

A Digital Companion Architecture for Ambient Intelligence

KIMBERLY GARCIA, University of St.Gallen, Switzerland
JONATHAN VONTOBEL, University of St.Gallen, Switzerland
SIMON MAYER, University of St.Gallen, Switzerland

Ambient Intelligence (AmI) focuses on creating environments capable of proactively and transparently adapting to users and their activities. Traditionally, AmI focused on the availability of computational devices, the pervasiveness of networked environments, and means to interact with users. In this paper, we propose a renewed AmI architecture that takes into account current technological advancements while focusing on *proactive* adaptation for assisting and protecting users. This architecture consist of four phases: *Perceive, Interpret, Decide, and Interact*. The AmI systems we propose, called Digital Companions (DC), can be embodied in a variety of ways (e.g., through physical robots or virtual agents) and are structured according to these phases to assist and protect their users. We further categorize DCs into *Expert DCs* and *Personal DCs*, and show that this induces a favorable separation of concerns in AmI systems, where user concerns (including personal user data and preferences) are handled by Personal DCs and environment concerns (including interfacing with environmental artifacts) are assigned to Expert DCs; this separation has favorable privacy implications as well. Herein, we introduce this architecture and validate it through a prototype in an industrial scenario where robots and humans collaborate to perform a task.

CCS Concepts: • **Human-centered computing** → **Personal digital assistants**; *Empirical studies in ubiquitous and mobile computing*; **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Scene understanding**.

Additional Key Words and Phrases: architecture, ambient intelligence, digital companion systems, scene graph generation algorithm, connected devices, knowledge graph, mixed reality, industrial environments.

ACM Reference Format:
Kimberly Garcia, Jonathan Vontobel, and Simon Mayer. 2024. A Digital Companion Architecture for Ambient Intelligence. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 2, Article 66 (June 2024), 26 pages. https://doi.org/10.1145/3659610

1 INTRODUCTION

In the 2000s, the Ambient Intelligence (AmI) research field flourished thanks to the growing availability of sensors in everyday spaces (e.g., office buildings, houses, etc.), the miniaturization of electronics, the permeation of everyday environments with (wireless) communication technology, the increasing amount of available data, and the enthusiasm for creating spaces capable of adapting to their users. Such enthusiasm was powered by Weiser's vision of disappearing technology [69]. Leading to Zelkha et al. [72] minting the term *Ambient Intelligence* which focused on the future of computational devices at home. A few years later, the Information Society and Technology Advisory Group at the European Commission [21] proposed several scenarios for AmI. In these scenarios, a user navigates seamlessly from one personalized smart environment to another. The environments facilitate co-located and remote interactions with other users; they provide assistance and make arrangements for users in a proactive but non-invasive manner; communicating when necessary, and at times making decisions on behalf of a user

Authors' Contact Information: Kimberly Garcia, kimberly.garcia@unisg.ch, University of St.Gallen, St. Gallen, Switzerland; Jonathan Vontobel, jonathan.vontobel@student.unisg.ch, University of St.Gallen, St. Gallen, Switzerland; Simon Mayer, simon.mayer@unisg.ch, University of St.Gallen, St. Gallen, Switzerland.



This work is licensed under a Creative Commons Attribution International 4.0 License.
© 2024 Copyright held by the owner/author(s).
ACM 2474-9567/2024/6-ART66
https://doi.org/10.1145/3659610

based on the preferences and habits learned through observation. Obstacles such as siloization of devices and systems in such intelligent environments would have been already overcome. Spaces would adapt and provide adequate conditions for users to carry on their business, social, leisure, and routine activities. In 2007, Augusto [4] proposed to focus on intelligence to develop systems capable of assisting people in a *sensible* way. Later on, Cook et al. [17] specified further the term *sensible* by analysing the common characteristics of the research in the field, concluding that the AmI community was set to create *physical spaces that are sensitive, responsive, adaptive, transparent, ubiquitous, and intelligent*. Even though the term *Ambient Intelligence* has decreased in popularity in the recent years, several elements of these early visions have become part of our lives, such as commercial digital assistants (e.g., *Siri* and *Alexa*) that resemble the AmI's *Butler* vision, as well as assisted living [14], and building automation [52] applications. In a recent survey, Dunne et al., [22] recognize that some of the major achievements over the past years in AmI have been enabled by the Internet of Things's (IoT) sensing capabilities, as well as machine learning algorithms for activity, gesture, speech, and emotion recognition. On this basis, Dunne et al. proposes an updated definition of AmI: *technologies that combine the Internet of Things (IoT) and Artificial Intelligence (AI) to create adaptive environments*. We argue that it is time for a modern AmI architecture that takes advantages of the current advancements of technologies capable of injecting fine grain understanding to data obtained from IoT devices; better understanding of a user and their current situation will allow AmI systems to make better decisions on how to support their users.

Thus, as the main contribution of this paper, we propose a software architecture that situates Digital Companions (DCs)—smart agents that operate in virtual and physical environments to assist and protect their users [28]—as AmI systems specialized in assisting and protecting users. In our architecture, current advancements in connected devices, computer vision, symbolic knowledge representation, and even generative AI, are taken advantage of to develop modern AmI systems. Our proposed software architecture enriches the traditional agent decision cycle [71] by proposing four phases: *Perceive*, *Interpret*, *Decide*, and *Interact*. It furthermore proposes the distinction between *Expert* and *Personal* DCs: Expert DCs are in charge of managing a specific AmI environment, while personal DCs are concerned with a specific user—it knows their preferences and has access to their (personal) data. In our architecture, Personal and Expert DCs work together to create *physical spaces that are sensitive, responsive, adaptive, transparent, ubiquitous, and intelligent* for users. We validate our proposed architecture through an implementation of a system of a Personal and an Expert DC that operate in an industrial scenario in which robots collaborate with humans to accomplish a transportation task and where interaction is mediated through a Mixed Reality (MR) application. We evaluated the implemented DCs through a user study with 13 participants, which indicate an overall positive response to the MR application in relationship to how well they understood what was happening in the AmI environment, and the effectiveness of the MR application.

This paper is structured as follows: We contextualize our architecture and implementation with respect to related work in Section 2. We then introduce our proposed architecture for modern AmI systems in Section 3, and describe its validation through an implementation in an industrial scenario in Section 4. The results from our usability studies are presented in Section 4.6. We validate the generalizability of the proposed architecture in Section 5; and we present our conclusions and future work in Section 6.

2 RELATED WORK

A DC is capable of assisting and protecting a user by adapting the environment through connected devices, and it can enhance human capabilities by leveraging the computing capabilities of machines [28]. Even though the term *Digital Companion* has gained traction over the last years [13, 27, 43, 53, 58, 65], similar concepts have been proposed across several fields. Thus, we discuss relevant architectures for DCs brought forward in three main areas: Companion Systems (Section 2.1), Intelligent Assistants (Section 2.2), and Companion Robots (Section 2.3).

2.1 Companion Systems Architectures

As part of a multi-disciplinary research project on Companion Technology, Wendemuth et al. propose the creation of companion systems that demonstrate *individuality, adaptability, availability, cooperativeness and trustworthiness* [70]. To achieve these properties, they propose a general architecture [38] for Companion Systems consisting of four types of components: *Recognition* to sense the user and the environment, *Knowledge* to contextualize the recognized data, *Planning, Reasoning, and Decision* to create an adapted response based on the contextualized data, and *Interaction* to establish an adaptive dialog with a user. Based on this architecture, Bercher et al. implemented a Companion System for setting up a complex home entertainment system and specialize the general architecture with a focus on hierarchical planning [8]. This helps users achieve tasks by providing them with situated adapted step-by-step instructions and explanations. The specialized architecture consist of six components: *Planning, Dialog, Interaction Management, Knowledge Base, Sensors, and User Interface*. During an explicit interaction initiated by a user, the system utilizes the *Planning* component to compute instructions, which are then processed by the *Dialog* component to personalize the instructions for a specific user. Next, the *Interaction Management* component selects a means to communicate the instructions, which are then delivered through a *User Interface*. All these components use a *Knowledge Base* to perform their tasks. Some implicit interactions are supported by incorporating the input of the *Sensor* component, which could modify a computed plan. One of the most recent works from this research team is *Robert* [7], a Do-It-Yourself (DIY) companion that, given a formal model of a DIY project, is capable of computing step-by-step guidance for a user while adapting to the user's skill level. *Robert*'s architecture has four components: *Planner, Ontology, Dialog, and User Interface*, and it uses hierarchical planning to adapt the instructions it provides to a user by establishing an ongoing dialog.

The general and specialized architectures proposed for Companion Systems have a strong focus on hierarchical planning to compute step-by-step guidance for a user. Although in later contributions of the project, sensing is positioned as a first-class component of the architecture, it is merely used to modify an already computed planning process. In these architectures, behaviors that are seemingly proactive merely occur as a side effect: For instance, *Robert* becoming active by itself is contingent on the presence of smart tools that inform the system about currently performed user activities. For DCs, we rather desire an architecture that emphasizes the *proactive* interpretation of scene information to enable autonomous user support.

2.2 Intelligent Assistants Architectures

In [34] a uniformed framework for building intelligent assistant applications is proposed. Guzzoni's objective is to create human-centered applications designed to assist users capable of *observing, understanding, learning, and acting*. To achieve this, an architecture with three components is proposed, namely *Observe, Understand, and Act*. Thus, an intelligent assistant uses multimodal sensing means to *Observe* a user and the environment. The *Understand* component interprets and learns from the observed data, and plans for the next action to take (either in reaction to a command, or in an anticipatory manner). Finally, the planned actions are executed by the *Act* component. In [55], Sarikaya analyses Personal Digital Assistant (PDAs) applications on smartphones that help users complete their tasks (e.g., creating reminders, setting alarms, providing recommendations for flights). This analysis resulted in generalized architectures for proactive and reactive assistance. Relevant to our work is the proactive assistance architecture, which consists of five main components: *Collect and Process, Aggregate, Inference and Learn, Rule Recipe Authoring, and Deploy and Publish*. The *Collect and Process* component parses, enriches and filters raw data. The *Aggregate* component contextualizes data on time and location. Such data is used by the *Inference and Learn* component to learn about users preferences and habits, and to add new rules through the *Rule Recipe Authoring* component. The two latest components are also used to compute personalized recommendations communicated through the *Deploy and Publish* component.

