://doi.org/10.3233/AIS-170435. doi:10.3233/AIS-170435.
[5] H. Kim, T. Matuszka, J.-I. Kim, J. Kim, W. Woo, An ontology-based augmented reality application exploring contextual data of cultural heritage sites, in: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, 2016, p. 468–475. URL: http://ieeexplore.ieee.org/document/7907506/. doi:10.1109/SITIS.2016.79.
[6] A. Katsaros, E. Keramopoulos, Farmar, a farmer's augmented reality application based on semantic web, in: 2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), 2017, p. 1–6. doi:10.23919/SEEDA-CECNSM.2017.8088230.
[7] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, A. ten Teije, Modular design patterns for hybrid learning and reasoning systems, Applied Intelligence 51 (2021) 6528–6546. URL: https://doi.org/10.1007/s10489-021-02394-3. doi:10.1007/s10489-021-02394-3.
[8] A. Deguchi, C. Hirai, H. Matsuoka, T. Nakano, K. Oshima, M. Tai, S. Tani, What Is Society 5.0?, Springer Singapore, Singapore, 2020, pp. 1–23. URL: https://doi.org/10.1007/978-981-15-2989-4_1. doi:10.1007/978-981-15-2989-4_1.
[9] S. Krings, E. Yigitbas, I. Jovanovikj, S. Sauer, G. Engels, Development framework for context-aware augmented reality applications, in: Companion Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '20 Companion, Association for Computing Machinery, New York, NY, USA, 2020. URL: https://doi.org/10.1145/3393672.3398640. doi:10.1145/3393672.3398640.
[10] F. Zhou, H. B.-L. Duh, M. Billinghurst, Trends in augmented reality tracking, interaction and display: A review of ten years of ismar, in: 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE, 2008, p. 193–202. URL: http://ieeexplore.ieee.org/document/4637362/. doi:10.1109/ISMAR.2008.4637362.
[11] F. Muff, H.-G. Fill, Towards embedding legal visualizations in work practices by using augmented reality (2021). URL: https://doi.org/10.38023/e40dcea9-9724-4318-bdca-52ce1cb04e68. doi:10.38023/e40dcea9-9724-4318-bdca-52ce1cb04e68.
[12] D. Lindlbauer, A. M. Feit, O. Hilliges, Context-aware online adaptation of mixed reality interfaces, in: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, ACM, 2019, p. 147–160. URL: https://dl.acm.org/doi/10.1145/3332165.3347945. doi:10.1145/3332165.3347945.
[13] D. Rumiński, K. Walczak, Semantic model for distributed augmented reality services, in: Proceedings of the 22nd International Conference on 3D Web Technology, ACM, 2017, p. 1–9. URL: https://dl.acm.org/doi/10.1145/3055624.3077121. doi:10.1145/3055624.3077121.

[14] J. Grubert, T. Langlotz, S. Zollmann, H. Regenbrecht, Towards pervasive augmented reality: Context-awareness in augmented reality, IEEE Transactions on Visualization and Computer Graphics 23 (2017) 1706–1724. doi:10.1109/TVCG.2016.2543720.

[15] R. Hervás, A. Garcia-Lillo, J. Bravo, Mobile augmented reality based on the semantic web applied to ambient assisted living, in: J. Bravo, R. Hervás, V. Villarreal (Eds.), Ambient Assisted Living - Third International Workshop, IWAAL 2011, Held at IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011. Proceedings, volume 6693 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 17–24. URL: https://doi.org/10.1007/978-3-642-21303-8_3. doi:10.1007/978-3-642-21303-8_3.

[16] M. Ruta, F. Scioscia, G. Loseto, A. Pinto, E. D. Sciascio, Machine learning in the internet of things: A semantic-enhanced approach, Semantic Web 10 (2019) 183–204. URL: https://doi.org/10.3233/SW-180314. doi:10.3233/SW-180314.

[17] O. B. Sezer, E. Dogdu, A. M. Özbayoglu, Context-aware computing, learning, and big data in internet of things: A survey, IEEE Internet Things J. 5 (2018) 1–27. URL: https://doi.org/10.1109/JIOT.2017.2773600. doi:10.1109/JIOT.2017.2773600.

[18] F. van Harmelen, A. t. Teije, A boxology of design patterns for hybrid learning and reasoning systems, Journal of Web Engineering 18 (2019) 97–124. doi:10.13052/jwe1540-9589.18133, arXiv: 1905.12389.

[19] X. Wang, D. Zhang, T. Gu, H. Pung, Ontology based context modeling and reasoning using owl, in: IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second, 2004, pp. 18–22. doi:10.1109/PERCOMW.2004.1276898.

[20] H.-G. Fill, SeMFIS: A flexible engineering platform for semantic annotations of conceptual models, Semantic Web 8 (2017) 747–763. URL: https://doi.org/10.3233/SW-160235. doi:10.3233/SW-160235.

[21] H.-G. Fill, Using semantically annotated models for supporting business process benchmarking, in: Perspectives in Business Informatics Research, Springer, 2011, pp. 29–43.

[22] M. Wieland, H. Schwarz, U. Breitenbücher, F. Leymann, Towards situation-aware adaptive workflows: Sitopt — a general purpose situation-aware workflow management system, in: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 2015, pp. 32–37. doi:10.1109/PERCOMW.2015.7133989.

[23] B. Pittl, H. Fill, A visual modeling approach for the semantic web rule language, Semantic Web 11 (2020) 361–389. doi:10.3233/SW-180340.

[24] M. Wiemer, C. Chair, Mojo vision: Designing anytime, anywhere ar contact lenses with mojo lens, in: SPIE AVR21 Industry Talks II, volume 11764 of *SPIE AR, VR, MR Industry Talks II*, SPIE, 2021-4-1.

[25] F. Muff, H.-G. Fill, Initial concepts for augmented and virtual reality-based enterprise modeling, in: R. Lukyanenko, B. M. Samuel, A. Sturm (Eds.), Proceedings of the ER Demos and Posters 2021 co-located with 40th International Conference on Conceptual Modeling (ER 2021), St. John's, NL, Canada, October 18-21, 2021, volume 2958 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 49–54. URL: http://ceur-ws.org/Vol-2958/paper9.pdf.