Similar to the classical agent's decision cycle [71], the abstracted view that Guzzoni's architecture proposes is relevant for our work. However, the *Understand* components is a very busy component in charge of several activities, such as contextualizing, disambiguating, validating data, and learning. Thus, providing fine grain configuration of a specialized component (such as a learning algorithm) might entail a large task that can disrupt other functions of the larger *Understand* component. On its side, Sarikaya's architecture for proactive assistant applications for smartphones is a pragmatic view that considers components for providing basic assistance such as setting reminders or making suggestion given the user's current location and past history. However, assistance that requires sensing the current state of the environment and the current actions of a user are not considered, since this would need additional contextualization of sensor data, rather than mere aggregation.

2.3 Companion Robots Architectures

Romeo2 [44] is a companion for everyday life. It relies on a *sensing-interaction-perception* architecture composed by five layers, namely *Sense, Cognize, Recognize, Track, and Interact*. Through these layers, Romeo2 achieves three levels of situation awareness: In Level 1, the state of the environment is perceived through the *Sense* and *Cognize* layers; in Level 2, a goal-oriented understanding of a situation is generated using the *Cognize* and *Recognize* layers; and in Level 3, the *Track* and *Interact* layers make predictions on the next action. Since Romeo2 puts an emphasis on situation awareness, it proposes a taxonomy of elements to perceive, which includes the environment as a geographical location, objects, people and the robot itself. In [36] a cognitive architecture for teaching assistant robots is proposed. This architecture consists of four units: *Logic, Memory, Perception, and Action*. Interactions with the user are handled by the *Perception* and *Action* units. The *Perception* unit processes the raw sensor data into meaningful information that is then stored in the (short term) *Memory* unit. The *Memory* unit stores data and acts as the bridge among all other units. The *Logic* unit makes decisions and creates plans executed by the *Action* unit. This architecture is validated through a teaching assistant robot for children with hearing impairments. SYMPARTNER [30] is a functional-emotional at-home assistant robot for elderly people that provides cognitive and motor stimulation to its users. The architecture of SYMPARTNER is divided in several layers that range from the hardware level to the arrangement of scenarios that the robot can support. Relevant to our work is the *Skills* layer, which proposes five main components, namely *Mapping* of the environment, *Perception* of obstacles, objects, people, and gestures, *Localization*, *Interaction* with the user, and robot *Navigation*.

Romeo2 puts a strong emphasis on the perception of the environment and the user: Four of its five components deal with sensing. We argue that proactive companions should have a stronger focus on the actual computation of assistance or protection for a user. The teaching robot proposes a more balanced architecture. However, the *Memory* unit is a single point of failure. Each unit could contextualize the data further to avoid overloading a single unit. The components proposed by the SYMPARTNER architecture shows a strong focus on perceiving the environment so the robot can navigate in it; and it is mostly focused on assistance that is initiated by a user.

The surveyed related work shows that all the architectures propose at least three components for: *sensing* the environment, the user or both, *computing* assistance for a user, and *delivering* it. Moreover, these architectures show a strong focus on sensing; which is in line with Dunne et al. analysis on the positive role that IoT has played in the development of AmI systems. However, given the current advancements in generative AI (e.g., Large Language Models—LLMs), and neuro-symbolic approaches (e.g., Knowledge Graphs—KG—and Convolutional Neural Networks) we argue that a modern architecture for AmI systems should pay special attention to contextualizing the data that IoT devices produce. This contextualization can enable a finer grain understanding of the current situation, which in turn can produce better computation of assistance and protection for users of AmI systems.

3 A MODERN ARCHITECTURE FOR AMBIENT INTELLIGENCE SYSTEMS

To achieve some of the most relevant AmI attributes, i.e., *sensitivity, responsiveness, adaptivity, transparency, and intelligence* [17] our architecture has two specific features: First, we propose distinguishing between four phases in the operation of an AmI system: *Perceive*, *Interpret*, *Decide*, and *Interact* (see Figure 1). These four phases are inspired by the *Perceive-Decide-Act* loop of intelligent agents that are situated in an environment [61, 71]. However, given that AmI environments are equipped with a large number of sensors, there is vast availability of data that can be obtained during the *Perceive* phase, such data needs to be contextualized to get a good understanding of the current state of an environment. Thus, we propose to add the *Interpret* phase to the traditional cycle. In this phase, shared meaning is given to the data that such sensors (and other connected devices) produce; this is of special importance in comparison with previous architectures because shared meaning produces a finer grain understanding of the current situation, since data is not only classified (e.g., as high and low for a temperature sensor), instead data is enriched with further knowledge (e.g., from a KG or an LLM). Moreover, since the AmI systems we strive to create aim at assisting and protecting human users, we highlight the importance of keeping the user at the center of the system by transforming the traditional *Act* phase into the *Interact* phase. Second, we propose a separation of concerns between types of AmI systems: *Expert DCs* manage a (physical or virtual) environment and *Personal DCs* are concerned with a specific user. In the following, we detail the four proposed phases (and matching components) of our architecture, and in Section 3.5 we explain the distinction between Expert and Personal DCs in greater detail.

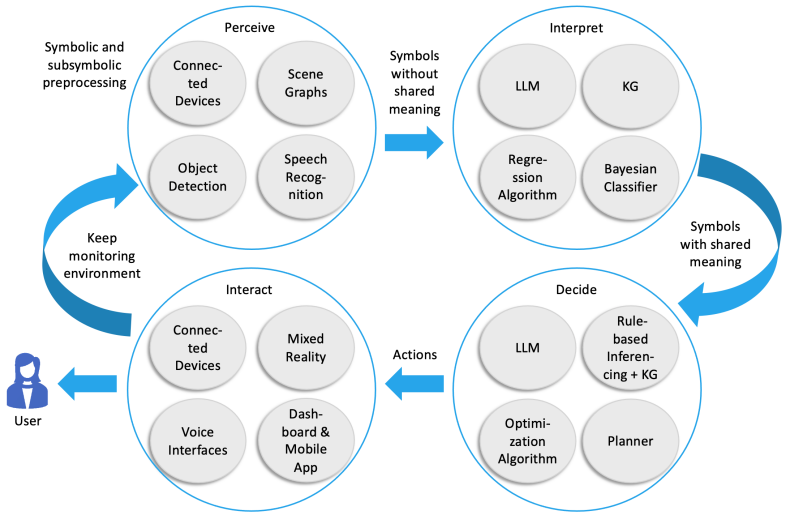


Fig. 1. Phases of our proposed Digital Companions architecture. The phases are populated with examples of what technologies and approaches could be used in each phase. However, the concrete technology to use depends on the implementation context of a Digital Companion.

3.1 Perceive

In this phase, a DC is in charge of observing the current state of a physical environment through *Connected Devices* (e.g., sensors) that AmI environments are traditionally equipped with (e.g., cameras, temperature, presence, and CO2). Information from connected devices can be obtained using IoT protocols (e.g., MQTT and the Constrained Application Protocol—CoAP), and higher-level technologies such as the Web of Things [33], which proposes the

creation of uniform descriptions of the programming interface of connected devices through the usage of W3C WoT Thing Descriptions (TD). The data these connected devices produce corresponds to symbols that have not yet been contextualized in the AmI environment they sense, neither have they been contextualized in terms of a specific user. Thus, a temperature sensor reading of 50, without further contextualization—*shared meaning*—does not convey that an action might be needed given that it is 50*C in a server room*. Once the connected device readings are obtained, the *Perceive* phase considers symbolic and sub-symbolic techniques to pre-process such readings (see Figure 1). Thus, in the case of a temperature sensor, the pre-processing step might correspond to classifying the value as `higher-than-average` or `lower-than-average`. In the case of a visual sensor, the *Perceive* phase might use a Scene Graph algorithm [16] that takes images as input and outputs triples that describe the detected objects in a scene and relationships among them (e.g., `man holding bottle`—*subject, predicate, object*); or a speech recognition algorithm might detect the sentence *"book an appointment"* from a microphone's recording. Regarding Cook et al. [17] most relevant AmI attributes, the *Perceive* phase produces useful data towards the creation of *sensitive* and *adaptive* (to the user) AmI systems.

3.2 Interpret

In the *Interpret* phase, a DC provides *shared meaning* to the symbols produced in the *Perceive* phase. *Shared meaning* refers to situating such symbols within a broader context: This contextualization happens with respect to the relevant AmI environment (e.g., that the system is sensing 50C *in a server room*) and takes into account further information, including common sense ("Is 50C a high or low temperature value?"), domain expert knowledge ("Is 50C a high or low temperature value for a server room?"), or knowledge in a broader perspective (e.g., cultural knowledge). In our architecture, we do not intend to constrain *how* this contextualization happens: Some of the algorithms capable of achieving this broader contextualization can be categorized as symbolic or sub-symbolic AI [40]. Regarding symbolic approaches, *KGs* are more relevant for specialized domains that rely on knowledge gained through the experience of human domain experts (e.g., a KG used to trace products in a production line [20]); and where efforts have already been made to represent such knowledge in a machine-readable and machine-understandable way. For applications in need of a broader context (e.g., a smart home for everyday AmI situations), a common-sense KG [39] might be more suitable. In terms of sub-symbolic AI, *LLM* [12] can also provide a wider useful context to the symbols computed in the *Perceive* phase, e.g., given that a temperature sensor is located in a server room, a proactive system could use an LLM to know the considered *normal temperature* for this type of spaces. A combination of sub-symbolic and symbolic AI algorithms could also be used in the *Interpret* phase. Such is the case of a regression algorithm that given the current and past observations of an AmI environment predicts that a room's temperature will increase by 10% in the next hour. Even if this is not a usual pattern, this prediction could still be made if the regression algorithm had access and means to understand (*shared understanding*) a symbolic representation (e.g., in a KG) of the latest workload schedule of the servers in the room. This *shared understanding* could be achieved among algorithms and systems by using standard ontologies[1] and well-known vocabularies[2] that provide *shared meaning* to the data. This *shared meaning* enables the creation of *transparent* AmI environments whose provenance can be explained.

3.3 Decide

In this phase, a DC computes timely and pertinent *proactive* and *reactive* actions to take in an AmI environment, for assisting or protecting a user. We define a *proactive* action as one that is taken *in anticipation of* a user need. Thus, a DC must be able to continuously perceive the environment and the user, and to interpret these data to understand a situation. Thus, if a user is in close proximity of a situation that might become dangerous for them,

[1] e.g., http://xmlns.com/foaf/0.1/ and https://www.w3.org/TR/vocab-ssn/
[2] e.g., https://schema.org/

a DC could take *proactive* actions to alert them, or could even actuate on the connected devices to contain such situation. An action taken by a DC is *reactive* if a user interacts with a DC through any kind of user interface (e.g., a dashboard, a mixed reality interface, or a mobile app) to request an *under-specified* action (e.g., *prepare the meeting room for my board meeting*). An action is considered *under-specified*, if it is too general, and parameters that are relevant for adapting an AmI environment have not been determined, e.g., desired ambient conditions such as light and temperature, audio, and visual content needed to be shared with other meeting attendees, and external hardware required (e.g., a loudspeaker). A well-specified request might not need a DC to go through the proposed cycle, since assisting a user to *increase the volume of a specific speaker by* 50% only requires a DC to actuate directly over a connected device. Even though the possibility of enabling such *proactive* actions might sound more relevant in the context of AmI environments, we consider that both *proactive* and *reactive* actions are very valuable. To ensure users are always in direct control of an action if desired. Additionally, we aim at respecting a user's autonomy by creating *proactive* actions as much as a user feels comfortable with them.

For a DC to decide on an action to take based on the output of the *Interpret* phase, it needs knowledge about the possible states in an AmI environment, and how to reach them. To reach a state, several variables might need to be changed (e.g., cooling a room might require starting the air conditioning unit, closing the windows, and constantly monitoring the temperature to keep it stable); this change might be achieved by actuating a connected device, or by using a virtual service (e.g, initiating an algorithm to forecast energy consumption of a building). To reach those states in an AmI environment, we consider symbolic and sub-symbolic AI approaches capable of understanding the *shared meaning* that the inputs from the *Interpret* phase have been enriched with. Some examples of these approaches are: a rule-based inference system operating on a KG, an LLM capable of providing concrete decisions (e.g., start fan, stop heating), and a planning algorithm that produces a set of steps to reach the desired environmental state. Thus, the *Decide* phase provides the means to create *adaptive* and *intelligent* AmI environments; *adaptive*, since it computes the actions to take on the physical environment, and *intelligent* because decisions on the actions to take are not made in an isolated way, they are based on the input from the previous phases that consider the AmI environment and the user.

3.4 Interact

In this phase, a DC establishes a dialogue with a user to deliver the computed assistance or protection. Such communication may be accomplished through a traditional user interface such as a dashboard or a mobile application, or through innovative user interfaces, e.g., mixed reality or voice interfaces. Moreover, in the case of *proactive* actions that can be directly actuated in an environment, a DC may make use of connected device (e.g., starting the air condition unit, or opening the blinds to adjust the brightness in a room). This type of *proactive* actions, where no active user feedback is required, are usually referred as *implicit interactions* [56]. This phase provides the *responsiveness* feature to the AmI environments we propose, since given the input from the previous phases, a DC is capable of delivering the computed assistance or protection to a user.

3.5 Expert and Personal Digital Companions

In our architecture, all DCs follow the proposed phases to *Perceive* the current state of their environment and *Interpret* the collected data to contextualize what happens in a space and compute possible intentions of users; they then *Decide* on proactive and reactive actions to assist or protect users, and *Interact* with them through appropriate human interfaces to communicate their intent or to obtain further input before performing an action.

We consciously distinguish between two types of DCs to achieve a separation of concerns that we propose is highly relevant in human-centered AmI environments: On the one hand, *Expert* DCs are put in charge of a specific virtual or physical environment. An Expert DC is aware of artifacts that are situated in this environment and is in charge of setting and possibly enforcing deontic requirements in the environment. Specifically, in the

proposed industrial scenario, an Expert DC is in charge of monitoring the robot operation and the environment conditions to determine if a situation is safe or if risk might be present. On the other hand, we propose *Personal* DCs, which are specialists with respect to a specific user and have access to their (personal) data. They have possibly gathered information about a user through observation, learning habits, preferences, and a profile; or the user might have shared personal data with them, e.g., through a decentralized personal data store—recently proposed for sharing gaze data [5]. Hence, such DCs are capable of supporting users in a personalized way.

Our proposed distinction between Expert and Personal DCs achieves a *favorable separation of concerns* that is relevant for responsible AmI in human-centered settings. Additionally, it allows for increasing the coverage of AmI systems towards *ubiquity*. According to our architecture, Personal DCs take the role of gatekeepers to personal information: While these DCs are trusted enough by users to be permitted to observe them (e.g., through direct observation or through access to user data pods), they are not specialists with respect to the environments a user might roam. This role is rather assigned to Expert DCs—an environment-oriented mirror image of Personal DCs (see Figure 2). Expert DCs do not concern themselves with personal information about the users, but they are experts in the environment they are situated in. For instance, they might have learned, or might have been programmed with, information about interface descriptions or artifact manuals [66] of virtual services and physical devices in the environment. This enables them to support users in that environment—either through direct interaction with these users or while using users' Personal DCs as proxies. This separation of concerns furthermore is central to enable *evolvable* AmI systems: Expert DCs and Personal DCs are not tightly coupled in our architecture, which allows them to evolve independently. This argument for the reduction of architectural coupling stems from software architecture research and is also central to the evolution of long-lived, open, and highly scalable systems such as the World Wide Web [23].

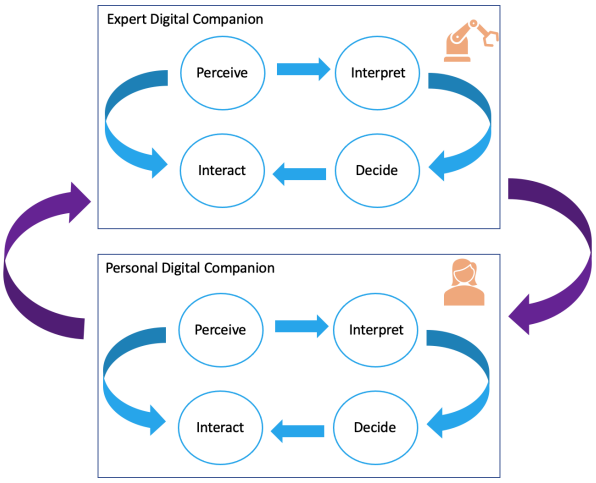


Fig. 2. *Expert* and *Personal* DCs collaborate to create AmI systems

4 ROBOTS THAT UNDERSTAND THEIR ENVIRONMENT

To validate our proposed architecture, a system has been implemented to operate in an industrial scenario in which human workers collaborate with robots to accomplish their tasks. Our scenario involves a (UFactory xArm) robotic arm transporting workpieces from a shelf into a box. This task is performed next to a grinding station that might produce sparks, creating a safety hazard when flammable substances are present: If such

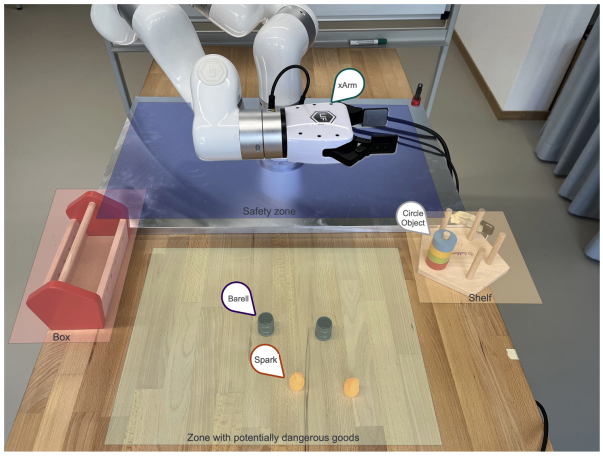


Fig. 3. Laboratory setup to simulate industrial scenario.

materials caught fire, the workpieces and the human workers would be in immediate danger. The environmental conditions—predominantly the ambient temperature—also influence the current (fire) risk level and the system's response to dangerous situations: In a low-temperature scenario, it might be possible for the worker to extinguish a localized fire, but at higher temperatures they should leave the premises and a fire alarm should be raised.

To simulate this scenario, we created a laboratory setup utilizing plastic pieces to represent sparks and barrels containing flammable materials, colorful circular objects to represent the circular workpieces, a wooden shelf, and a wooden box for the transportation task (see Figure 3). Under normal conditions, a robotic arm extends from the safety zone (the silver plate the robot is mounted on), a wooden box is on the table, circular object are on the shelf or in the box, sparks and barrels containing flammable goods are in the working zone (located between the shelf and the box) considered as *potentially dangerous*. In normal operation, the robot performs three steps:

(1) Initialization. The wooden box is either in the safety zone or in the working space, the circular workpieces are on the shelf, and the flammable materials are not in direct contact with the sparks.
(2) Working. The box is on the working zone, and the transporting task has started. In this mode, the scene is inspected visually to determine if there are workpieces on the shelf that need to be transported.
(3) Done. The robot has finished transporting all the workpieces to the box and has put the box in a zone for collection (i.e., in the back of the robot in Figure 3).

Considering this scenario and our proposed architecture (Section 3), we created an AmI system that manages dynamic risks in the workspace. The system should react appropriately to dangerous situations, by continuously monitoring the environment and by taking appropriate *proactive* action if the risk level rises. To this end, we created an Expert and a Personal DC. The Expert DC *perceives* the environment, *interprets* the current state, *decides* on the next action to take and *interacts* with connected devices. During routine operation, the Expert DC transitions between the *Initialization*, *Working*, and *Done* states, as presented above. In case of a detected anomaly, the expert DC *decides* to safeguard the workpieces (e.g., move them to the safety zone), and *interacts* with connected devices if needed (e.g., start the fire sprinklers in the ceiling). On its side, a personal DC communicates with the Expert DC to obtain the decision that has been made regarding the environment, so it can *interact* with the workers and keep them in the loop. Under potentially dangerous conditions, the Personal DC *perceives* and *interprets* the current situation from the worker's perspective, so it can *decide* on the next action considering the worker's individual characteristics (e.g., a trained worker may try to extinguish the fire in case of moderate

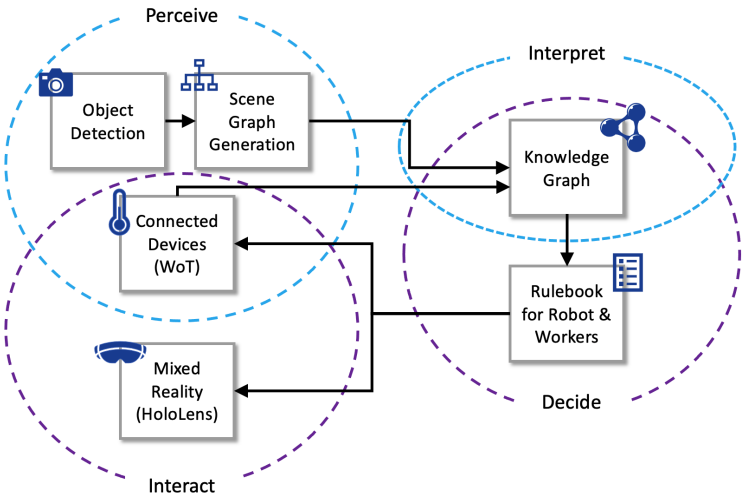


Fig. 4. Software Components of the Proposed System

risk). Finally, the Personal DC *interacts* with its user utilizing an appropriate interface (e.g., visual instead of voice in noisy spaces as it is our case). Together, the Personal and the Expert DC hence are able to integrate their knowledge—about the worker and the environment, respectively—where each of them *interprets* the given situation from their own (worker-centric or environment-centric) viewpoint; they both then *decide* (by entering in a negotiation if necessary) how to *interact* with other entities—the worker and entities in the environment, respectively—according to the separation of concerns that our architecture induces (see Section 3.5).

For the DCs to operate, we implemented the software components shown in Figure 4: For the *Perceive* phase, a visual object detection system, a scene graph algorithm, and connected devices are available. A KG is used in the *Interpret* phase. For the *Decide* phase, a rulebook was implemented to evaluate, detect, and follow procedures for normal or at-risk operation. For the *Interaction* phase, an MR application and connected devices are used. In the following, we describe each implemented software component.

4.1 Connected Devices

Our implementation makes use of Web of Things-enabled [33, 62] devices in the *Perceive* and *Interact* phases. The Web of Thing's (WoT) core tenet is to create a common application layer for IoT devices, which has recently been standardized by the World Wide Web Consortium (W3C).The data and services that WoT-enabled devices provide is made available for consumption through W3C WoT Thing Descriptions[3] (TD), which are standardized semantic representations of the programming interface of the device, i.e., a type of interface definition language. Hence, a DC can read from sensors described through a TD, or actuate a robot through (e.g., HTTP or CoAP[4]) interfaces that are also described using the TD standard.

During the *Perceive* phase, we use a temperature sensor and two cameras that our AmI environment (i.e., our laboratory) is equipped with. The TDs of these connected devices lead us to current and historical readings of the temperature sensor, and to the live-stream of two cameras observing the AmI environment. The live-stream of these cameras is used by the visual object detection and scene graph algorithms to detect visual relationships

[3] https://www.w3.org/TR/wot-thing-description11/
[4] https://www.rfc-editor.org/rfc/rfc7252.html

among objects in a scene. The readings of the temperature sensor are later used in the *Interpret* and *Decide* phases to determine the risk level (*moderate or high*) and decide on the actions to take. In the *Interact* phase a DC uses connected devices to actuate on the environment, e.g., controlling the robot to safe guard the workpieces and actuating on a simulated fire sprinkler to contain a fire.

4.2 A Scene Graph for the Robot's Transportation Task

Convolutional Neural Networks (CNNs) [47] have brought robustness and speed to computer vision tasks such as object detection. YOLO (You Only Look Once) [10, 50, 51, 67] is a one-stage object detection algorithm capable of computing bounding boxes and class probabilities at the same time, which makes this algorithm extremely fast, and has enabled the creation of applications that require near real-time detection, which was not possible with two stage algorithms (e.g., R-CNN, Fast R-CNN [29], and R-FCN [19]). Even though these robust object detection algorithms exists, they provide limited visual contextual information for a smart agent to understand the current state of an environment. Thus, we are interested in Scene Graph Generation (SGG) algorithms that produce a symbolic representation of objects and their relationships in a scene (a Scene Graph—SG). These representations typically take the form of triples: *subject, predicate, object* [41]. Similar to object detection algorithms, there have been proposed one [45, 46] and two stage SGG approaches [32]. In two stage approaches, objects are detected first and relationships are computed later [18, 49]. Among other applications SGG algorithms have been used for improving image captioning [2, 48], visual question answering [37] and image retrieval [41, 68].

To achieve a well-informed visual perception of the environment for our DCs' *Perceive* phase, we created a custom SGG algorithm that uses the YOLOv4 [10] algorithm to recognize objects in our industrial scenario (e.g., barrels that contain flammable materials, wooden boxes, circular workpieces, and sparks), then through Intersection over Union (IoU) and a decision tree, our algorithm predicts spatial relationships among detected objects in an image. A richer visual understanding of the industrial environment is of special interest in our implementation, since for example, barrels that are located in the same zone as sparks should not be considered as risks; hence, the robot should continue performing its normal transporting task. However, when barrels start catching fire, a risk should be detected and safe-guarding actions should be taken. We created our own implementation of a SGG algorithm, since the current implementations are still in an early state, showing long processing times and requiring large data sets to train models with custom objects (e.g., a barrel, and a robot).

To train YOLOv4, we created a custom dataset of our laboratory setup in different lighting conditions (i.e., natural and artificial lighting), at different distances from the objects, and with different levels of clutter in the scene. Given that some of the considered objects (e.g., the circular workpieces inside the wooden box) are not always visible in a front camera perspective, we added a *bird's-eye view* with a camera mounted on the ceiling of our laboratory (see Figure 5). A total of 1,186 pictures were annotated using the Labelbox tool[5]. The resulting dataset is composed of 23'139 annotations, including objects and the relationships among them. The considered objects are: `Gripper`, `Table`, `Barrel`, `Safety Zone`, `Shelf`, `Box`, `Spark`, *and* `Circular Object`; while the relationships between objects include `on`, `on fire`, `grasps`, *and* `in` (see Table 1).

From YOLOv4, we obtain labels of the detected objects, the coordinates of the bounding boxes where the objects were detected, and a confidence score. If more than one object is detected in an image, we analyze the image to find relationship among all objects. To this end, the IoU between every two bounding boxes is computed; an $IoU = 1$ indicates a complete overlap and an $IoU = 0$ indicates no overlap at all. In our custom SG algorithm, if the IoU between two boxes is greater than 0 but less than 1, such objects are considered to be related. Then, to determine which relationship exists among two objects, a decision tree was trained with the relationships in Table 1. Since our setup consists of two visual perspectives (from the available cameras in the AmI environment), the captured images are analyzed almost simultaneously and the generated SGs are merged. When merging both

[5] https://labelbox.com/ with an education license

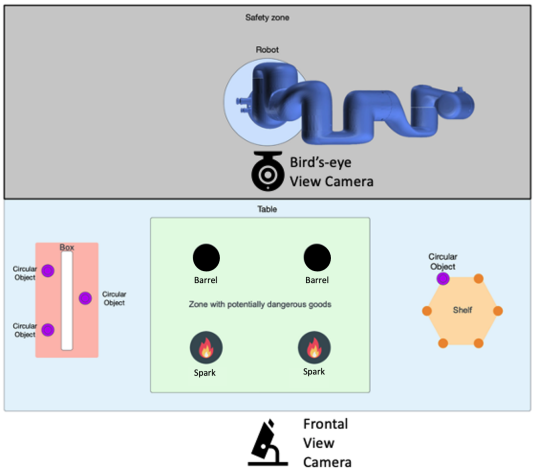


Fig. 5. In our setup, two cameras (bird's eye view and frontal) are used to capture the current scene.

Object 1	Relationship	Object 2
Gripper	grasps	Box
Gripper	grasps	CircularObject
CircularObject	in	Box
Barrel	on	Table
CircularObject	on	Shelf
CircularObject	on	Table
Box	on	Table
Box	on	SafetyZone
Barrel	on fire	Spark
Spark	on	Table
Spark	on	SafetyZone
Shelf	on	Table
Shelf	on	SafetyZone

Table 1. Considered relationships among objects detected in a scene.

SGs, those triples that include relationships among objects that are better detected from one perspective—such as the bird's eye camera—are deleted from the SG generated from the other camera (the front perspective)—e.g., the `CircularObject in Box` triple is clearer from the bird's eye perspective. A limitations of using the overlapping of bounding boxes to determine relationships is that in the cause of objects located close to each other, *false-positive* relationships may occur with a slight overlap of bounding boxes. To minimize this *false-positive*, we implemented a post-processing step that verifies that to be considered a spatial relationship, the bounding boxes (*IoU*) of objects should overlap at least in 5% . Figure 6 shows our laboratory setup; on the left side there is an overlaid of the generated SG and the right side shows only the bounding boxes of the detected objects.

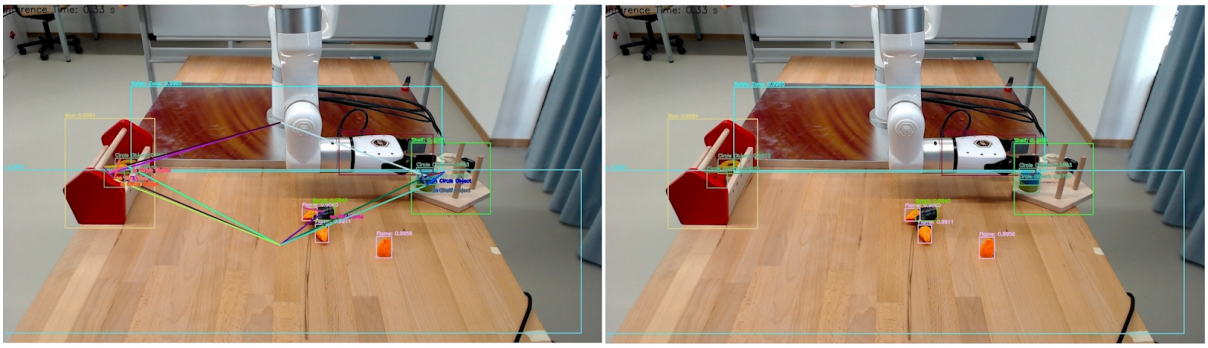


Fig. 6. Our setup showing detected objects and their relationships on the left, and only detected objects on the right.

4.3 Knowledge Graph

Knowledge Graphs (KGs) have been associated to a broad variety of knowledge-based approaches, implementations, and technologies; all with the common objective of *contextualizing data with richer descriptions*. However, semantic approaches appeared since the first wave of AI [1], in which Knowledge Bases (KBs) were created to act as the oracle of a system. In 1991, Gruber [31] proposed the creation of reusable *ontologies*: formal machine-readable and machine-understandable representations of a domain, capable of providing *shared understanding* across systems. Ontologies define concepts, taxonomies and other relationships among such concepts [59]; as well as logic for reasoning and inferring [3]. These then became known as Semantic Technologies. Perhaps, the most well-known application of Semantic Technologies is Tim Berners-Lee vision of the Semantic Web [9, 35]; in which autonomous agents discover and consume information and services available on the Web. In recent works, KGs have been combined with sub-symbolic approaches [26] to for example contextualize further a label that has been produced by a convolutional neural network for object detection.

In our implementation, we created a KG to describe in a machine-readable and machine-understandable way our AmI environment. Such KG incoporates well-known and standardized ontologies, and it is used in the *Interpret* and *Decide* phases of the proposed architecture. In the *Interpret* phase, the KG semantically contextualizes the symbols (objects, relationships, and sensor data) from the *Perceive* phase. In the *Decide* phase, the KG acts as a KB to chose an appropriate action. This approach renders our system semantically interoperable with other AmI systems (specifically, with other DCs). The KG uses the following ontologies:

- DUL: The DOLCE ontology [11, 25] is an upper ontology for the Semantic Web [9] that describes general concepts such as `PhysicalObject` and `Non-PhysicalObject`. We are interested in the `PhysicalArtifact` and `PhysicalAgent` concepts. We extended the `PhysicalArtifact` class with subclasses that describe the components involved in the Robot transportation task (e.g., `Shelf`, `Box`, `Spark`, `Barrel`, and `EndEffector`). We also subclassed DOLCE's `PhysicalAgent` class with the `Robot` and `RoboticArm` concepts. We then specialized the DOLCE class `Situation` by creating a `FireCondition` class denoting *normal* or *abnormal* conditions as well as risk levels such as `highRisk`, `moderateRisk`, and `lowRisk`. To provide steps to follow in specific situations, we specified the class `Task` to define steps to `EvacuatePersonnel`, `MitigateRisk`, `OperateNormally`, and `SafeguardWorkPieces`.
- SSN: We take advantage of the alignment of DUL with the Semantic Sensor Network Ontology[6](SSN) [15] to describe `Sensors` in our AmI environment (i.e., `Camera` and `Temperature` sensors).

[6] https://www.w3.org/TR/vocab-ssn

- FOAF: We use the Friend of a Friend ontology[7] FOAF to describe users of our system. A Personal DC uses this information to compute customized assistance or protection, and for personalized interactions.
- QUDT: We use the Quantities, Units, Dimensions and Types ontology[8] to define a `Quantity` subclass that describes a material's `Flashpoint`—the lowest temperature at which a substance is ignited by an external source (e.g., a spark). We also use the unit `Cel` (i.e., Celsius) to characterize temperature ranges that are considered as *normal* or as *risky* when there are combustible or flammable materials in the environment.
- BOT: We use the Building Topology Ontology to describe the zones in our AmI environment; thus we created the specialized `Zone` with the subclasses `SafetyZone` and `WorkingSpace`.
- TD: The Thing Description ontology[9] is used to describe the programming interface of the WoT-enabled devices that our AmI environment is equipped with, such as temperature sensor, cameras, and the robot.
- SBEO: We use the Smart Building Evacuation Ontology[10] [42] to describe objects that can be used in emergency situations, such as `FireExtinguisher` and `Mobilephone`.
- DCAmI: Is our own ontology that imports the previous ontologies. Here, we define our scenario-specific concepts, including `FireCondition`, `EvacuatePersonnel`, and `MititageRisk`, which are subclasses of concepts defined in the DOLCE ontology. Moreover, we define the `Substance` concept and create subclasses for `Combustible` and `Flammable` substances.

4.4 Rulebook for Robot and Workers

The best action to perform at a specific moment is computed in the *Decide* phase by evaluating the current state of the environment. Hence, it is necessary to know the set of possible (*proactive or reactive*) actions that a Personal or an Expert DC can perform. In our prototype, this set of actions is defined in a rulebook that makes use of the information that the system has obtained in the *Perceive* phase and which has been put in context using the introduced ontologies in the *Interpret* phase. Thus, when the `onFire` relationship between a barrel and a spark is detected, and the temperature sensor reading is above 25C, the risk is computed as *high* and actions to *safeguard workpieces* and *evacuate workers* are selected. In case the risk is *moderate* (given a lower environmental temperature) the actions to *safeguard workpieces* is also selected, but *extinguish fire* is selected for the workers. If there is no detected relationship between a barrel and a spark, the robot transportation task continues normally.

In the *Interact* phase, the components of our AmI system—driven by Expert and Personal DCs—execute the actions that have been determined in the *Decide* phase based on the contextualized scene information. For instance, an Expert DC that decided to perform the *safeguard workpieces* action instructs the robot (using its programming interface described in a TD) to stop the transportation task, and to place the box with the workpieces in the safety zone. On its side, a Personal DC utilizes an MR application to navigate workers to the emergency exit in case of *high risk* (following the *evacuate workers* actions); or to navigate workers to the fire extinguisher in case of *moderate risk* (following the *extinguish fire* set of actions).

4.5 Mixed Reality Application

From the *seven grand challenges* that Stephanidis et al., define [60] to address in the era of intelligent and interactive systems; we are specifically interested in tackling the *human-technology symbiosis* and *human-environment interactions* challenges. *Human-technology symbiosis* refers to how humans can live in harmony with computer systems that exhibit human-like characteristics (e.g., learning, reasoning, and language understanding). In this regard, Gaggioli et al., [24] describe *human-computer confluence*, referring to systems that approach

[7] http://xmlns.com/foaf/0.1/
[8] https://qudt.org/
[9] https://www.w3.org/TR/wot-thing-description11/
[10] https://qasimkhalid.github.io/SBEO/



Fig. 7. Functionalities of the implemented MR application for HoloLens2.

humans, instead of humans approaching systems—very much in line with the ambient intelligence ideals. Thus, Gaggioli et al. propose creating technology that considers the user perspective from the very beginning of its conception. *Human-environment interactions* concern smart ecosystems that are equipped with a plethora of devices and resources (as AmI environments are), which in turn increases interactivity. Some of the highlighted innovative human interfaces to facilitate such interactions and get closer to human-computer confluence are virtual, augmented and Mixed Reality (MR). MR is of particular interest for the AmI systems we propose, since it ensures that the user is never detached from the physical world while being engaged in the virtual world [57].

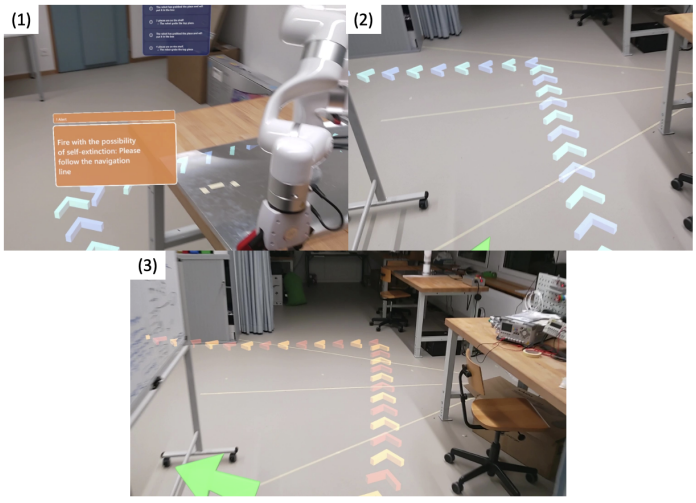


Fig. 8. Alert window and navigation paths (screenshot from MR app)

.

Thus, we created an MR application on a Microsoft HoloLens 2 that enables Personal DCs to communicate with human users. This application has the following functionalities:

(1) A *Scene Graph Panel* that shows to the user triples of the continuously generated) SG (see Figure 7(1)) and thereby provides a real-time glimpse into the *Perceive* phase, keeping the user in the loop.
(2) A *Tooltip Panel* that provides insights about the autonomous actions of the AmI system by using the SG and enriching it with information from the KG to provide better explanations of the robot actions, and about the decisions that a DC makes (see Figure 7(2)).
(3) A *Semantic Risk Indicator* that displays the current risk level of the AmI environment (e.g., low, moderate and high) computed in the *Decide* phase (see Figure 7(3)).
(4) *Navigation Paths* are displayed when the risk computed is moderate or high. If moderate, the user is guided to a fire extinguisher (see Figure 8(1) and (2)) or to the emergency exit if it is high (see Figure 8(3)).

4.6 Testing our Prototype with Potential Users

To test our prototype, we conducted a user study with 13 participants: 9 participants identified themselves as male and 4 as female. The average age is 28.15 years old. 10 participants have a bachelor's degree, and 3 participants have also a Masters degree. Each participant took around 45 minutes to do the study, including familiarization, actual usage of the MR application, and 20 minutes to respond to a questionnaire about their experience. Previous to the experiment, participants were briefed about the task that the robot performs, i.e., the robot transportation task consist in moving the circular workpieces from the shelf to the box. Participants were also informed about the sparks and barrels containing flammable materials on the work surface. During the experiment, two dangerous situations were simulated by placing the sparks on top of a barrel.

After the experiments, participants were asked to answer a questionnaire (see Appendix A) designed considering Tintarev and Masthoff research on effective explanations [63, 64]; their work focuses on the creation of explanations that help users make qualified decisions; specifically for recommender systems. However, we argue that effective explanations are important for the AmI systems we propose, since a Personal DC should effectively interact with users to deliver the computed assistance or protection. Tintarev and Masthoff identified seven possible aims for creating explanations [64]; a system might be designed to optimize for one or several of these aims. Our questionnaire focuses on the following aims:

(1) Transparency: to describe how the system works.
(2) Effectiveness: to help users make good decisions.
(3) Trust: to increase the confidence of users in the system.
(4) Efficiency: to speed up the decision making process of users.
(5) Satisfaction: to increase the usability or enjoyment of the system.

Participants answered the questionnaire in Appendix A on a Likert scale from 1 (*strongly disagree*) to 5 *strongly agree*. Following we report on a the responses to a few questions in each category.

Transparency and Effectiveness

- *The system was successful in conveying the robot actions and its status:* 69% of participants strongly agreed, while 23% moderately agreed, and 8% strongly disagreed.
- *The scene graph the system shows was easy to understand:* 54% of participants strongly agreed while 38% moderately agreed, and 8% neither agreed nor disagreed.
- *The instructions the system provided me with were easy to understand:* 54% of participants strongly agreed, 38% moderately agreed, and 8% took neutral position.

- *I was able to forecast the behaviour of the robot given the information the system provided me with:* 46% of participants strongly agreed, 38% moderately agreed, 8% neither agreed nor disagreed, and 8% moderately disagreed.
- *I understood the decision making process of the robot:* 62% of participants strongly agreed, 31% moderately agreed, and 8% neither agreed nor disagreed.

Trust

- *I was confident in the system's ability to navigate me towards a safety place:* 77% of participants strongly agreed, while 23% moderately agreed.
- *I trust the system to provide me with accurate information:* 69% of participants strongly agreed, while 23% moderately agreed, and 8% neither agreed nor disagreed.

Efficiency

- *The system was fast enough to direct me to a safe place:* 54% of participants strongly agreed, while 46% moderately agreed.
- *The system was effective at guiding me in the situations that required me to take action:* 69% of participants strongly agreed, while 31% moderately agreed.

Satisfaction

- *It was not hard to understand the current state of the robot:* 62% of participants strongly agreed, while 38% moderately agreed.
- *The system provided me with all the required information to handle the situations I faced:* 23% of participants strongly agreed with this statement, 62% moderately agreed, and 15% were neutral about it.
- *I am satisfied with the overall experience using the system:* 46% of participants strongly agreed, 46% moderately agreed, and 8% neither agreed nor disagreed.

The results of the user study indicate that the system was well received by the majority of participants, with high scores in overall user experience and its ability to convey the robot's actions and status. However, there were some areas of confusion and difficulty to be addressed, participants reported to be confused when clicking some buttons and to find points that were only visible when the door was open (in the evacuation scenario).

5 ARCHITECTURE VALIDATION

We argue that using our proposed architecture is applicable for the creation of AmI systems across application domains, and that it is beneficial for the creation of human-centered AmI systems. To validate this, we present in Sections 5.1 and 5.2 two different scenarios; in Section 5.3 we discuss the advantages of using our architecture, and in Section 5.4 we point out some aspects to consider when implementing the AmI systems we propose.

5.1 Proactive Robots for Smart Homes

Christine is a technology enthusiast living in a smart home equipped with several connected devices (e.g., smart speakers, smart home appliances, and a modern entertainment system) and an AmI system consisting of an Expert DC in charge of managing the environment to assist and protect its inhabitants. Christine enjoys home improvement projects, currently she is working on a bookshelf to add to her office space. The AmI system perceives that she is about to operate the drill to fix one of the shelves to the wall. Thus, it actuates a robot vacuum cleaner and directs it to the office to clean as Christine's drilling produces debris. Christine can utilize her smart watch or a smart speaker nearby to (if desired) actively control the vacuum cleaner with voice commands. The proactive assistance of the robot vacuum cleaner is possible given that the Expert DC was able to *understand the current context* of the environment from visual information. Specifically, in the *Perceive* phase, the Expert DC constantly analyzes the camera stream of Christine's office and creates SGs. One of these SGs produces the

following triples: `person holding drill`, `person next to shelf`, `person wearing goggles`, and `person wearing protective globes`. This information is contextualized in the *Interpret* phase by an LLM that has been trained with everyday data. The output of this phase is the understanding that those triples correspond to a *drilling action* and that the possibility of *producing dust and debris is high*. In the *Decide* phase, the Expert DC computes the most suitable action to take, in this case *initiate cleaning*. In the *Interact* phase, the Expert DC actuates on the physical environment through the robot's TD, it directs it to the office and start vacuuming. Given the actions computed in the *Decide* phase, Christine's Personal DC sends a notification to her smartwatch reminding her about the possibility to control the robot vacuum cleaner through voice commands.

5.2 Taking your Personal DC to Work

Liam is a sound engineer working on setting up a studio for a video podcast recording in a smart building equipped with a HVAC system, smart blinds, temperature sensors, augmented reality projectors, cameras, and an AmI system in the form of an Expert DC that manages the physical environment. The working space is crowded, given that lighting technicians and producers are also on the move preparing the studio. Since Liam is a full-time employee, his Personal DC communicates with the Expert DC to get access to the sensing data produced by the connected devices in the environment. Thus, Liam's Personal DC is capable of gathering data about the current situation from a personal and an environmental perspective. In the *Perceive* phase, the Personal DC is reading and processing the environmental sensor data, and the data produced by Liam's smart watch. From the camera feed, a SGG algorithm's outputs triples such as `man carrying camera`, `tripod next to man`, `screwdriver in toolbox`; and a classifier evaluating the wearable data determines Liam's current physical effort as *Moderate to High*. These symbols are then contextualized on the *Interpret* phase using a domain specific KG for sound engineers, which describes professional equipment and links to additional content, such as videos and text-based manuals for installing, giving maintenance, and repairing the equipment. Moreover, the Personal DC has access to Liam's KG in which his level of expertise on different equipment is described. The output of the *Interpret* phase is then the understanding that *Liam is currently setting up a camera that he has only limited experience with and that his physical effort is rising*. Thus, in the *Decide* phase, a set of inference rules that have been learned a priori (by observing Liam at work) are used as well as both KGs to determine to assist Liam with step-by-step instructions to setup the camera. In the *Interact* phase, the Personal DC communicates with the Expert DC to negotiate the usage of an augmented reality projector that is in the space Liam is working. Finally, the Personal DC projects on the wall the instructions for Liam to follow. On its side, the Expert DC is constantly monitoring the environment to for example track mobile tools and equipment that tends to get misplaced, and watch out for the physical integrity of the people in the environment given its dynamicity.

5.3 Discussion

Our demonstrated prototype as well as the introduced scenarios emphasize the *versatility* of our architecture, the *interoperability* that it enables among AmI systems, and the importance of integrating a dedicated the *Interpret* phase, which is dedicated to contextualize the perceive data and can take advantages of currently strong technologies such as KG and LLMs. We argue that our architecture is highly *versatile* because it can be used to create AmI systems capable of assisting and protecting users in different physical spaces and in heterogeneous situations: Each component of the architecture permits the inclusion of a large variety of technologies to accomplish the component's purpose, which we have specifically emphasized for the *Interpret* components. And the separation between Expert and Personal DCs induces that individuals are accompanied by trusted personal AmI systems (that possibly have access to sensitive personal data) that support them while roaming heterogeneous environments by linking with expert AmI systems (that, likewise, have access to possibly sensitive data about the environment). This separation of concerns also enables *interoperability* among AmI systems that

follow our architecture: A first-time visitor in an office building could be given the same personalized assistance and protection that a well-known employee receives—this is enabled since the visitor's Personal DC knows its user in detail and can mediate (given a common understanding of the situation) the interaction with an Expert DC in the new environment to maximize its user's comfort.

Moreover, in comparison with other architectures described in Section 2, our proposed architecture abstracts from specific technologies to phases that could better define the role of a specific technology. Hence, a model for learning preferences of a user would continue learning on the *Interpret* phase, but it would be used for inferring on the *Decide* phase. Thus, if such a machine learning model becomes unusable (e.g., due to corrupted data), it could be replaced by a pre-defined user profile, allowing the AmI system to continue operating. Finally, as mentioned in Section 3, the proposed phases of our architecture are able to cover the features that Cook et al. [17] have found to be the most desired ones when creating AmI systems, namely *sensitive, responsive, adaptive, transparent, ubiquitous, and intelligent.*

5.4 Implementation Considerations

One of the main considerations when implementing the proposed AmI architecture is semantic interoperability across the four proposed phases. Consider the demonstrator in Section 4: The SGG algorithm produces triples, e.g., `barrel on table`, and `spark on table`; in the *Perceive* phase, the subjects, predicates, and objects in these triples are merely uncontextualized strings, and it is the purpose of the *Interpret* phase to contextualize them through a KG. However, for this contextualization to succeed, the KG needs to be synchronized with the SGG algorithm. Utilizing large KGs such as Wikidata or DBpedia might minimize this problem. However, specialized terms and relationships still need to be specified manually. In the *Decide* phase, the algorithm in charge of computing an action to take (e.g., a rule-based algorithm) should again be aware of the concepts that the KG uses to make a decision on the action to take in the *Interaction* phase. This semantic interoperability across phases can be achieved through integration already from the *Perceive* phase, for example by replacing the labels that the SGG algorithm considers to refer to the objects in a scene and their relationships (e.g., *barrel* and *spark*) with the IRIs that correspond to the machine-understandable description expressed in the considered KG. Even though utilizing other technologies such as LLMs in the *Interpret* phase (as proposed in Section 5.1) might sound like a remedy to the requirement for semantic interoperability, using an LLM out of the box is possible only for scenarios in which large quantities of data are available (such as the cleaning robot scenario); for highly specialized scenarios (e.g., an industrial process), a KG might be one of the most viable options.

Given the amount of personal data that can be captured in the AmI systems we propose, data privacy is a highly relevant topic that should be considered and acted upon. Specifically, for Personal DCs in charge of computing personalized assistance and protection for a user, transient data might not be sufficient. Thus, we propose incorporating technologies such as Solid Pods [54], whose aim is to give users the control of their data. Solid proposes decoupling systems from the data they use and allowing users to give fine grain access and processing control to their data. Thus, personal data captured in the *Perceive* phase of a DC could be stored in a user's Pod. The user then can decide to grant access to these data to its Personal DC, and not to the Expert DCs operating in the different environments that a user regularly spends time in (e.g., at work, or at a friend's house). The usage of Solid Pods with highly sensitive data, such as gaze-data has been demonstrated in [6].

6 CONCLUSIONS AND FUTURE WORK

In this paper, we present an architecture for modern AmI systems. This architecture aligns research on DCs and the objectives of the AmI research field. The objective of DCs is to assist and protect users in a *proactive* manner. To achieve this, we propose the creation of systems in charge of managing a physical environment called Expert DCs, and systems capable of learning about a specific user called Personal DCs. In our architecture, we

emphasize the separation of concerns between these types of DCs. We further propose that the operation of a DC, expert or personal, should be structured into four phases to compute and deliver assistance or protection, namely *Perceive, Interpret, Decide, and Interact*. For each phase, we propose a set of technologies that can be used. We validated our architecture in a simulated industrial environment in which a worker is in charge of an operation that has the potential to become dangerous. We tested the MR application by conducting a small user study with 13 participants and received positive feedback. Several streams for continuing this work will be followed. On the one hand, we are looking into utilizing end-to-end solutions for SGG that might be capable of recognizing other relationships outside spatial ones. As mentioned before, current SGG algorithms require vast amounts of data for training with custom objects (e.g., robots, sparks, and barrels) and they are much slower than well-established object detection algorithms. Moreover, we are interested in implementing other sub-symbolic approaches such as LLMs to contextualize data in the *Interpret* phase. We will investigate proactive interactions in AmI environments and will implement them in more complex scenarios.

REFERENCES

[1] Rajendra Akerkar and Priti Sajja. 2009. *Knowledge-based systems*. Jones & Bartlett Publishers.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference*. Springer, 382–398.

[3] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. 2018. A survey of ontology learning techniques and applications. *Database* 2018 (Oct. 2018), bay101. https://doi.org/10.1093/database/bay101

[4] Juan Carlos Augusto. 2007. Ambient Intelligence: The Confluence of Ubiquitous/Pervasive Computing and Artificial Intelligence. In *Intelligent Computing Everywhere*, Alfons J. Schuster (Ed.). Springer London, London, 213–234. https://doi.org/10.1007/978-1-84628-943-9_11

[5] Kenan Bektaş, Jannis Strecker, Simon Mayer, Kimberly Garcia, Jonas Hermann, Kay Erik Jenß, Yasmine Sheila Antille, and Marc Solèr. 2023. GEAR: Gaze-enabled augmented reality for human activity recognition. In *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications* (Tubingen, Germany) *(ETRA '23)*. Association for Computing Machinery, New York, NY, USA, Article 9, 9 pages. https://doi.org/10.1145/3588015.3588402

[6] Kenan Bektaş, Jannis Strecker, Simon Mayer, and Kimberly Garcia. 2024. Gaze-enabled activity recognition for augmented reality feedback. *Computers Graphics* 119 (2024), 103909. https://doi.org/10.1016/j.cag.2024.103909

[7] Pascal Bercher, Gregor Behnke, Matthias Kraus, Marvin Schiller, Dietrich Manstetten, Michael Dambier, Michael Dorna, Wolfgang Minker, Birte Glimm, and Susanne Biundo. 2021. Do It Yourself, but Not Alone: Companion-Technology for Home Improvement—Bringing a Planning-Based Interactive DIY Assistant to Life. *KI - Künstliche Intelligenz* 35, 3-4 (Nov. 2021), 367–375. https://doi.org/10.1007/s13218-021-00721-x

[8] Pascal Bercher, Felix Richter, Frank Honold, Florian Nielsen, Felix Schussel, Thomas Geier, Thilo Hornle, Stephan Reuter, Daniel Holler, Gregor Behnke, Klaus Dietmayer, Wolfgang Minker, Michael Weber, and Susanne Biundo. 2018. A Companion-System Architecture for Realizing Individualized and Situation-Adaptive User Assistance. *Report Universität Ulm* (2018).

[9] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American* 284, 5 (2001), 34–43. http://www.jstor.org/stable/26059207

[10] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. *CoRR* abs/2004.10934 (2020). arXiv:2004.10934 https://arxiv.org/abs/2004.10934

[11] Stefano Borgo, Roberta Ferrario, Aldo Gangemi, Nicola Guarino, Claudio Masolo, Daniele Porello, Emilio M. Sanfilippo, and Laure Vieu. 2022. DOLCE: A Descriptive Ontology for Linguistic and Cognitive Engineering. *Applied Ontology* 17, 1 (March 2022), 45–69. https://doi.org/10.3233/AO-210259 arXiv:2308.01597 [cs].

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[13] Michael Burmester, Katharina Zeiner, Katharina Schippert, and Axel Platz. 2019. Creating Positive Experiences with Digital Companions. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312821

[14] Davide Calvaresi, Daniel Cesarini, Paolo Sernani, Mauro Marinoni, Aldo Franco Dragoni, and Arnon Sturm. 2017. Exploring the ambient assisted living domain: a systematic review. *Journal of Ambient Intelligence and Humanized Computing* 8, 2 (April 2017). https://doi.org/10.1007/s12652-016-0374-3

[15] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. 2012. The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics* 17 (2012), 25–32. https://doi.org/10.1016/j.websem.2012.05.003

[16] Yuren Cong, Michale Yang, and Bodo Rosenhahn. 2023. RelTR: Relation Transformer for Scene Graph Generation. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence* 45, 09 (sep 2023), 11169–11183. https://doi.org/10.1109/TPAMI.2023.3268066

[17] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula. 2009. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing* 5, 4 (2009), 277–298. https://doi.org/10.1016/j.pmcj.2009.04.001

[18] Bo Dai, Yuqi Zhang, and Dahua Lin. 2017. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE conference on computer vision and Pattern recognition*. 3076–3086.

[19] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc.

[20] Xuhai Duan, Yong Chen, Zuzhen Ji, Zhi Pei, and Wenchao Yi. 2022. Safety-Critical Components Analysis Using Knowledge Graph For CNC Machine. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. 1617–1621. https://doi.org/10.1109/CASE49997.2022.9926662

[21] K Ducatel, M Bogdanowicz, F Scapolo, J Leijten, and J-C Burgelman. 2001. Scenarios for Ambient Intelligence in 2010. *Union européenne. Technologies de la société de l'information, Union européenne. Institut d'études de prospectives technologiques, and Union européenne. Société de l'information conviviale* (2001). http://www.ist.hu/doctar/fp5/istagscenarios2010.pdf

[22] Rob Dunne, Tim Morris, and Simon Harper. 2022. A Survey of Ambient Intelligence. *Comput. Surveys* 54, 4 (May 2022), 1–27. https://doi.org/10.1145/3447242

[23] Roy T. Fielding. 2000. *Architectural styles and the design of network -based software architectures*. Ph. D. Dissertation. https://www.proquest.com/dissertations-theses/architectural-styles-design-network-based/docview/304591392/se-2

[24] Andrea Gaggioli, Alois Ferscha, Giuseppe Riva, Stephen Dunne, Isabell Viaud-Delmon, et al. 2016. *Human computer confluence: Transforming human experience through symbiotic technologies*. Vol. 2016. De Gruyter Open Berlin, Germany.

[25] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. 2002. Sweetening Ontologies with DOLCE. In *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, Asunción Gómez-Pérez and V. Richard Benjamins (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 166–181.

[26] Artur d'Avila Garcez and Luís C. Lamb. 2023. Neurosymbolic AI: the 3rd wave. *Artificial Intelligence Review* 56, 11 (Nov. 2023), 12387–12406. https://doi.org/10.1007/s10462-023-10448-w

[27] Kimberly Garcia, Daniel David Schwyn, and Florian Michahelles. 2020. A Digital Companion for Air Travelers. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, Oldenburg Germany, 1–6. https://doi.org/10.1145/3406324.3410544

[28] Kimberly Garcia, Simon Mayer, Alessandro Ricci, and Andrei Ciortea. 2020. Proactive Digital Companions in Pervasive Hypermedia Environments. In *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, Atlanta, GA, USA, 54–59. https://doi.org/10.1109/CIC50333.2020.00017

[29] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[30] Horst-Michael Gross, Andrea Scheidig, Steffen Muller, Benjamin Schutz, Christa Fricke, and Sibylle Meyer. 2019. Living with a Mobile Companion Robot in your Own Apartment - Final Implementation and Results of a 20-Weeks Field Study with 20 Seniors. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, Montreal, QC, Canada, 2253–2259. https://doi.org/10.1109/ICRA.2019.8793693

[31] Thomas R Gruber. 1991. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. *KR 91* (1991).

[32] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. 2019. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1969–1978.

[33] Dominique Guinard and Vlad Trifa. 2009. Towards the Web of Things. Web Mashups for Embedded Devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*. MEM 2009, Madrid.

[34] Didier Guzzoni. 2007. Active: A unified platform for building intelligent assistant applications. (2007).

[35] Ian Horrocks. 2007. Semantic Web: The Story so Far. In *Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A)* (Banff, Canada) *(W4A '07)*. Association for Computing Machinery, New York, NY, USA, 120–125. https://doi.org/10.1145/1243441.1243469

[36] Seyed Ramezan Hosseini, Alireza Taheri, Ali Meghdari, and Minoo Alemi. 2018. "Let There Be Intelligence!"- A Novel Cognitive Architecture for Teaching Assistant Social Robots. In *Social Robotics*. Springer International Publishing. https://doi.org/10.1007/978-3-030-05204-1_27

[37] Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*.

[38] Thilo Hörnle, Michael Tornow, Frank Honold, Reinhard Schwegler, Ralph Heinemann, Susanne Biundo, and Andreas Wendemuth. 2017. Companion-Systems: A Reference Architecture. In *Companion Technology*, Susanne Biundo and Andreas Wendemuth (Eds.). Springer International Publishing, Cham, 449–469. https://doi.org/10.1007/978-3-319-43665-4_22

[39] Filip Ilievski, Pedro Szekely, and Bin Zhang. 2021. CSKG: The CommonSense Knowledge Graph. In *The Semantic Web ESWC 2021*, Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Oscar Corcho, Petar Ristoski, and Mehwish Alam (Eds.). Springer International Publishing, Cham, 680–696.

[40] Eleni Ilkou and Maria Koutraki. 2020. *Symbolic Vs Sub-symbolic AI Methods: Friends or Enemies?* https://doi.org/10.1145/3340531.3414072

[41] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3668–3678.

[42] Qasim Khalid, Alberto Fernandez, Marin Lujak, and Arnaud Doniec. 2023. SBEO: Smart Building Evacuation Ontology. *Computer Science and Information Systems* 20, 1 (2023), 51–76.

[43] Mareike Kritzler, Jack Hodges, Dan Yu, Kimberly Garcia, Hemant Shukla, and Florian Michahelles. 2019. Digital Companion for Industry. In *Companion Proceedings of The 2019 World Wide Web Conference* (San Francisco, USA) *(WWW '19)*. Association for Computing

Machinery, New York, NY, USA, 663–667. https://doi.org/10.1145/3308560.3316510
[44] Amit Kumar Pandey, Rodolphe Gelin, Rachid Alami, Renaud Viry, Axel Buendia, Roland Meertens, Mohamed Chetouani, Laurence Devillers, Marie Tahon, David Filliat, Yves Grenier, Mounira Maazaoui, Abderrahmane Kheddar, Frédéric Lerasle, and Laurent Fitte-Duval. 2014. Romeo2 Project: Humanoid Robot Assistant and Companion for Everyday Life: I. Situation Assessment for Social Intelligence. In *AIC: Artificial Intelligence and Cognition*. CEUR Workshop Proceedings (CEUR-WS.org), 140–147.
[45] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'ou Tang. 2017. Vip-cnn: Visual phrase guided convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
[46] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. 2018. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 335–351.
[47] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2022. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems* 33, 12 (2022), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827
[48] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*.
[49] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. 2016. Visual relationship detection with language priors. In *European conference on computer vision*. Springer, 852–869.
[50] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
[51] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767 http://arxiv.org/abs/1804.02767
[52] Sílvia Resendes, Paulo Carreira, and André C. Santos. 2014. Conflict detection and resolution in home and building automation systems: a literature review. *Journal of Ambient Intelligence and Humanized Computing* 5, 5 (Oct. 2014), 699–715. https://doi.org/10.1007/s12652-013-0184-9
[53] Ayanda Rogge. 2023. Defining, Designing and Distinguishing Artificial Companions: A Systematic Literature Review. *International Journal of Social Robotics* 15, 9 (Oct. 2023), 1557–1579. https://doi.org/10.1007/s12369-023-01031-y
[54] Andrei Vlad Sambra, Essam Mansour, Sandro Hawke, Maged Zereba, Nicola Greco, Abdurrahman Ghanem, Dmitri Zagidulin, Ashraf Aboulnaga, and Tim Berners-Lee. 2016. Solid: a platform for decentralized social applications based on linked data. *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.* (2016).
[55] Ruhi Sarikaya. 2017. The Technology Behind Personal Digital Assistants: An overview of the system architecture and key components. *IEEE Signal Processing Magazine* 34, 1 (2017), 67–81. https://doi.org/10.1109/MSP.2016.2617341
[56] Barış Serim and Giulio Jacucci. 2019. Explicating "Implicit Interaction": An Examination of the Concept and Challenges for Research. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3290605.3300647
[57] Maximilian Speicher, Brian D. Hall, and Michael Nebeling. 2019. What is Mixed Reality?. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300767
[58] Janick Spirig, Kimberly Garcia, and Simon Mayer. 2021. An Expert Digital Companion for Working Environments. In *11th International Conference on the Internet of Things*. ACM, St.Gallen Switzerland, 25–32. https://doi.org/10.1145/3494322.3494326
[59] Steffen Staab and Rudi Studer (Eds.). 2009. *Handbook on Ontologies*. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-92673-3
[60] Chairs Constantine Stephanidis, Gavriel Salvendy, Members of the Group Margherita Antona, Jessie Y. C. Chen, Jianming Dong, Vincent G. Duffy, Xiaowen Fang, Cali Fidopiastis, Gino Fragomeni, Limin Paul Fu, Yinni Guo, Don Harris, Andri Ioannou, Kyeong ah (Kate) Jeong, Shin'ichi Konomi, Heidi Krömker, Masaaki Kurosu, James R. Lewis, Aaron Marcus, Gabriele Meiselwitz, Abbas Moallem, Hirohiko Mori, Fiona Fui-Hoon Nah, Stavroula Ntoa, Pei-Luen Patrick Rau, Dylan Schmorrow, Keng Siau, Norbert Streitz, Wentao Wang, Sakae Yamamoto, Panayiotis Zaphiris, and Jia Zhou. 2019. Seven HCI Grand Challenges. *International Journal of Human–Computer Interaction* 35, 14 (2019), 1229–1269. https://doi.org/10.1080/10447318.2019.1619259
[61] Ron Sun. 2006. *Cognition and multi-agent interaction: From cognitive modeling to social simulation*. Cambridge University Press.
[62] Ioan Szilagyi and Patrice Wira. 2016. Ontologies and Semantic Web for the Internet of Things - a survey. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 6949–6954. https://doi.org/10.1109/IECON.2016.7793744
[63] Nava Tintarev and Judith Masthoff. 2007. Effective explanations of recommendations: user-centered design. In *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, Minneapolis MN USA, 153–156. https://doi.org/10.1145/1297231.1297259
[64] Nava Tintarev and Judith Masthoff. 2012. Evaluating the effectiveness of explanations for recommender systems: Methodological issues and empirical studies on the impact of personalization. *User Modeling and User-Adapted Interaction* 22, 4-5 (Oct. 2012), 399–439. https://doi.org/10.1007/s11257-011-9117-5

[65] Stefan Tretter, Axel Platz, and Sarah Diefenbach. 2023. Matching Mind and Method: Augmented Decision-Making with Digital Companions based on Regulatory Mode Theory. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–10. https://doi.org/10.1145/3544548.3581529

[66] Danai Vachtsevanou, Andrei Ciortea, Simon Mayer, and Jérémy Lemée. 2023. Signifiers as a First-Class Abstraction in Hypermedia Multi-Agent Systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1200–1208.

[67] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv:2207.02696 http://arxiv.org/abs/2207.02696

[68] Yulu Wang, Pengwen Dai, Xiaojun Jia, Zhitao Zeng, Rui Li, and Xiaochun Cao. 2023. Hi-SIGIR: Hierachical Semantic-Guided Image-to-Image Retrieval via Scene Graph. In *Proceedings of the 31st ACM International Conference on Multimedia* (Ottawa ON, Canada) *(MM '23)*. Association for Computing Machinery, New York, NY, USA, 6400–6409. https://doi.org/10.1145/3581783.3612283

[69] Mark Weiser. 1999. The computer for the 21 st century. *ACM SIGMOBILE Mobile Computing and Communications Review* 3, 3 (July 1999), 3–11. https://doi.org/10.1145/329124.329126

[70] Andreas Wendemuth and Susanne Biundo. 2012. A Companion Technology for Cognitive Technical Systems. In *Cognitive Behavioural Systems*. Vol. 7403. Springer Berlin Heidelberg, Berlin, Heidelberg, 89–103. https://doi.org/10.1007/978-3-642-34584-5_7

[71] Michael Wooldridge. 2002. Intelligent Agents: The Key Concepts. In *Multi-Agent Systems and Applications II*, G. Goos, J. Hartmanis, J. Van Leeuwen, Vladimír Mařík, Olga Štěpánková, Hana Krautwurmová, and Michael Luck (Eds.). Vol. 2322. Springer Berlin Heidelberg, Berlin, Heidelberg, 3–43. https://doi.org/10.1007/3-540-45982-0_1

[72] E Zelkha, B Epstein, S Birrell, and C Dodsworth. 1998. From Devices to. In *Ambient Intelligence"", Digital Living Room Conference (June 1998)*.

A USER STUDY QUESTIONNAIRE

After experiencing the AmI system, participants were asked to answer the following questionnaire.

(1) The system was successful in conveying the robot actions and its status

Strongly disagree				Strongly agree
1	2	3	4	5

(2) The scene graph the system shows was easy to understand

Strongly disagree				Strongly agree
1	2	3	4	5

(3) The instructions the system provided me with were easy to understand

Strongly disagree				Strongly agree
1	2	3	4	5

(4) Did you find anything unclear while using the system?

---

(5) I was able to forecast the behavior of the robot given the information the system provided me with

Strongly disagree				Strongly agree
1	2	3	4	5

(6) I understood the decision making process of the robot

Strongly disagree				Strongly agree
1	2	3	4	5

(7) It was not hard to understand the current state of the robot

Strongly disagree				Strongly agree
1	2	3	4	5

(8) The system was effective at guiding me in the situations that required me to take action

Strongly disagree				Strongly agree
1	2	3	4	5

(9) I was confident in the system's ability to navigate me towards a safety place

Strongly disagree				Strongly agree
1	2	3	4	5

(10) I trust the system to provide me with accurate information

Strongly disagree				Strongly agree
1	2	3	4	5

(11) The system was fast enough to direct me to a safe place

Strongly disagree				Strongly agree
1	2	3	4	5

(12) The system provided me with all the required information to handle the situations I faced

Strongly disagree				Strongly agree
1	2	3	4	5

(13) The system help me locating the fire extinguisher or leave the area at risk

Strongly disagree				Strongly agree
1	2	3	4	5

(14) In the event of a real emergency, I am likely to use a system like the one I just experienced

Strongly disagree				Strongly agree
1	2	3	4	5

(15) I am satisfied with the overall experience using the system

Strongly disagree				Strongly agree
1	2	3	4	5

(16) Were there any points of confusion while using the system?

---

(17) Did the system frustrate you in any particular way?

---

(18) Is there anything else you would like to share?
